EDITED BY
VICTORIA YANEVA
AND MATTHIAS VON DAVIER

ADVANCING NATURAL LANGUAGE PROCESSING IN EDUCATIONAL ASSESSMENT

APPLICATIONS OF
EDUCATIONAL MEASUREMENT
AND ASSESSMENT

NCME
national
council on
measurement
in education

ROUTLEDGE

Advancing Natural Language Processing in Educational Assessment

Advancing Natural Language Processing in Educational Assessment examines the use of natural language technology in educational testing, measurement, and assessment. Recent developments in natural language processing (NLP) have enabled large-scale educational applications, though scholars and professionals may lack a shared understanding of the strengths and limitations of NLP in assessment as well as the challenges that testing organizations face in implementation. This first-of-its-kind book provides evidence-based practices for the use of NLP-based approaches to automated text and speech scoring, language proficiency assessment, technology-assisted item generation, gamification, learner feedback, and beyond. Spanning historical context, validity and fairness issues, emerging technologies, and implications for feedback and personalization, these chapters represent the most robust treatment yet about NLP for education measurement researchers, psychometricians, testing professionals, and policymakers.

Victoria Yaneva is Senior NLP Scientist at the National Board of Medical Examiners, USA.

Matthias von Davier is Monan Professor of Education in the Lynch School of Education and Executive Director of TIMSS & PIRLS International Study Center at Boston College, USA.

The NCME Applications of Educational Measurement and Assessment Book Series

Editorial Board:

Brian E. Clauser, National Board of Medical Examiners, Editor
Henry Braun, Boston College
Robert L. Brennan, The University of Iowa
Fritz Drasgow, The University of Illinois
Michael J. Kolen, The University of Iowa
Rochelle Michel, Educational Testing Service

Meeting the Challenges to Measurement in an Era of Accountability
Edited by Henry Braun

Fairness in Educational Assessment and Measurement
Edited by Neil J. Dorans and Linda L. Cook

Testing in the Professions: Credentialing Policies and Practice
Edited by Susan Davis-Becker and Chad W. Buckendahl

Validation of Score Meaning for the Next Generation of Assessments: The Use of Response Processes
Edited by Kadriye Ercikan and James W. Pellegrino

Preparing Students for College and Careers: Theory, Measurement, and Educational Practice
Edited by Katie Larsen McClarty, Krista D. Mattern, and Matthew N. Gaertner

Score Reporting Research and Applications
Edited by Diego Zapata-Rivera

Classroom Assessment and Educational Measurement
Edited by Susan M. Brookhart and James H. McMillan

Integrating Timing Considerations to Improve Testing Practices
Edited by Melissa J. Margolis and Richard A. Feinberg

Advancing Natural Language Processing in Educational Assessment
Edited by Victoria Yaneva and Matthias von Davier

For more information about this series, please visit: www.routledge.com/NCME-APPLICATIONS-OF-EDUCATIONAL-MEASUREMENT-AND-ASSESSMENT/book-series/NCME

Advancing Natural Language Processing in Educational Assessment

Edited by Victoria Yaneva and Matthias von Davier

Routledge
Taylor & Francis Group
NEW YORK AND LONDON

Designed cover image: © Getty images

First published 2023
by Routledge
605 Third Avenue, New York, NY 10158

and by Routledge
4 Park Square, Milton Park, Abingdon, Oxon, OX14 4RN

Routledge is an imprint of the Taylor & Francis Group, an informa business

© 2023 selection and editorial matter, Victoria Yaneva and Matthias von Davier; individual chapters, the contributors

The right of Victoria Yaneva and Matthias von Davier to be identified as the authors of the editorial material, and of the authors for their individual chapters, has been asserted in accordance with sections 77 and 78 of the Copyright, Designs and Patents Act 1988.

The Open Access version of this book, available at www.taylorfrancis.com, has been made available under a Creative Commons Attribution-Non Commercial-No Derivatives 4.0 license.

Trademark notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Library of Congress Cataloging-in-Publication Data
Names: Yaneva, Victoria, editor. | Davier, Matthias von, editor.
Title: Advancing natural language processing in educational assessment / edited by Victoria Yaneva and Matthias von Davier.
Description: New York, NY : Routledge, 2023. | Series: NCME applications of educational measurement and assessment | Includes bibliographical references and index.
Identifiers: LCCN 2022061829 | ISBN 9781032203904 (hardback) | ISBN 9781032244525 (paperback) | ISBN 9781003278658 (ebook)
Subjects: LCSH: Educational tests and measurements—Technological innovations. | Natural language processing (Computer science)
Classification: LCC LB3060.5 .A48 2023 | DDC 371.26/1—dc23/eng/20230221
LC record available at https://lccn.loc.gov/2022061829

ISBN: 978-1-032-20390-4 (hbk)
ISBN: 978-1-032-24452-5 (pbk)
ISBN: 978-1-003-27865-8 (ebk)

DOI: 10.4324/9781003278658

Typeset in Minion Pro
by Apex CoVantage, LLC

Contents

Preface by Victoria Yaneva and Matthias von Davier vii

Part I: Automated Scoring 1

1 The Role of Robust Software in Automated Scoring 3
NITIN MADNANI, AOIFE CAHILL, AND ANASTASSIA LOUKINA

2 Psychometric Considerations When Using Deep Learning for Automated Scoring 15
SUSAN LOTTRIDGE, CHRIS ORMEROD, AND AMIR JAFARI

3 Speech Analysis in Assessment 31
JARED C. BERNSTEIN AND JIAN CHENG

4 Assessment of Clinical Skills: A Case Study in Constructing an NLP-Based Scoring System for Patient Notes 58
POLINA HARIK, JANET MEE, CHRISTOPHER RUNYON, AND BRIAN E. CLAUSER

Part II: Item Development 75

5 Automatic Generation of Multiple-Choice Test Items from Paragraphs Using Deep Neural Networks 77
RUSLAN MITKOV, LE AN HA, HALYNA MASLAK, THARINDU RANASINGHE, AND VILELMINI SOSONI

6 Training Optimus Prime, M.D.: A Case Study of Automated Item Generation Using Artificial Intelligence – From Fine-Tuned GPT2 to GPT3 and Beyond 90
MATTHIAS VON DAVIER

7 Computational Psychometrics for Digital-First Assessments: A Blend of ML and Psychometrics for Item Generation and Scoring 107

GEOFF LAFLAIR, KEVIN YANCEY, BURR SETTLES, AND ALINA A VON DAVIER

Part III: Validity and Fairness **125**

8 Validity, Fairness, and Technology-Based Assessment 127

SUZANNE LANE

9 Evaluating Fairness of Automated Scoring in Educational Measurement 142

MATTHEW S. JOHNSON AND DANIEL F. MCCAFFREY

Part IV: Emerging Technologies **165**

10 Extracting Linguistic Signal From Item Text and Its Application to Modeling Item Characteristics 167

VICTORIA YANEVA, PETER BALDWIN, LE AN HA, AND CHRISTOPHER RUNYON

11 Stealth Literacy Assessment: Leveraging Games and NLP in iSTART 183

YING FANG, LAURA K. ALLEN, ROD D. ROSCOE, AND DANIELLE S. MCNAMARA

12 Measuring Scientific Understanding Across International Samples: The Promise of Machine Translation and NLP-Based Machine Learning Technologies 200

MINSU HA AND ROSS H. NEHM

13 Making Sense of College Students' Writing Achievement and Retention With Automated Writing Evaluation 217

JILL BURSTEIN, DANIEL F. MCCAFFREY, STEVEN HOLTZMAN, AND BEATA BEIGMAN KLEBANOV

Contributor Biographies 235
Index 244

Preface

> *From its beginnings in 1967, the use of NLP for scoring has provoked concerns and even outrage. Over the ensuing 55 years, the technology has seen remarkable advances and, coupled with exponentially greater computing power, NLP has borne out the most extravagant promises of its early proponents – with more surprises on the horizon.*
>
> Henry Braun,
> Boisi Professor of Education and Public Policy and
> Education Research at Boston College
> Formerly Vice President of Research at ETS[1]

Technological developments continuously amaze and sometimes unsettle us by transforming and enriching almost every area of modern life. Educational assessment is no exception – like other fields, the use of technology has not only enabled improving tasks such as test delivery and data collection, but has ushered a creative wave of entirely new approaches to assessing what test-takers know and can do. It is enough to point to the transition from paper-and-pencil exams to computer-based test delivery and its many advantages – from accurate measures of timing to automated scoring and computer adaptive testing – to seal the argument for the pivotal place technology occupies in current assessment practice. But what about technological advances for tackling the *content* of exams? Has language technology matured enough to give hope that fully automated item generation is within reach? Can advances in natural language processing (NLP) be leveraged to measure new constructs? What are some emerging applications of NLP that promise shifts in assessment practice? And, last but not least, how do we uphold the pillars of psychometrics – validity, reliability, and fairness – in the face of a growing lack of interpretability in machine-learning models?

In putting together this volume, we aim to address some of the questions that the field of human assessment grapples with when it comes to the practical use of NLP. This book will be useful to educational researchers, assessment researchers, psychometricians, and practitioners who may have limited background in NLP but want insights into how NLP can improve assessment practice and the data we collect. The chapters of this volume aspire to introduce important concepts and methods to novice readers. Experts in language technology will also benefit from understanding the challenges that testing organizations face in implementing NLP applications in practice. Ultimately, it is our hope to bring the NLP and assessment communities closer together to accelerate innovation in this cross-disciplinary area.

Part I of this volume is dedicated to the topic of automated scoring of text and speech and covers its historical context, psychometric and validity considerations, best practices for building robust software, as well as use cases requiring automated concept matching between a rubric and a student response. In this section, one notices an imbalance favoring automated scoring for assessing language proficiency, with only one chapter focusing on a different domain (assessing clinical reasoning). This imbalance occurred naturally since the field of automated scoring has been preoccupied with language proficiency, where NLP solutions are highly accurate in evaluating the spelling, grammaticality, and coherence of responses, as its most feasible application to date. For other domains, where spelling and syntax are construct-irrelevant (as in the case of the chapter on clinical reasoning), automated scoring based on concept mapping emerges as a novel approach. In the heyday of NLP advances in semantic processing, such content-scoring applications that go beyond the realm of language proficiency should be a natural next step.

Part II discusses various aspects related to technology-supported item generation. These include automated item generation using traditional and deep learning approaches, the generation of reading passages, and their alignment to standards and instructional materials. The studies presented in this section feed into the enthusiasm that fully automated item generation (AIG) can become common practice. While, in some cases, AIG is already used in practice, this application of technology appears, once again, most feasible in testing for language proficiency. State of the art in automated generation of reading passages at the desired readability level for reading comprehension tests is much more advanced than that of generating rich, factually correct items for other domains (e.g., medicine or law). In such domains, technology-assisted item writing, such as a system suggesting distractors to a human writer, is currently explored as a more feasible application of NLP.

Part III of this volume focuses on validity and fairness implications when using language technology in assessment. Just as is the case when evaluating the fairness of tests in general, or the validity of human scoring in particular, scores that were generated using NLP technologies have to be scrutinized to ensure that they align with the intended construct. It is equally important that differences in NLP-generated scores are not affected by construct-irrelevant differences between subgroups. The chapters in Part III describe the state of the art in this area, while also exposing the fact that the same dilemma that affects the evaluation of human scores and tests in general also holds when looking at NLP- or AI-based scores: we have to come to grips with the fact that we can only evaluate the consequences of the decisions made. In both scenarios, we do not have direct access to what the true underlying differences are between the groups for which we desire to assess fairness and validity of scores.

Finally, Part IV is a look towards the future: it presents emerging technologies such as the automated prediction of item characteristics, stealth literacy assessment, and the use of machine translation for automated scoring in international samples. The conclusion discusses the potential for a shift from assessment technology to technology for feedback and personalization. These cutting-edge applications offer a glimpse into a reimagined assessment practice, where technology is used to fundamentally transform the way we construct tests. Such solutions branch out from the intersection of NLP and assessment and engage the even wider ecosystem of learning analytics, learning science, and serious games.

For many of the questions discussed here, the chapters in the volume suggest approaches and evaluate their utility but do not pretend to posit a simple, conclusive answer. This was intended, as we hope this volume will advance the discussion of use cases, issues, and perspectives for solutions around the use of NLP in assessment. Indeed, many open questions remain to be solved, and close collaboration between the NLP and assessment communities is needed.

Among the questions that need to be addressed, we feel there are several emerging themes that urgently need attention:

- *Interpretability vs. accuracy:* Deep learning models for automated scoring are very useful but are black boxes. Do we need interpretable models? Are human-provided scores interpretable? Or are both trained, but imperfect implementations of the scoring intent? Should the latter be the case, we can only monitor accuracy and consistency over time; we will not be able to explain at a deep level how scores were generated.
- *Implications for fairness:* Do NLP-based technologies mirror the fairness issues of human scoring? Or can these technologies be used to reduce (or even eliminate) fairness issues resulting from certain aspects of human-authored items and human scoring?
- *Bias:* How important it is to understand the effects of training data, and how can we evaluate bias? What can we learn from issues that have been observed when training and scoring or from generative models using limited training data? Can sampling/harvesting of training data using principles developed in other domains such as sampling statistics or quality control help reduce bias?

These questions remain at the forefront of current assessment practice that involves NLP and will likely persist in one form or another as our understanding of fairness and validity evolves.

Being the nexus of many fields with their own open questions and limitations, the topic of NLP in assessment is a complex yet highly productive area, which requires interdisciplinary understanding. Therefore, the chapters presented here would not have had the breadth and depth of their current form had it not been for the thorough and constructive feedback of the NLP and assessment experts who served as reviewers. Our gratitude goes to (in alphabetical order by first name): Alina von Davier, Brian E. Clauser, Christopher Runyon, Danielle S. McNamara, Janet Mee, Kimberly Swygert, Le An Ha, Matthew S. Johnson, Monica Cuddy, Nitin Madnani, Peter Baldwin, Richard Evans, Richard Feinberg, Susan Lottridge, Suzanne Lane, and Thai Ong. We are also indebted to Kerbie Addis for her patience and thoroughness in collecting the metadata for each chapter. Last but far from least, we greatly appreciate the valuable feedback we received from the NCME Editorial Board reviewers – Brian E. Clauser and Roy Levy.

Victoria Yaneva and Matthias von Davier
September 26, 2022

Note

1 Professor Braun was involved in establishing research groups around NLP and automated essay scoring at ETS more than three decades ago.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Part I

Automated Scoring

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

1

The Role of Robust Software in Automated Scoring

Nitin Madnani, Aoife Cahill, and Anastassia Loukina

1. Introduction

Automated scoring applications are, first and foremost, pieces of software. This aspect of automated scoring has often been overlooked in the literature, which generally focuses on functionality and evaluation metrics and leaves implementation details to a technical footnote. Developing software for automated scoring – an application that can have a potentially profound personal and societal impact – tends to be uniquely challenging. Recently, several publications discussed various practical aspects of developing operational automated scoring software and putting it into production (Lottridge & Hoefer, 2020; Schneider & Boyer, 2020; Shaw et al., 2020). In this chapter, we continue to add to this discussion. We propose that to implement automated scoring at scale *and* support the validity of the automated scores being produced, it is essential that the software used is *robust*, i.e., well-developed, well-tested, and well-documented, and we discuss some practical steps necessary to achieve this goal.[1]

Software used in modern automated scoring applications encompasses more than just the machine learning[2] model that predicts a final score for a given response. While such *scoring models* are the most well-known and well-discussed parts of these applications, many other application components are also based on pre-trained machine learning models (e.g., automatic speech recognition (ASR) for speech scoring, grammatical error detection, content scoring features, and many others) in addition to having further dependencies on external resources (e.g., corpora or open-source libraries). Some of these numerous dependencies may require only sporadic updates – once every few years – whereas others may require more frequent attention depending on their context of use. For example, when new items are added to the pool of automatically scored items, the scoring models used in an application for automatically scoring *writing quality* may not require updates. On the other hand, new scoring models might need to be trained for each new item for applications that automatically score content knowledge. As another example, consider that a part-of-speech tagging model, or the ASR acoustic model, might need to be updated when significant differences in the demographics of the test-taking population are reflected in the characteristics of the written or spoken responses

DOI: 10.4324/9781003278658-2
This chapter has been made available under a CC-BY-NC-ND 4.0 license.

being submitted to the system. Whether a new model is added to the application or an existing model is updated, it is critical to re-evaluate the quality of that model's predictions and the downstream scores assigned by the application to make sure that they remain accurate, valid, and fair.

Given this line of reasoning, it should be clear that the software for training and evaluating machine learning models – while physically separate from the software powering the scoring application – relies on the same set of dependencies and needs to follow the same robustness principles to ensure that the integrity of the final scores is not compromised. A further complexity with any machine-learning-based application[3] such as automated scoring is that the software for training a system is typically separate from the software that ultimately makes a prediction for a specific input (potentially even maintained by different teams). Maintaining alignment between shared components (models, resources, dependencies) is essential for ensuring a valid end-to-end system. In addition, the training and prediction pipelines may have different requirements in terms of computational resources or speed of computation. Therefore, in this chapter, we use the term *automated scoring software* to refer to all three of the following: the software and models used in scoring applications, the software used to train said machine learning models, and the software used to evaluate those models.

This chapter makes the following contributions: (1) It introduces concepts from the field of software engineering as they relate to the application of automated scoring; (2) it discusses some of the trade-offs and considerations that need to be kept in mind when developing automated scoring software, and (3) it outlines procedures for the development of robust automated scoring software. Our goal is to make it clear to readers that the task of developing *robust* automated scoring software is much more than simply developing a smart algorithm, and to describe some techniques for implementing applications that are easier and cheaper to maintain in the long term. Of course, many of the techniques and decision points discussed here are relevant for the development and maintenance of any application whose core is a machine learning algorithm, but here we focus only on the development of software for automated scoring applications.

2. Multiple Stakeholders With Different Needs

Automated scoring has multiple stakeholders (Madnani & Cahill, 2018). These include:

- *NLP scientists and engineers:* They are responsible for designing and implementing an automated scoring solution that adequately measures the construct of interest as defined by subject matter experts while remaining within any constraints set by other stakeholders (e.g., compute or time requirements). Their focus is usually on the NLP algorithms and the accuracy and fairness of the predicted scores.
- *Other experts involved in developing automated scoring engines – subject matter experts and psychometricians:* This group of stakeholders would like to ensure that any automated scoring system deployed to score the assessments is consistent with the scoring rubric, that only construct-relevant information is used by the system during the scoring process, and that the final scores comply with standards for accuracy, validity, and fairness.
- *Product managers and other business representatives:* Their task is to put the software on the market. In addition to the accuracy, fairness, and validity of automated scores, these stakeholders also care about low costs and fast turnaround time for development, deployment, and score predictions.
- *Score users – teachers, students, and institutions:* These users care that the automated scores are valid, accurate, and unbiased.

All these users/stakeholders have different needs. In the rest of this section, we discuss the process through which changes should be introduced to an already-deployed engine, and the potentially conflicting points of view that arise.

- NLP scientists and engineers need to make sure the engine codebase remains compatible with current versions of all external software packages and is secure, stable, and performant. They also want to continuously improve the code given the latest advances in natural language processing, machine learning, and software development.
- Experts want to ensure that the engine scores remain valid, and no group of users is disadvantaged by a change.
- Business units want to react to market needs quickly.
- Score users want new features, in addition to requiring accurate, fair, and reliable scores. Therefore, we need to ensure that, for example, scores do not change during the administration.

Our proposed workflow ensures that all subsequent changes to the code, including the most trivial ones, are documented and can be reviewed and reverted as necessary. But who should make the final decision? One solution here is to follow the principles of continuous integration, where once the code changes pass the code review, they are added to the main engine. An alternative is an approach in which the main engine code is changed very rarely and only after a comprehensive review and approval by multiple experts.

The tension between improving an automated scoring engine and maintaining consistency can influence which kind of engine updates are implemented in any given scenario, with high-stakes applications favoring a slow-to-update solution and lower-stakes applications generally being more accepting of the risks and cohort limitations associated with the continuous update solutions. In the subsequent sections, we discuss the advantages and disadvantages of both approaches and how they might affect the four groups of stakeholders.

2.1 Continuous Engine Updates

The advantage of continuously updating the engine once proposed changes have been reviewed is that it is easy to respond to user feedback. Business units and score users may desire such frequent updates. For example, if a client notices that the engine generates incorrect scores or feedback, the developers can investigate and implement a solution and redeploy an updated engine. However, there is a tension here between quickly addressing user needs while at the same time maintaining the accuracy, validity, and fairness of the engine (at ETS, we generally use these terms as defined by Williamson et al., 2012). Typically, the evaluations necessary to comprehensively document an engine's accuracy, validity, and fairness require human expertise and time. Of course, many of the evaluations can be automated. However, it may still be necessary to include a human in the loop to ensure that (a) no additional evaluations are necessary due to the change or (b) allow flexibility for borderline cases. Continuous updates may also introduce fairness concerns, especially for high-stakes applications: If two *different* versions of the engine have produced the scores for two test-takers, is it appropriate to treat these scores as equivalent for high-stakes decisions such as university admissions? Should all scores be updated with the engine release? And what happens if the change in the engine leads to a lower score for a given test-taker?

2.2 Conservative Engine Updates

A more conservative approach to engine updates is to update them rarely and on a fixed schedule agreed well in advance by all stakeholders (e.g., every 1–5 years). This approach is generally favored by experts focused on engine quality and stability. Infrequent updates have the advantage of maintaining consistency of the automated score predictions and feedback over a long period of time. It also facilitates consistency within student cohorts since the updates can be planned exactly for a time that allows for a cohort to be completely processed before the update. Another advantage of the conservative approach is that it allows for a much more comprehensive evaluation of the proposed changes. A comprehensive evaluation is time-consuming and expensive in terms of the human effort required from experts. Typically, such an evaluation would include analyses from: (a) assessment developers (to study the relevance of the changes to the construct measurement); (b) psychometricians and statisticians (to study the empirical impact of the changes, not only at the individual item level, but also at section and test levels); (c) experts in automated scoring (to evaluate the updates along the many relevant dimensions); and (d) business units (to study the impact of changes related to non-scorable responses, or changes in processing time/compute requirements). Such thorough evaluations can provide evidence to support the accuracy, validity, and fairness of the updated engine.

There are also disadvantages to this approach. Specifically, very infrequent engine updates mean that improvements in response to client feedback cannot be implemented in a reasonable timeframe. Infrequent updates also preclude the inclusion of advances in state-of-the-art NLP techniques, which have the potential to lead to better score prediction and feedback from the engine.

3. Best Practices for Development

An automated scoring application is, at the end of the day, a complex piece of machine learning–based software, and as such, its implementation – like that of any other high-stakes application – must follow industry best practices (e.g., those outlined in Spolsky, 2004) and relevant aspects of current industry standards (e.g., the IEEE Standard of Software Quality Assurance 730–2014). Adhering to these practices ensures that the logic used to compute the automated scores remains consistent with the original intent, free of major bugs, and reproducible at any point in time.

This section presents four important elements of such best practices: *comprehensive testing*, *version control*, *reproducibility*, and *code review*. We first discuss each of these in greater detail and show how they are integral to ensuring the validity and reproducibility of automated scores.

3.1 Comprehensive Testing

An automated scoring application is, in essence, a sequence of steps that maps the submitted response to the final predicted score. These steps can include pre-processing the submitted response, computation of specific NLP features (e.g., part-of-speech tags, syntactic parses), running the computed features through an already-trained machine learning model to compute the raw score, and performing statistical transformations on the raw prediction to produce the final score. Many of these steps are relatively complex, and errors could potentially be introduced during their implementation, which would negatively affect the validity of the final scores.

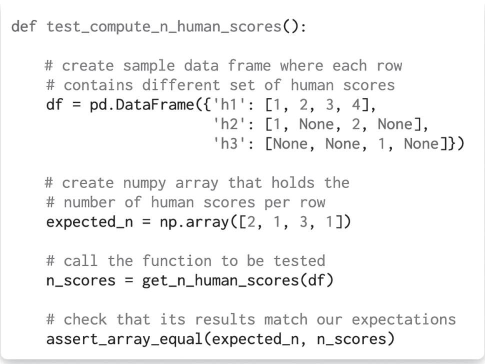
A well-known and effective solution in software engineering literature is to incorporate comprehensive testing into the process. Software testing, in general, is a large undertaking. There are professional testers whose responsibilities include ensuring software quality from a user's perspective. Here we focus only on testing from the software developer's point of view, since typically the 'user' of automated scoring software is another piece of software that integrates the automated scoring output into a downstream workflow. Tests are small subroutines that explicitly define what output should be produced by a piece of code given a *specific* input. Tests can be written at different levels of granularity:

1. The first type of tests are *unit tests*, i.e., very specific tests that use a single input (usually embedded in the test itself) and compare the output computed at test time with previously known or expected output. These tests should have a very narrow and well-defined scope. For example, a series of unit tests might test a computational routine that calculates the total number of pauses in an automated speech transcription. These tests would then confirm that the code returns expected results given several sample transcriptions, including the so-called ‘edge cases’, where, for example, a transcription may not include any pauses at all or may only include pauses and nothing else. See Figure 1.1 for an example of a unit test.
2. The second type of tests are *functional tests*, which are generally written from the users’ perspective to test that the entire application is behaving as expected. For example, one such test would submit an input response, have it run through the entire automated scoring application, compute its final predicted score, and match that prediction against the correct, expected score. Functional tests (also known as *integration tests*) can also be used for model deployments, i.e., to validate whether a deployed model produces the same predictions on a known dataset as the originally trained model. These tests can also be used to confirm that the automated scoring system, as a whole, can correctly deal with atypical or unexpected inputs, e.g., responses with no punctuation or those containing random keystrokes.

Writing effective tests not only helps confirm the validity of the initial implementation but is also critical to guard against any unexpected changes to the code that may impact the reproducibility of previously generated automated scores as further development is carried out. To maximize the effectiveness of such test-driven development, one must ensure that any proposed changes to the application are *always* accompanied by tests that adequately cover any newly added code. We will discuss this in greater detail later in this section.

3.2 Version Control

The second cornerstone of responsible software development that is necessary to ensure validity and reproducibility of automated scores is to check the initial implementation into *version*



```
def test_compute_n_human_scores():

    # create sample data frame where each row
    # contains different set of human scores
    df = pd.DataFrame({'h1': [1, 2, 3, 4],
                       'h2': [1, None, 2, None],
                       'h3': [None, None, 1, None]})

    # create numpy array that holds the
    # number of human scores per row
    expected_n = np.array([2, 1, 3, 1])

    # call the function to be tested
    n_scores = get_n_human_scores(df)

    # check that its results match our expectations
    assert_array_equal(expected_n, n_scores)
```

Figure 1.1 A Python unit test for a function ‘get_n_human_scores()’ that counts the number of human scores for each response. The test creates a sample data frame and confirms that the numbers returned by the function match the expected counts.

control (also known as source control or source code management) from the very beginning and to track and review all further changes made to the codebase. Popular version control software includes *Git*, *Mercurial*, *Subversion*, and *Team Foundation Server*.

Currently the most widely used, open-source software for version control, Git powers application suites such as GitHub[4] or Atlassian Bitbucket[5] that enable a team to set up efficient, collaborative development workflows for very large codebases. There are several possible workflows that teams working on developing automated scoring applications can follow; the simplest one is to create a new branch ("version") in the code repository for every feature or bug fix and carry out all the development related to that feature or that fix in that branch. Once the developer(s) feel that the implementation is complete (including tests), a *merge request* (also known as a *pull request*) should be submitted, soliciting code review from other developers on the team. This code review process is not that different in spirit from academic peer review and ensures the quality of the final product. Code review is an iterative process with reviewers making suggestions and the developer(s) discussing them and, once consensus emerges, implementing them in the branch. Once the reviewers have approved the request, the changes in the feature or bug fix branch are merged into the main repository branch.

Another useful feature provided by Git is the use of 'tags'. Specific points or milestones can be *tagged* with a custom string allowing the state of the codebase to be easily and temporarily reverted to that which existed at the time the tag was created. This is extremely useful for reproducibility, especially combined with released application versions. Say a specific release of the application has been deployed for a client, and the client complains that it is producing inaccurate scores for certain types of essays. The team can easily check out the code *as it exists* in the released version on their own machines – without needing to interact with the deployed version in a production environment – and start debugging to find the cause of the inaccuracy. Once a fix is found, a new release can be created – along with an accompanying git tag – and deployed into production.[6]

Although Git is frequently employed to version code used for training, evaluating, and analyzing scoring models, it is not ideal for large data files or resources. These resources can usually be versioned using more complex techniques such as the Large File Storage (LFS) extensions for Git. The various machine learning models themselves can be versioned and tracked by using similar technologies or in fully managed *model stores* provided by comprehensive off-the-shelf solutions such as DVC,[7] Neptune.ai,[8] or Weights & Biases.[9]

Using version control software and collaborative workflows enabled by such software are critical to developing a modern, responsive, and robust automated scoring application.

3.3 Reproducibility

In the previous section, we described the use of Git tags for ensuring reproducibility. While this is certainly necessary, it may not be sufficient. Modern NLP applications, including automated scoring software, require several underlying machine learning or language processing libraries, among many other dependencies. The developers of these libraries make changes, and different versions of libraries might require different inputs or produce slightly different results as the field evolves.

For example, automated scoring of spoken responses relies on automated speech recognition (ASR) to obtain an automatic transcription of the response. For noisy, hard-to-understand responses, the transcription may depend not only on the version of the ASR software but also on how it was compiled. This, in turn, may lead to small differences in scores computed in different environments. These differences are rarely large enough to have a substantial effect on scores. Yet even small discrepancies may be sufficient to delay or even stop the deployment of automated scoring if the scores used for system evaluation cannot be reproduced in the production environment.

One way to capture the state of the application at any point in time is to capture the dependency versions in a file at the time of any release. Then, whenever a release version is to be restored, the dependencies can be restored along with it.

The disadvantage of this solution is that it does not account for the case in which older versions of dependencies might be entirely unavailable from the usual public channels since they have been deprecated. Automated scoring applications deployed for high-stakes use often continue to be used by clients for years at a time in *legacy mode* without any major changes. If older versions of dependencies can no longer be obtained from public channels, tasks such as adding a new scoring model to a legacy scoring application or debugging any issues arising with such an application become impossible.

Another common solution to ensure reproducibility is to freeze and capture the code of a running application along with the code of *all* its dependencies into an 'image'. This image can then be easily deployed as a self-contained, lightweight virtual machine as needed, providing a replica of the original environment with the code and its dependencies. These lightweight virtual machines (known as *containers*) are completely isolated from any other containers that may also be running on the same hardware. Modern containerizing solutions such as *Docker*[10] and *AWS Elastic Container Service* (ECS)[11] make this process relatively painless and accessible to a wider audience.

3.4 Code Review

Tests and version control make it possible to track changes to the code and flag updates that lead to changes in the outputs. However, the decision about which changes are appropriate ultimately lies with a group of human experts through a process known as *code review*.

Code review, as mentioned earlier, is an iterative process whereby the author(s) of the proposed code changes work with a set of reviewers – other team members intimately familiar with the codebase – to get the changes into a state where both parties feel confident that the changes add value to the codebase *without* introducing any bugs or unexpected changes.

Given the tension between adding new features (or fixing bugs) at a reasonably fast rate and the increased likelihood of introducing bugs if reviewers are rushed, getting the code review process right requires empathy on the part of both the code author(s) and the reviewers and the need to treat it as a cooperative exercise rather than an adversarial one. The code review process tends to have a major impact on code quality (McIntosh et al., 2014; Kononenko et al., 2016), and automated scoring applications are no different in this regard. Each team needs to discover and implement its own version of the code review process that works for its members. However, we recommend making use of appropriate tools (some provided by the version control software itself) to improve the process, e.g., auto-suggestion of appropriate reviewers (based on the files where the proposed changes are being made), threaded discussions, and conversion of agreed-upon comments into merge-blocking tasks are some ways in which modern code reviews can be made more 'lightweight' (Sadowski et al., 2018).

While a review of the code powering a new feature is certainly important, it may be equally important for additional evidence to be provided and examined as part of the code review process – for example, an explanation of how the feature works and how it was developed, and a detailed error analysis on an internal or external benchmark dataset.

4. Transparency and Documentation

Understanding how the scores are computed is important for supporting their validity as well as for ensuring user trust. From NLP scientists who might need to understand a particular part of the scoring pipeline, to business units who communicate the scoring process to test-takers and other score users, the different stakeholders we identified in Section 2 have different needs when it comes to documenting the score computation process: there is no one size fits all.

Comprehensive documentation is also instrumental in reducing single-point failures. An important tenet of team-based software development is that any one member of the team should be *valuable* but never *critical*. The underlying message, of course, is that there should be no *single* team member who is the only person to own or understand any one part of the application or infrastructure, since that can be a significant bottleneck and liability to the entire team and ultimately affect the accuracy and validity of automated scores.

4.1 Codebase Readability

No external documentation explaining the logic of the score computation can compensate for a poorly documented codebase. Over time, documentation might get out of date and no longer reflect the current state of the engine. In extreme cases, the documentation might not reflect the actual implementation. For example, it might describe the original research version of a feature that has subsequently been revised when incorporated into the engine. Instead of relying on external documentation explaining the computational logic, the code for the automated scoring engine itself should contain sufficient information to allow scientists, engineers, and anybody looking at the codebase to understand the 'why', the 'what', and the 'how'. This is achieved by using self-explanatory variable names, opting for implementation that increases the readability of the code, and adding multiple comments throughout the codebase.

The logic of computation should be accessible not only to NLP scientists and engineers but also to other stakeholders such as psychometricians or data scientists who might want to review the algorithm for computing certain features. The specific programming language in which the code is written can vary by team. It is not uncommon for cross-functional teams to use multiple languages for different parts of the pipeline, e.g., Python for NLP/ML and R for psychometric and educational measurement analyses. A well-documented and readable codebase should make it possible for stakeholders to understand the logic of computation even if they are not familiar with a particular programming language.

4.2 Stand-Alone Technical Documentation

The comprehensive documentation should include not only detailed comments in the codebase but also stand-alone documentation describing the architecture and the detailed working of the application. In many cases, some technical documentation can be automatically generated from the comments in the codebase, e.g., docstrings for Python functions and classes. However, technical documentation should also include tutorials and walkthroughs to ensure that new users and developers are able to orient themselves. Additionally, the documentation *must* also make it easy for these new users to contribute to the project, by including: (a) instructions for setting up a development environment; (b) best practices for writing tests; and (c) adding to the documentation itself.

We also recommend formalizing a release process consisting of specific actions that must be taken to produce a new release – a new version of the automated scoring system deployed for the users – and including this process document in the documentation. This makes it easy for any developer to create a new release and makes the process transparent to the users. One important action that must be part of the release process is the creation of detailed *release notes*, a document explaining everything in the system that has changed since the previous release. This document should clearly describe the different types of changes contained in the release – new features, bug fixes, and backward-incompatible changes, if any. Each change should ideally be linked to the corresponding issue and pull request, providing the full context and discussion for the change to the interested user. Release notes are essential for allowing the stakeholders to plan any changes that may be required in downstream workflows or systems if they decide to use the newly released version. They may also decide to keep using the older release, assuming it continues to be available.

4.3 Nontechnical Documentation

Domain experts, product owners, and business representatives are likely to require more general documentation structured as a general-purpose memo and written without too much technical detail. When such memos are written and shared as separate documents, they eventually become outdated. Our solution to this problem is to keep the documentation as part of the main codebase under a separate subdirectory. This documentation includes both technical and general sections. Technical sections contain detailed information about specific functionality and are aimed at NLP scientists and engineers as well as other stakeholders interested in technical details. General sections are written for a nontechnical audience and provide a broader overview of the functionality. To ensure that the documentation stays up to date, any new functionality proposed for the code must include a documentation component that is reviewed for accuracy as well as readability during the code review process.

4.4 Open-Sourcing the Code

Making the code and the models available for inspection to all stakeholders, including test-takers, is the ultimate way to ensure transparency and fairness. Many major software projects have adopted the well-established 'open-source' model where the code is released publicly, and anyone can review the code, change it, or contribute to it. Open-sourcing software can drive innovation and, in some cases, make the software more reliable. Open-source software has been used for many educational applications, including learning management systems, where it offers flexibility and promotes equal access not hindered by substantial licensing fees (for a review of what was available at the time, see Lakhan and Jhunjhunwala, 2008).

Open-sourcing the code for automated scoring engines may have several disadvantages, however. Automated scoring engines do not score the responses in the same way as human raters do; even the most modern engines do not offer full construct coverage. Access to the scoring logic can make it possible to reverse-engineer a strategy that would result in a higher score than would be appropriate given test-taker skills. As a result, publicly available code for automated scoring engines may make it easier to game the system, thus reducing the validity of the automated scores. It may also encourage undesirable washback, where teachers and test-takers would focus on improving the skills that are currently covered by the automated scoring engine. Finally, the models used in automated scoring are usually trained using existing test-taker data. Recent studies (e.g., Dwork et al., 2017) show that in some cases, it may be possible to reconstruct substantial amounts of personal information about an individual by cross-referencing multiple public datasets and models. Publicly releasing models for automated scoring without appropriate safeguards might violate test-takers' privacy, especially as the scoring engines adopt more complex 'black-box' models, or if models are trained using personally identifiable information such as voice or video recordings.

While the engines themselves might remain proprietary, there are many reasons to open-source other components of the automated scoring ecosystem. These could increase trust in the system as well as encourage knowledge sharing across the educational technology community. For example, at ETS we have open-sourced multiple tools for training and evaluating models for automated scoring. These include SKLL[12] (Scikit-Learn Laboratory) for running machine learning experiments efficiently, RSMTool[13] for comprehensive evaluation of automated scoring models (Madnani et al., 2017; Madnani & Loukina, 2020), RSTFinder for identifying discourse structure, and several other tools. Some of these tools, such as RSMTool, have been developed in close collaboration with psychometricians and allow wider community access to methodologies developed in the educational measurement community that may not be commonly known to NLP scientists. They also allow any interested party to inspect the exact algorithms for the metrics used for evaluating automated scoring engines.

5. Example Workflow

At ETS, we have adopted the following workflow for automated scoring software development. We present it here as an example workflow that has been successfully implemented in a commercial product.

a. All code lives under version control, including the code used for model training and evaluation.
b. Unit and functional tests are written for all software with the aim to have as much of the codebase covered by the tests as possible.
c. Any proposed changes to software are *never* made directly in the main branch of the code. Instead, a new branch of the code is created, and the proposed changes are then reviewed by one or more stakeholders via a merge request.
d. The stakeholders examine the proposed changes not only for programmatic efficiency, but also for any potential negative impacts on accuracy and validity.
e. To aid code review, all tests are usually run automatically, and their results are made available to the reviewers. Code reviews do not even start unless all the tests are passing.
f. All merge requests *must* be accompanied by new or updated tests as well as updates to the documentation describing the changes in detail, if appropriate.
g. Any proposed changes can be merged into the main code *if and only if* the reviewers explicitly approve the merge request.
h. Once the changes are merged, the full suite of tests is automatically run again via a continuous integration (CI) plan to ensure application health.
i. All releases are tagged in the code repository; a corresponding container image artifact is automatically generated by a continuous deployment plan for any release tag. This container can be easily instantiated anytime this specific release needs to be used for any purpose.
j. These container images are then used by the DevOps (or IT) division to deploy the new release in a production environment for use by the stakeholders.
k. After the deployment, the deployed model needs to be continuously monitored so that any issues that may arise (data drifts, unexpected prediction errors, etc.) can be shared with the stakeholders. Depending on the nature or severity of issues, the relevant stakeholder determines a plan of action.

6. Conclusion

A lot of previous literature on automated scoring focused on statistical, psychometric, and validity aspects of the output of the software (both the scores and extracted features).

In this chapter, we discussed the role of software *robustness* as another important dimension in the field of automated scoring. We outlined the best practices that we follow at ETS to ensure that the scores produced by our scoring engines remain accurate and valid throughout the engine development process. We also described our approach to introducing and documenting changes and highlighted how this process needs to accommodate the often-conflicting needs of different stakeholders. Finally, we touched upon the advantages and disadvantages of open-sourcing the code for automated scoring.

Since the world of software is rapidly changing, not all the practices, examples, or tools that we mention might be applicable several years from now. However, we firmly believe that the main principles we outlined will persist. Change tracking, continuous and comprehensive testing, and documentation are three cornerstones of reliable and robust automated scoring software, and constitute prerequisites for accurate, valid, and fair automated scores.

Notes

1 There are several other aspects of software development, such as requirements gathering and specification design, that might be equally important but are considered out of scope for this chapter.

2 Here we use the very broad reading of the term *machine learning* to include any kind of algorithm that can make a prediction given some input (e.g., ranging from simple linear regression to deep neural networks and beyond).

3 Note that machine learning–based applications are typically considered 'back-end' applications, meaning that a user does not directly interact with the system. Usually additional types of software, such as computer interfaces and/or visual components, take care of sending information to and from the backend. We do not consider those kinds of software here.

4 https://github.com

5 https://bitbucket.com

6 There can be multiple deployment strategies; for example, the application may be deployed not *directly* to production, but to a testing or staging environment first.

7 https://dvc.org

8 https://neptune.ai

9 https://wandb.ai/

10 https://docker.com

11 https://aws.amazon.com/ecs/

12 https://github.com/EducationalTestingService/skll

13 https://github.com/EducationalTestingService/rsmtool

References

Dwork, C., Smith, A., Steinke, T., & Ullman, J. (2017, December). Exposed! A survey of attacks on private data. *Annual Review of Statistics and Its Application*, *4*, 61–84.

Kononenko, O., Baysal, O., & Godfrey, M. W. (2016). *Code review quality: How developers see it*. Proceedings of the 38th International Conference on Software Engineering.

Lakhan, S. E., & Jhunjhunwala, K. (2008, April–June). Open-source software in education. *Educause Quarterly*, *31*(2).

Lottridge, S., & Hoefer, N. (2020). System architecture design for scoring and delivery. In D. Yan, A. A. Rupp, & P. Foltz (Eds.), *Handbook of automated scoring: Theory into practice* (pp. 195–216). Chapman and Hall.

Madnani, N., & Cahill, A. (2018). *Automated scoring: Beyond natural language processing* (pp. 1099–1109). Proceedings of the 27th International Conference on Computational Linguistics.

Madnani, N., & Loukina, A. (2020). *User-centered & robust NLP OSS: Lessons learned from developing & maintaining RSMTool* (pp. 141–146). Proceedings of 2nd Workshop for Natural Language Processing Open Source Software (NLP-OSS).

Madnani, N., Loukina, A., von Davier, A., Burstein, J., & Cahill, A. (2017). *Building better open-source tools to support fairness in automated scoring* (pp. 41–52). Proceedings of the First Workshop on Ethics in Natural Language Processing.

McIntosh, S., Kamei, Y., Adams, B., & Hassan, A. E. (2014). *The impact of code review coverage and code review participation on software quality: A case study of the qt, VTK, and ITK projects*. Proceedings of the 11th Working Conference on Mining Software Repositories.

Sadowski, C., Söderberg, E., Church, L., Sipko, M., & Bacchelli, A. (2018). *Modern code review: A case study at Google*. Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice.

Schneider, C., & Boyer, M. (2020). Design and implementation for automated scoring systems. In D. Yan, A. A. Rupp, & P. Foltz (Eds.), *Handbook of automated scoring: Theory into practice* (pp. 217–240). Chapman and Hall.

Shaw, D., Bolender, B., & Meisner, R. (2020). Quality control for automated scoring in large-scale assessment. In D. Yan, A. A. Rupp, & P. Foltz (Eds.), *Handbook of automated scoring: Theory into practice* (pp. 241–262). Chapman and Hall.

Spolsky, J. (2004). The Joel test: 12 Steps to better code. In *Joel on Software*. Apress. https://doi.org/10.1007/978-1-4302-0753-5_3

Williamson, D. M., Xi, X., & Breyer, F. J. (2012). A framework for evaluation and use of automated scoring. *Educational Measurement: Issues and Practice*, *31*, 2–13.

2

Psychometric Considerations When Using Deep Learning for Automated Scoring

Susan Lottridge, Chris Ormerod, and Amir Jafari

1. Introduction

Automated scoring refers to the use of statistical and computational linguistic methods to assign scores or labels to examinee responses to unconstrained open-ended test items. Automated scoring has been widely adopted in K-12 assessment, licensure, and certification programs primarily in writing, reading, and math proficiency assessment and is arguably the most recognized application of machine learning in education measurement (Foltz et al., 2020). Since the 1990s, research has been conducted on automated scoring in many assessment domains, including essay scoring (Shermis, 2014; Shermis & Burstein, 2006), short answer scoring (Burrows et al., 2015; Cahill et al., 2020; Liu et al., 2014; NCES, 2022; Ormerod et al., 2022; Riordan et al., 2020; Sakaguchi et al., 2015), mathematical equations (Fife, 2013), and spoken responses (Bernstein et al., 2000; Chevalier, 2007; Xi et al., 2008). Automated scoring can also be used to identify and extract elements from responses (e.g., relevant clinical concepts) to be used by downstream systems for scoring (Sarker et al., 2019).

Incorporating automated scoring into assessment programs offers many benefits including cost savings, faster scoring, improved score consistency within and across administrations, and potentially higher-quality scores when combined with human scoring (Foltz et al., 2020). Most automated scoring engines use a classical approach whereby features (e.g., grammatical errors, source citation) are expertly crafted and statistical models are used to predict scores using extracted feature values (Cahill & Evanini, 2020). Deep learning engines learn features alongside the predictive model using large, multilayered neural networks, often with millions of parameters (Ghosh et al., 2020; Matthias & Bhattacharyya, 2020; Ormerod et al., 2022; Riordan et al., 2020; Rodriguez et al., 2019; Taghipour & Ng, 2016). Deep learning engines offer the possibility of producing end-to-end scoring systems without the need for explicitly designed features, typically using models trained on large corpora intended to represent language.

In this chapter, we provide an overview of automated scoring engines and methods, describe deep learning in greater detail and how it can be used in automated scoring, and then discuss psychometric challenges in using deep learning in automated scoring. The high-level overview of automated scoring engine design and use is intended to help orient the reader to

DOI: 10.4324/9781003278658-3

This chapter has been made available under a CC-BY-NC-ND 4.0 license.

methodological details of automated scoring systems and their validation. Four psychometric challenges in using deep learning are then presented along with suggested mitigations.

2. High-Level Overview of Automated Scoring Engines

Automated scoring engines are statistical in nature; they model human-assigned scores or labels. A common pipeline in automated scoring appears in Figure 2.1. In this pipeline, responses are preprocessed to standardize their format, numeric features are extracted, and features are entered into a statistical model to optimally predict a target score or label.

Response preprocessing is often conducted in two places. Responses are initially preprocessed minimally (e.g., removing formatting tags, standardizing white space and punctuation, and tokenization into morphemes, words, sentences, or paragraphs) before the feature extraction phase. Responses are also processed at the feature extraction phase to further improve the quality of specific feature extraction algorithms. Examples include spell correction, cleansing of unusual punctuation, and conversion to lower case.

In the feature extraction phase, linguistic features designed to align with the intended application of the rubric are extracted. In classical automated systems, the features can be carefully engineered to model elements of speech or writing (Cahill & Evanini, 2020), can consist of many low-level proxies of language such as part of speech or individual word tokens (Woods et al., 2017), and/or can leverage unsupervised methods to create statistically derived features such as those from latent semantic analysis (LSA; Deerwester et al., 1990) or latent Dirichlet allocation (Blei et al., 2003).

Statistical modeling applied to the extracted features includes multivariate linear and logistic regression, support vector machines, and tree-based methods, among others. These same methods can be applied to multiple automated scoring models using a process called ensembling (Zhou et al., 2002).

3. Automated Scoring Processes

The high-level flow for training and evaluating automated scoring models has six steps (Figure 2.2). First, responses that are representative of the population and the conditions of testing are collected. Hand scores are then obtained using high-quality procedures such as the

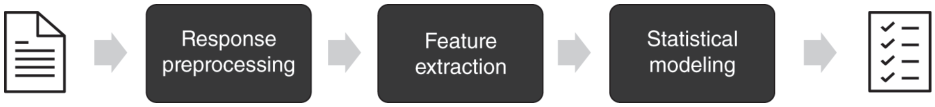


Figure 2.1 Automated scoring pipeline

Source: Adapted from Lottridge, Burkhardt, et al. (2020).

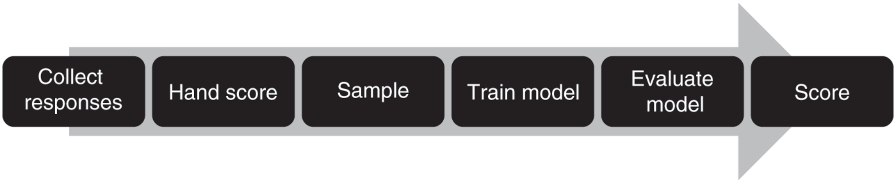


Figure 2.2 Automated scoring high level flow

Source: Adapted from Lottridge, Burkhardt, et al. (2020).

use of clear and well-defined rubrics, well-trained raters, scoring designs using at least two independent reads, careful monitoring of the scoring process, and well-defined performance criteria (Wolfe, 2020). Hand-scored data are then sub-sampled into a minimum of two sets: train and held-out validation. The train data can be further divided into sub-samples using single- or *k*-fold cross validation or bootstrapping (James et al., 2017). These sub-samples are used to build competing models and to select the best-performing model. Because automated scoring engines use large numbers of parameters, they are particularly susceptible to overfitting the train data even when using regularization procedures (Srivastava et al., 2014). For this reason, the model performance is evaluated on a held-out validation dataset, which is scored using the final chosen model. The results on this sample are assumed to generalize when scoring responses from similar populations and testing conditions. Once the performance of the model has been validated, the model is packaged and deployed for scoring.

Evaluation metrics for picking the best-performing model and evaluating the final model performance examine the distributional and agreement characteristics of the engine compared to humans, using human scores as the ‘gold standard’. Most evaluations use a combination of criteria outlined in Williamson et al. (2012), among other metrics. The Williamson et al. metrics include quadratic weighted kappa (QWK) and the standardized mean difference (SMD). Thresholds can be absolute (e.g., engine–human QWK must exceed .70) or relative (e.g., engine–human QWK can be no lower than 0.1 of the human–human QWK). Other metrics include the proportion reduction of mean squared errors (PRMSE: Yao et al., 2019), the ratio of standard deviations of the engine to the human (Wang & von Davier, 2014), and the Kullback-Leibler divergence entropy metric (Kullback & Leibler, 1951). Metrics can be computed in the aggregate and by subgroup.

4. Validating Automated Scoring Models

The validation of automated scoring extends beyond the evaluation methods described to this point. A core concern in validation is the interpretation and use of scores (AERA et al., 2014). In automated scoring, validation considers how scores are arrived at, how they align to the rubric, how they relate to other measures, and how automated scores influence the combination of scores from other items, such as test scores. The comparability of automated scores and human scores is often a core concern because automated scores often replace or supplement human scores (Yan & Bridgeman, 2020). In this situation, validation approaches focus on three key elements: construct validity; item-level comparability; and test-level comparability (Lottridge, Burkhardt, et al., 2020). Construct validity evaluations examine engine designs relative to the task and rubric and consider methods for detecting unusual, aberrant, or gaming responses (Bejar et al., 2014; Filighera et al., 2020; Lottridge, Godek, et al., 2020; Rupp, 2018). Item-level comparability evaluations include comparing the relationships of automated and human scores to other measures, such as response length and test scores. Test-level comparability evaluations include comparing engine and human scores in typical psychometric analyses such as item-total correlations, test reliabilities, item parameter estimates, and test-level correlations with other measures (Nicewander et al., 2015; Wang, 2021).

The key criticism of automated essay scoring is that engines do not understand language and can be ‘tricked’ into giving high scores (Page, 2003; Shermis & Lottridge, 2019; Wood, 2020). Filters play an important role in identifying responses that either do not merit rubric-based scores, are written to artificially inflate engine scores, or require human intervention. Engines have been found to be susceptible to these responses, but the impact of such responses varies by item (Burkhardt & Lottridge, 2013; Zhang et al., 2016) and engine feature set (Higgins & Heilman, 2014). Both classical and deep learning engines have been found to be susceptible (Lottridge, Godek, et al., 2020). Examples of gaming behavior include: duplication of text to increase length (Higgins & Heilman, 2014; Lochbaum et al., 2013; Zhang et al., 2016); extensive use of prompt text (Lochbaum et al., 2013; Zhang et al., 2016); the inclusion of key topic-related words in an otherwise off-topic essay (Higgins & Heilman, 2014; Kolowich, 2014; Lochbaum et al., 2013; Zhang et al., 2016); and off-topic essays, including those written to other prompts, pre-written and memorized essays, or those with original writing but not addressing the prompt (Burkhardt & Lottridge, 2013; Higgins et al., 2005; Zhang et al., 2016).

Finally, fairness is a key validity concern in automated scoring because of known issues with fairness in machine learning more generally (Corbett-Davies & Goel, 2018; Hutchinson & Mitchell, 2019). Most published automated scoring fairness evaluations have been conducted on ETS' e-rater engine and have a focus on international and nonnative English speakers (Burstein & Chodorow, 1999; Bridgeman et al., 2009, 2012; Ramineni & Williamson, 2018). Recent fairness research on U.S. K-12 students was conducted by Gregg et al. (2021) for English language learners (ELL) and by Lottridge and Young (2022) for race/ethnicity, gender, ELL status, and economic status. These two investigations focused on deep learning automated scoring engines. Williamson et al. proposed examining fairness by flagging items in which the absolute standardized mean difference between the model and human scores within a subgroup (e.g., females) exceeds .10. Lottridge and Young (2022) examined bias with and without controlling for examinee ability and obtained different results using the two approaches. If differences are identified, then it is important to investigate the source of those differences, by examining feature differences (Ramineni & Williamson, 2018) or how responses are represented throughout the scoring pipeline (Gregg et al., 2021). Bias at the feature level can be mitigated by removing features that display bias (Madnani et al., 2017; Shermis et al., 2017).

5. Deep Learning Automated Scoring Models

Deep learning methods model language using complex designs – multilayered neural networks – that consider word use in context and focus attention on word patterns optimally related to a prediction task. Currently, the most successful neural networks are initially trained on very large corpora to form a pretrained (language) model. Pretrained models can be used for a variety of predictive tasks, including score or label prediction. While the details of the designs are rapidly changing, the designs typically involve an over-specified (i.e., million parameter) pretrained model that is subjected to regularization during training using the random removal of parameter dependencies to address overfitting. Innovation in deep learning models has been aided by the open-source, community-driven repository of large, pretrained language models (Wolf et al., 2019) and publicly available text resources such as Wikipedia data (Merity et al., 2016), the Penn Treebank (Marcus et al., 1993), and the one-billion-word corpus from Google (Chelba et al., 2013).

Automated scoring by neural networks typically involves preprocessing a response, mapping the preprocessed response to a vocabulary and semantic space (called an embedding), applying a deep neural network that maps the sequence of embedding values to a vector, and then using classification or regression to produce a score from that vector (Figure 2.3).

After preprocessing, a response is tokenized using the embedding tokenization scheme, typically using characters, words, or subwords (i.e., components of words) as tokens. Embeddings

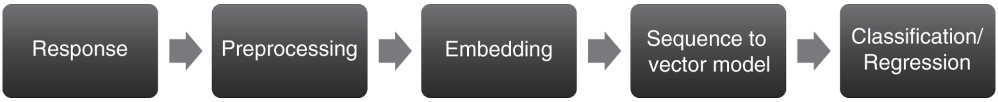


Figure 2.3 A high-level design of a neural network–based automated scoring engine

using character-level tokenization are often robust to spelling errors but make it difficult to ascertain semantic-level information. Word-level tokenization retains semantic information but can suffer from the effects of word sparsity (Guthrie et al., 2006). Embeddings using subwords retain some of the robustness of character-level tokenization and some ability to model semantics without the drawbacks of sparsity (Sennrich et al., 2015). As an example, the word 'decompose' can be divided into three subword tokens: 'de', 'com', and 'pose'. In current systems, this decomposition is based upon subword frequency using a method called byte pair encoding (Gage, 1994).

The embedding space reflects the contextual relationship between tokens in a vocabulary and allows for the vocabulary to be represented as a lower-dimensional (e.g., 100–300) space. Tokens that appear in similar contexts tend to have high cosine similarities in the embedding space. Embeddings are trained for a specific task, such as predicting missing tokens or predicting the next token in some defined window. Embedding spaces can be optimized for a particular task based upon the order *and* content of surrounding tokens or based on the content of tokens alone like the skip-gram or continuous bag-of-words methods (Mikolov et al., 2013). Several embeddings exist (e.g., Word2Vec, BERT, ELMo, GPT-3) and are characterized by the data on which they were trained, their tokenization scheme and resulting vocabulary, how they model word order, their model architecture, and their training task.

Once a response is represented in the embedding space, that representation serves as input into neural network layers that extract key features of the text. The first neural networks used to score essays (Taghipour & Ng, 2016) used two types of layers; convolutional layers, which consider finite collections of words as features, and recurrent layers, which model sequences. The main types of recurrent layers are long-short-term-memory (LSTM) units (Hochreiter, 1997) and gated recurrent units (GRU) (Chung et al., 2014). In both approaches, word order is critically important. While convolutional and recurrent units dominated deep learning research for a long time, the use of attention has had a profound effect on neural network design in recent years. Attention is a trainable weighting scheme that reflects the importance of a particular token or output (Graves et al., 2014). Attention applied to recurrent neural networks has been responsible for many accuracy gains in natural language processing (NLP) tasks. Self-attention, where the output of an attention mechanism is used as input into another attention mechanism, led to the development of the transformer model (Vaswani et al., 2017) and its utilization in the Bidirectional Elementary Representation by Transformers model (Devlin et al., 2018), known as BERT.

The transformer models and their variations started a revolution in NLP by producing state-of-the-art performance on the GLUE benchmarks (Wang et al., 2018), which are a collection language understanding tasks such as grammar, sentiment, paraphrasing, and semantic similarity. Like recurrent networks, transformer models are a class of pretrained models with an architecture that captures language more generally. Pretrained models are trained on a specific task such as predicting the likely next token (Mikolov et al., 2010), or predicting masked tokens to identify the 'missing' word in context (Devlin et al., 2018). These pretrained models can be fine-tuned to perform well on other tasks, such as identifying sentiment, classifying tokens (e.g., named entity recognition), and, of course, automated scoring. This process replaces the layer that predicts the missing or next word with a classification or regression layer. The pretrained model and classification/regression layer are fine-tuned simultaneously to optimize performance on the new predictive task. In fine-tuning, the weights learned in pretraining are used as a starting point for training the model for the more specific classification task. The output of the neural network model is a weighted linear combination of the features, akin to linear regression or classification.

Transfer learning has been a key factor in the success of deep learning, which is the ability to leverage vast corpora of text trained on one task and apply it to another. In Figure 2.4, we see

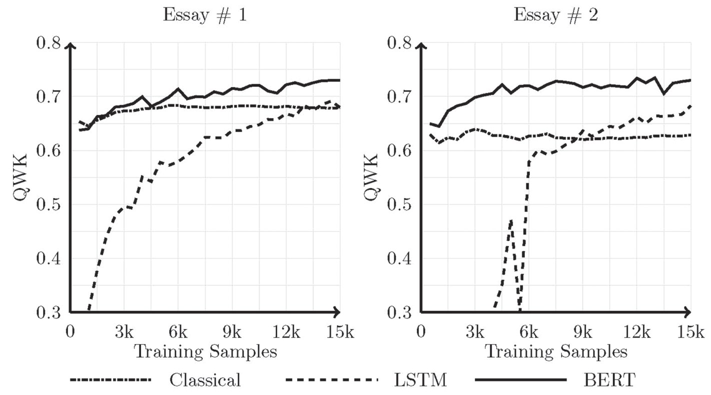


Figure 2.4 Comparison of QWK performance on two essays for Classical, LSTM, and Transformer (BERT)-based models across increasing training sample sizes

Table 2.1 QWK Results from Various Deep Learning Models on the Kaggle Essays, by Prompt

	Essay Prompt								
Scoring Method	**1**	**2***	**3**	**4**	**5**	**6**	**7**	**8**	**Avg.**
Human baseline	0.721	0.814	0.769	0.851	0.753	0.776	0.721	0.629	0.741
Classical model (EASE)	0.781	0.621	0.630	0.749	0.782	0.771	0.727	0.534	0.699
LSTM	0.775	**0.687**	0.683	0.801	0.806	0.805	0.805	0.594	0.746
LSTM+CNN	0.821	0.688	0.694	0.805	0.807	**0.819**	0.808	0.644	0.761
LSTM+CNN+ Attention	0.822	0.682	0.672	0.814	0.803	0.811	0.801	**0.705**	0.764
BERT	0.792	0.680	0.715	0.801	0.806	0.805	0.785	0.596	0.758
BERT+Features	**0.852**	0.651	**0.804**	**0.888**	**0.885**	0.817	**0.864**	0.645	**0.801**

Note: The EASE, LSTM and CNN+LSTM results can be found in Taghipour and Ng (2016). The LSTM+CNN with attention results are found in Dong et al. (2017). The results for BERT appear in Rodriguez et al. (2019), and the results for BERT with features appear in Uto et al. (2020). The best engine QWK is highlighted in bold.

* Only one of the two traits were analyzed for Item 2: Writing Applications.

how the QWK agreement between automated scoring and humans in one dimension of writing (Organization, scored 1, 2, 3, and 4) improves when the training sample increases for two items, one in grade 6 and one in grade 7, across three models. The LSTM and a classical model (a combination of LSA and writing features) were trained from scratch, while the BERT model was a fine-tuned pretrained model. The classical model does not improve with more data. The BERT model shows similar performance for samples of size 1,500 and then gradually improves over the classical model as sample size increases. The LSTM model performs poorly on small samples and does not approach the BERT model performance even at 15,000 samples.

Various deep learning configurations have been examined on the Automated Student Assessment Prize (ASAP) dataset on Kaggle, considered a standard benchmark (Shermis, 2014). This dataset consists of eight essay prompts from various K-12 assessment programs that vary in their use of stimuli, rubric, and type of score. Table 2.1 provides a summary of the published efforts to date. The classical model uses an open-source engine, called Enhanced AI Scoring Engine (EASE).[1] Results represent the average performance of models across a fivefold cross-validation design defined in Taghipour and Ng (2016) on the publicly available data, with each item having approximately 1,800 responses. The results suggest that deep learning approaches approximate or exceed human performance and classical performance, and that the best-performing models combine classical methods and deep learning methods.

6. Psychometric Challenges With Using Deep Learning Models

Deep learning engines have many psychometric challenges beyond performance on a held-out validation sample. The core of these challenges centers on the complexity of the models in terms of their size (i.e., number of parameters) and architecture, the need for specialized hardware and time for training and scoring (Mayfield & Black, 2020), the reliance on pre-trained open-source models that can be very difficult to train, and their highly empirical focus (Church & Liberman, 2021). Another challenge is their relatively recent introduction into automated scoring; little is known about how these models will work in practice, particularly in live scoring settings, and how robust they will be to adversarial examinee behaviors. In this section, we discuss four psychometric challenges that practitioners will face when using deep learning models in automated scoring.

6.1 Challenge #1: Explainability

The complexity of deep learning engines makes it difficult to explain how they produce scores. Explainability is important for many reasons, including the need to inform a validity argument around how scores are generated, to explain how a score is produced for one or more responses, to examine the source of score differences between sets of examinees (e.g., identifying source of bias), and to audit and debug scores. The feature values and statistical models from classical engines are typically interpretable because they are explicitly derived and have relatively few parameters. Deep learning engines, with implicitly defined features and millions of parameters, are not directly interpretable. That said, the explainability of deep learning engines can be investigated, albeit using different methods. These methods include associating elements of the response (e.g., words, sentences) with the predicted score and/or examining the data as it is processed through the flow outlined in Figure 2.3. We describe the approaches in two use cases.

In the first use case, elements of the response – such as words – are associated with the engine-produced score. In a deep learning context, these approaches are conducted post hoc – that is, the explanation is computed after scores are produced. One approach is to empirically examine the impact of the removal of a word or sentence on the predicted score. Two implementations of this approach are Local Interpretable Model–Agnostic Explanations, or LIME (Ribeiro et al., 2016), and Shapley Additive exPlanations, or SHAP (Lundberg & Lee, 2017). Both are model-agnostic or 'black box' methods that depend only on model outputs. Using either method, a sample of response perturbations are generated by removing one or more tokens, and then the model-predicted values associated with each perturbation are collected. Both methods then use statistical techniques to estimate the impact of the inputs (e.g., words) on score prediction. SHAP estimates how the inclusion of a word across all possible subsets of words impacts score, whereas LIME models the score prediction using a simple surrogate model, such as linear regression. The magnitude and the direction (i.e., positive or negative) of the values for each word reflect the influence of that word on the engine-predicted score.

Another approach is to compute the gradient associated with each input token compared to some baseline token (e.g., blank token), using the difference in the predicted probabilities and the difference of the token values. The gradient values are interpreted as a reflection of the importance of the token on the prediction. One such approach is integrated gradients (Sundararajan et al., 2017). As with the LIME and SHAP approaches, the gradient values have both a

Table 2.2 Average Inter-Annotator Agreement on Annotations Associated with Crisis Alerts

Comparison	Exact Agreement	Cohen's kappa
Human Annotator 1–Human Annotator 2	.88	.71
LIME–Human Annotator	.68	.22
Integrated Gradients–Human Annotator	.76	.37

magnitude and a direction. Additionally, thresholds applied to the values can annotate which words are associated with the engine score.

The outputs of these approaches need to be verified against human annotation. A recent study compared human annotations against LIME and integrated gradient annotations in explaining crisis alerts (Lottridge et al., 2021). The LIME method agreed less with human annotators compared to the integrated gradient methods, and both explainability methods agreed less than two human annotators (Table 2.2). Such work is at an early stage, however, and we expect that engine annotation methods will better match human annotations as methods improve.

In the second use case, the representation of responses as they flow through the engine can be examined. If bias is identified, for example, the representations can be examined for different subgroups. Differences can be examined in preprocessing outputs, in what tokens are and are not mapped to the embedding, where responses appear in the embedding space, and in the final weights estimated in the last layer before scores are predicted (Gregg et al., 2021). Additionally, characteristics of responses, such as the number and nature of misspellings, number and types of grammatical errors, or language styles, can be also identified and then examined for how they are processed throughout the engine and how they are scored by the engine.

The aforementioned methods represent early-stage attempts to better address explainability of deep learning methods. Explainability methods are a fertile research area as the field broadens its focus from prediction to explanation. We expect that more tools will become available to better understand these engines, particularly by inspecting aspects of the architecture related to language or by reducing the complexity of the network (Jawahar et al., 2019; Kovaleva et al., 2019). Finally, another solution to the explainability problem is to build features using deep learning methods that are then used in the classical framework; such an approach could also potentially improve accuracy for some features.

6.2 Challenge #2: Use of Pretrained Models

Current deep learning systems rely heavily on pretrained models; however, the pretrained model methods and architectures and their subsequent impact on automated scoring have not been examined in depth. We characterize these methods in terms of the vocabulary defined by pretrained models, the response length limitations, and the discrepancy between the language used in pretraining and the automated scoring task.

First, while all automated scoring models use a vocabulary – or list of tokens – in some form, the classical approaches tend to use vocabularies that are closely tied to the language of examinee responses or topic area. In contrast, pretrained models tend to be trained on publicly available text resources such as Wikipedia or Books Corpus, which tend to use very formal and general language. The language in the corpora can differ from examinee writing, in terms of topic, language patterns, and grammatical quality. As a result, examinee words may not be represented in the vocabulary (i.e., are 'out of vocabulary') and/or the encoded language model may not be able to accurately represent certain topics or styles of writing. Additionally, pretrained models may have only uncased versions, meaning that words are converted to a single case (i.e., upper or lower), which may not be appropriate for some items. The use of subwords and fine-tuning can help offset these limitations, but it is yet unclear how the choice of pretrained model can impact the quality of score prediction.

As mentioned earlier, pretrained models often use subwords to avoid the 'out of vocabulary' problem. It is important to analyze how many words are unmatched (and thus divided into subwords) and which words are unmatched. As an example, Table 2.3 presents the typical number of words not matched to the BERT vocabulary for the Kaggle responses. Across items, the average ranges from 5 to 11.9 words for a typical response (Table 2.3). Note that the conversion of unmatched words to subwords also increases the length (now in BERT tokens) of the responses. This increase in tokens can also impact scoring, as we discuss next.

Most pretrained language models have limits on token length (Wolf et al., 2019), which can pose problems for modeling substantive essays written by older children and adults. The limit, often 512 tokens, was imposed for architectural and computational reasons, and to align with standard GLUE task requirements that typically focus on sentence-level classification. Architecturally, the inputs to most neural networks, including the transformer and convolutional neural networks, require fixed-length input. LSTM networks do not require fixed-length input but do require one to specify the number of tokens to pay attention to. From a computational standpoint, allowing longer input requires larger models. In the case of transformer models, the amount of computing power required to implement the attention mechanism grows quadratically with the length (Vaswani et al., 2017). Table 2.4 illustrates the fact that the number of responses that exceed the maximum token length can vary dramatically by prompt; no responses exceed the threshold for two items, and a substantial number of responses exceed the threshold for three items.

Table 2.3 Average of Words, BERT Subword Tokens and Unmatched (i.e., Converted to Subwords) Words in Kaggle Essays, by Prompt

	Essay Prompt							
Statistic	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Count	1783	1800	1726	1772	1805	1800	1569	723
Ave. N. Words	410.3	430.2	123.4	105.5	138.8	171.2	191.5	681.1
Ave. N. BERT Tokens	436.8	441.6	127.4	110.4	147.4	188.4	210.3	727.5
Ave. N. Unmatched Words	11.9	11.4	3.8	5.1	5.0	9.7	5.8	10.4

Table 2.4 Distributional Characteristics of the BERT Token Length for Kaggle Essays, by Prompt

	Essay Prompt							
Statistic	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Count	1783	1800	1726	1772	1805	1800	1569	723
Min	10	35	10	4	5	4	6	5
25%	362.5	341	81	64	99	149	136	582.5
50%	466	451.5	124	106	152	199	204	806
75%	572.5	583	182.75	157.25	206	244	288	1018
Max	1182	1282	501	452	541	581	898	1574
> 512	692	651	0	0	2	6	59	587

Dealing with the length problem is not easy. Responses can be truncated, an inappropriate solution in most contexts. Alternatively, responses can be partitioned into multiple overlapping vectors and predictions are then aggregated across the partitions. This, however, means that the partitions are independently predicted, and any relationship among the partitions is lost in the prediction. The only architectures that account for the possibility of arbitrarily long dependencies are recurrent networks such as the LSTM. However, it is difficult to train an LSTM to account for these long-term dependencies on small datasets. More efficient methods for training pretrained models may allow for longer text to be modeled. Efficiencies can come from architectural improvements such as weight sharing (Sun et al., 2020), using different attention mechanisms such as a sliding window (Beltagy et al., 2020), hashing (Kitaev et al., 2020), or dimensionality reduction (Wang et al., 2020), or from training methods (Clark et al., 2020). However, their application to automated scoring is also still in experimental phases (Ormerod et al., 2021).

Finally, bias has been identified in pretrained models and embeddings primarily due to the choice of training data (Bolukbasi et al., 2016a, 2016b). Efforts have focused on mitigating bias statistically (Papakyriakopoulos et al., 2020) or removing biased text (Brunet et al., 2019). Pretrained language models such as BERT are generally less biased than word embeddings such as Word2Vec (Basta et al., 2019). Bias in pretrained models and embeddings is typically investigated by examining whether words are more closely associated with gender, ethnicity, or sexual orientation in ways that may impact downstream interpretation. For example, a female pronoun such as 'she' may be more strongly related to the word 'nurse' than to 'doctor', even when there is no reason to associate gender with an occupation. Relatedly, sentiment language (positive, negative) can be examined by subgroup. It is yet unclear whether bias in the embedding or pretrained language model translates into bias in a fine-tuned model in automated scoring.

6.3 Challenge #3: Training Complexity

A third psychometric challenge in deep learning is training and modeling complexity. Training requires advanced technical expertise to determine the appropriate architecture and parameterizations, to diagnose issues, and to utilize modern graphics processing unit (GPU) computing resources. The skillset required for training can also rely on computer programming and facility with emergent and often poorly documented software libraries. Some training decisions are directly tied to the nature of the task and rubric; however, many decisions are based purely on experience in training these models. These highly empirical tuning decisions are not terribly satisfying for those interested in understanding why a particular set of parameters works over another. As the field matures, we expect that reasons underlying successful versus unsuccessful parameter choices will become clearer and will provide better validation support for deep learning models.

Determining which approach (e.g., transformer, recurrent, convolutional) is appropriate for the modeling problem can be challenging. Once a general approach has been chosen, many decisions remain around parameterization. Convolutional and recurrent approaches come with a bewildering number of architectural choices, including the number, size, and type of layers. Many parameters in transformer models are defined at the pretraining stage, thereby limiting the number of decisions.

Once an architecture is defined, we find that the most important training parameters are the number of epochs, the learning rate, and the batch size. Deep learning networks are optimized using methods derived from stochastic gradient descent, which iteratively estimates the gradient using training batches to find the minimum of a loss function. The number of epochs determines how many times the neural network sees the entire dataset. The learning rate determines the size of the step taken in the optimization method. The batch size determines the number of training samples used to approximate the gradient vector. If the learning rate is too small, a model might never reach an intended optimal value. If the learning rate is too large, the optimization might skip over the optimal value. If the batch size is too small, the approximation of the gradient might not be stable enough. While larger batch sizes are often better, very large batch sizes have also been shown to be problematic because they reduce the number of optimization steps. It is also very difficult to accommodate large batch sizes on most hardware used to tune neural networks; smaller batches (8–12 responses) are typically used.

Hyperparameter tuning packages such as SigOpt (McCourt, 2019), Tune (Liaw et al., 2018), and Optuna (Akiba et al., 2019) are designed to approximate optimal hyperparameters. One may also tune hyperparameters manually by defining sets over which to iterate and selecting the best-performing model from among them. However, defining a sensible hyperparameter set requires expertise and experience.

6.4 Challenge #4: Robustness in Live Scoring

How deep learning models perform during live scoring is not well understood. While both classical and deep learning methods may perform well on a held-out validation sample, it is unclear how robust deep learning models are in live scoring situations. This is because, to our knowledge, deep learning automated scoring engines have only been recently introduced. Theoretically, fine-tuned pretrained models should be more robust than classical models because they utilize large datasets and vocabularies. However, these large models may overfit training datasets and thus may struggle to score responses that differ from those seen in training. It is our experience that deep learning approaches perform well, and we offer two illustrative examples.

Although based on small samples ($n \approx 75$ per item) not seen by either engine, we found a classical engine agreed at lower rates with teachers across 11 essay items and dimensions compared to a hybrid (classical + BERT) engine (Table 2.5). While the use of the hybrid model makes the contribution of BERT somewhat opaque, the improvement suggests that including BERT did not harm, and presumably helped, performance.

We see similar results from recent test administrations in one western state (Table 2.6). Agreement with trained human raters was examined for 12 items across two different administrations. In Administration 1, a classical engine was used, and in Administration 2, a hybrid engine was used. While human–human agreement results were not available on these same responses because no second human ratings are used in the live scoring, the human agreement metrics on the responses used to validate the engine are presented for reference (H1H2).

Table 2.5 Differences in Exact Agreement and QWK between Classical and Hybrid (Classical + BERT) Engines on Teacher-Scored Responses (Grades 3, 5, 7, and 9, $n \approx 75$ per Item, 11 Items)

		Exact Agreement			QWK		
Traits	Possible Scores	Teacher	Classical	Classical + BERT	Teacher	Classical	Classical + BERT
Conventions	0,1,2	60%	56%	57%	0.65	0.53	0.59
Evidence and Elaboration	1,2,3,4	63%	55%	59%	0.67	0.49	0.58
Purpose, Focus, and Organization	1,2,3,4	69%	67%	70%	0.58	0.55	0.61

Table 2.6 Differences in Exact Agreement and QWK between Classical and Hybrid (Classical + BERT) Engines on Operational Responses (Grades 3–8, $n \approx 500$ per Item, 12 Items)

		Exact Agreement			QWK		
Traits	Possible Scores	H1H2	Classical (Admin 1)	Classical + BERT (Admin 2)	H1H2	Classical (Admin 1)	Classical + BERT (Admin 2)
Conventions	0,1,2	71%	66%	68%	0.60	0.57	0.64
Evidence and Elaboration	1,2,3,4	69%	61%	75%	0.64	0.54	0.60
Purpose, Focus, and Organization	1,2,3,4	68%	65%	71%	0.65	0.59	0.63

Finally, it remains to be seen how deep learning systems perform on adversarial writing or on unusual writing. Deep learning engines, because they consider word order as part of their prediction, may be more robust to at least some types of gaming behavior. However, research suggests that they are susceptible to gaming behaviors. Filighera et al. (2020) found that the inclusion of certain unrelated words at the start of a response can artificially inflate scores. Lottridge, Godek, et al. (2020) found that BERT engines are more susceptible than a classical engine to word shuffling, on-topic, non-sense essays using complex language, and off-topic essays, giving higher scores than warranted. BERT, however, outperformed the classical engine when essays were duplicated. These results suggest that filters, like those used in classical engines, are still required for deep learning engines.

7. Conclusion

Automated scoring is a well-established area of machine learning that enjoys widespread use in many large-scale assessment programs. Most automated scoring engines use a classical approach whereby features are expertly crafted and statistical models are used to predict scores. Deep learning systems offer the possibility of producing end-to-end scoring systems without the need for explicitly designed features, and they enable the use of pretrained models built upon large corpora to better represent language.

As was raised throughout this chapter, deep learning automated scoring engines have challenges that would greatly benefit from further study by psychometricians, data scientists, computer scientists, and computational linguists. We expect that, as these models become more popular, core issues of these challenges will be thoroughly investigated and addressed. In fact, one significant benefit of deep learning engines based upon publicly available models is that they can be researched by a broad audience, unlike the black box approaches that use tightly controlled features. We hope that the automated scoring field endeavors to consider deep learning approaches as worthy of investigation and use, despite their complexity and relative immaturity compared to classical approaches. Finally, we hope that the field continues to leverage and benefit from the latest in deep learning approaches. As these approaches are refined and improved, we believe that they will support more reliable and accurate scoring of examinee responses.

Note

1 https://github.com/edx/ease

References

Akiba, T., Sano, S., Yanase, T., & Ohta, T. (2019). *Optuna: A next-generation hyperparameter optimization framework* (pp. 2623–2631). Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

American Educational Research Association, American Psychological Association, & National Council on Measurement in Education (AERA/APA/NCME) (2014). *Standards for educational and psychological testing*. AERA/APA/NCME.

Basta, C., Costa-jussà, M., & Casas, N. (2019). Evaluating the underlying gender bias in contextualized word embeddings. *arXiv*. https://arxiv.org/pdf/1904.08783.pdf

Bejar, I. I., Flor, M., Futagi, Y., & Ramineni, C. (2014). On the vulnerability of automated scoring to construct-irrelevant response strategies (CIRS): An illustration. *Assessing Writing*, *22*, 48–59.

Beltagy, I., Peters, M., & Cohan, A. (2020). Longformer: The long-document transformer. *arXiv*. https://doi.org/10.48550/arXiv.2004.05150

Bernstein, J., De Jong, J., Pisoni, D., & Townshend, B. (2000). *Two experiments on automatic scoring of spoken language proficiency* (pp. 57–61). Proceedings of InSTIL2000 (In8tegrating Speech Technology in Learning), University of Abertay Dundee.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *The Journal of Machine Learning Research*, *3*, 993–1022.

Bolukbasi, T., Chang, K.-W., Zou, J., Saligrama, V., & Kalai, A. (2016a). *Man is to computer programmer as woman is to homemaker? Debiasing word embeddings*. Presented at the 30th Conference on Neural Information Processing System (NIPS).

Bolukbasi, T., Chang, K.-W., Zou, J., Saligrama, V., & Kalai, A. (2016b). *Quantifying and reducing stereotypes in word embeddings*. ICML Workshop on #Data4Good: Machine Learning in Social Good Applications.

Bridgeman, B., Trapani, C., & Attali, Y. (2009). *Considering fairness and validity in evaluating automated scoring*. Paper presented at the annual meeting of the National Council on Measurement in Education.

Bridgeman, B., Trapani, C., & Attali, Y. (2012). Comparison of human and machine scoring of essays: Differences by gender, ethnicity, and country. *Applied Measurement in Education*, *25*(1), 27–40.

Brunet, M.-E., Alkalay-Houlihan, C., Anderson, A., & Zemel, R. (2019). *Understanding the origins of bias in word embeddings*. Proceedings of the 36th International Conference on Machine Learning.

Burkhardt, A., & Lottridge, S. (2013). *Examining the impact of training samples on identifying off-topic responses in automated essay scoring*. Paper presented at the annual meeting of Northern Rocky Mountain Educational Research Association.

Burrows, S., Gurevych, I., & Stein, B. (2015). The eras and trends of automatic short answer grading. *International Journal of Artificial Intelligence in Education*, *25*, 60–117.

Burstein, J., & Chodorow, M. (1999). *Automated essay scoring for nonnative English speakers*. Proceedings of Computer-Mediated Language Assessment and Evaluation of Natural Language Processing.

Cahill, A., & Evanini, K. (2020). Natural language processing for writing and speaking. In D. Yan, A. Rupp, & P. W. Foltz (Eds.), *Handbook of automated scoring: Theory into practice* (pp. 69–92). CRC Press.

Cahill, A., Fife, J., Riordan, B., Vajpayee, A., & Galochkin, D. (2020). Context-*based automated scoring of complex mathematics responses*. Proceedings of the 15th Workshop on Innovative Use of NLP for Building Educational Applications.

Chelba, C., Mikolov, T., Schuster, M., Ge, Q., Brants, T., Koehn, P., & Robinson, T. (2013). One billion word benchmark for measuring progress in statistical language modeling. *arXiv*. https://doi.org/10.48550/arXiv.1312.3005

Chevalier, S. (2007). *Speech interaction with Saybot player, a CALL software to help Chinese learners of English*. Proceedings of the International Speech Communication Association Special Interest Group on Speech and Language Technology in Education (SLaTE).

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv*. https://doi.org/10.48550/arXiv.1412.3555

Church, K., & Liberman, M. (2021). The future of computational linguistics: On beyond alchemy. *Frontiers in Artificial Intelligence*, *4*.

Clark, K., Luong, M.-T., Le, Q., & Manning, C. (2020). ELECTRA: Pre-training text encoders as discriminators rather than generators. *arXiv*. https://doi.org/10.48550/arXiv.2003.10555

Corbett-Davies, S., & Goel, S. (2018). The measure and mismeasure of fairness: A critical review of fair machine learning. *arXiv*. https://arxiv.org/abs/1808.00023

Deerwester, S., Dumais, S. T., & Landauer, T. K. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, *41*(6), 391–407.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv*. https://doi.org/10.48550/arXiv.1810.04805

Dong, F., Zhang, Y., & Yang, J. (2017). *Attention-based recurrent convolutional neural network for automatic essay scoring* (pp. 153–162). Proceedings of the 21st Conference on Computational Natural Language Learning.

Fife, J. H. (2013). *Automated scoring of mathematics tasks in the common core era: Enhancements to M-rater in support of CBAL mathematics and the common core assessments (RR-13-26)*. ETS Research Report Series.

Filighera, A., Steuer, T., & Rensing, C. (2020). Fooling automatic short answer grading systems. In I. I. Bittencourt, M. Cukurova, K. Muldner, R. Luckin, & E. Millán (Eds.), *Artificial intelligence in education* (pp. 177–190). Springer.

Foltz, P. W., Yan, D., & Rupp, A. A. (2020). The past, present, and future of automated scoring. In D. Yan, A. Rupp, & P. W. Foltz (Eds.), *Handbook of automated scoring: Theory into practice* (pp. 1–9). CRC Press.

Gage, P. (1994). A new algorithm for data compression. *The C Users Journal Archive*, *12*, 23–38.

Ghosh, D., Klebanov, B., & Song, Y. (2020). *An exploratory study of argumentative writing by young students: A transformer-based approach*. Proceedings of the 15th Workshop on Innovative Use of NLP for Building Educational Applications.

Graves, A., Wayne, G., & Danihelka, I. (2014). Neural turing machines. *arXiv*. https://doi.org/10.48550/arXiv.1410.5401

Gregg, N., Young, M., & Lottridge, S. (2021). *Examining fairness in automated scoring*. Paper presented at the annual conference of the National Council on Measurement in Education, Online.

Guthrie, D., Allison, B, Liu, W., Guthrie, L, & Wilks, Y. (2006). *A closer look at skip-gram modeling*. Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06).

Higgins, D., Burstein, J., & Attali, Y. (2005). Identifying off-topic student essays without topic-specific training data. *Natural Language Engineering*, *1*(1), 1–15.

Higgins, D., & Heilman, M. (2014). Managing what we can measure: Quantifying the susceptibility of automated scoring systems to gaming behavior. *Educational Measurement: Issues and Practice*, *33*(3), 36–46.

Hochreiter, S. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Hutchinson, B., & Mitchell, M. (2019). *50 years of test (un)fairness: Lessons for machine learning*. FAT* 19: Conference on Fairness, Accountability and Transparency.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). *An introduction to statistical learning, with applications in R*. Springer Science + Business Media.

Jawahar, G., Sagot, B., & Sedda, D. (2019). *What does BERT learn about the structure of language?* Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL).

Kitaev, N., Kaiser, L., & Levskaya, A. (2020). Reformer: The efficient transformer. *arXiv*. https://doi.org/10.48550/arXiv.2001.04451

Kolowich, S. (2014, April 28). Writing instructor, skeptical of automated grading, pits machine vs. machine. *The Chronicle of Higher Education*.

Kovaleva, O., Romanov, A., Rogers, A., & Rumshisky, A. (2019). *Revealing the dark secrets of BERT* (pp. 4365–4374). Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing.

Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, *22*(1), 79–86.

Liaw, R., Liang, E., Nishihara, R., Moritz, P., Gonzalez, J., & Stoica, I. (2018). Tune: A research platform for distributed model selection and training. *arXiv*. https://doi.org/10.48550/arXiv.1807.05118

Liu, O. L., Brew, C., Blackmore, J., Gerard, L., Madhok, J., & Linn, M. (2014). Automated scoring of constructed-response science items: Prospects and obstacles. *Educational Measurement: Issues and Practice*, *33*(2), 19–28.

Lochbaum, K. E., Rosenstein, M., Foltz, P., & Derr, M. A. (2013). *Detection of gaming in automated scoring of essays with IEA*. Paper presented at the annual meeting of the National Council on Measurement in Education.

Lottridge, S., Burkhardt, A., & Boyer, M. (2020). Automated scoring [Digital ITEMS module 18]. *Educational Measurement: Issues and Practice*, *39*(3).

Lottridge, S., Godek, B., Jafari, A., & Patel, M. (2020). *Comparing the robustness of deep learning and classical automated scoring approaches to gaming strategies*. Paper presented at the annual meeting of the National Council on Measurement in Education, Online.

Lottridge, S., Woolf, S., Young, M., Jafari, A., & Ormerod, C. (2021). *Explaining crisis alerts from humans and automated scoring engines using annotations*. Paper presented at the annual meeting of the National Council on Measurement in Education, Online.

Lottridge, S., & Young, M. (2022). *Examining bias in automated scoring of reading comprehension items*. Paper presented at the annual meeting of the National Council on Measurement in Education.

Lundberg, S., & Lee, S.-I. (2017). *A unified approach to interpreting model predictions*. Paper presented at the 31st Conference on Neural Information Processing Systems (NIPS).

Madnani, N., Loukina, A., von Davier, A., Burstein, J., & Cahill, A. (2017). *Building better open-source tools to support fairness in automated scoring* (pp. 41–52). Proceedings of the First ACL Workshop on Ethics in Natural Language Processing.

Marcus, M., Santorini, B., & Marcinkiewics, M.-A. (1993). Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*, *19*, 313–330.

Mayfield, E., & Black, A. (2020). *Should you fine-tune BERT for automated essay scoring?* (pp. 151–162). Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications.

Matthias, S., & Bhattacharyya, P. (2020). *Can neural networks automatically score essay traits?* (pp. 85–91). Proceedings of the 15th Workshop on Innovative Use of NLP for Building Educational Applications.

McCourt, P. H. (2019). *SigOpt documentation*. SigOpt, Inc. https://docs.sigopt.com/

Merity, S., Xiong, C., Bradbury, J., & Socher, R. (2016). Pointer sentinel mixture models. *arXiv*. https://doi.org/10.48550/arXiv.1609.07843

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv*. https://doi.org/10.48550/arXiv.1301.3781

Mikolov, T., Karafiát, M., Burget, L., Černocký, J., & Khudanpur, S. (2010). *Recurrent neural network based language model*. Paper presented at the Eleventh Annual Conference of the International Speech Communication Association.

National Center for Education Statistics (NCES). (2022). *Results from the NAEP automated scoring challenge*. www.info/results.md at main NAEP-AS-Challenge/infoGitHub

Nicewander, A., Sukin, S., & Lottridge, S. (2015). *IRT-based reliability estimates for human and machine scored essays*. Paper presented at the annual meeting of the National Council on Measurement in Education.

Ormerod, C., Lottridge, S., Harris, A., Patel, M., van Wamelen, P., Kodeswaran, B., Woolf, S., & Young, M. (2022). Automated short answer scoring using an ensemble of neural networks and latent semantic analysis classifiers. *International Journal of Artificial Intelligence in Education*, 1–30.

Ormerod, C., Malhotra, A., & Jafari, A. (2021). Automated essay scoring using efficient transformer-based language models. *arXiv*. https://doi.org/10.48550/arXiv.2102.13136

Page, E. (2003). Project essay grade: PEG. In M. Shermis & J. Burstein (Eds.), *Automated essay scoring: A cross-disciplinary perspective* (pp. 43–54). Routledge.

Papakyriakopoulos, O., Hegelich, S., Serrano, J., & Marco, F. (2020). *Bias in word embeddings*. Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT*).

Ramineni, C., & Williamson, D. (2018). *Understanding mean score differences between the e-rater® automated scoring engine and humans for demographically based groups in the GRE® general test (RR-18-12)*. GRE Board Research Report Series and ETS Research Report Series.

Ribeiro, M. T., Sigh, S., & Guestrin, C. (2016). 'Why should I trust you?' Explaining the predictions of any classifier. *arXiv*. https://arxiv.org/pdf/1602.04938v1.pdf

Riordan, B., Bichler, S., Bradford, A., Chen, J., Wiley, K., Gerard, L, & Linn, M. (2020). *An empirical investigation of neural methods for content scoring of science explanations*. Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications.

Rodriguez, P., Jafari, A., & Ormerod, C. (2019). Language models and automated essay scoring. *arXiv*. https://doi.org/10.48550/arXiv.1909.09482

Rupp, A. A. (2018). Designing, evaluating, and deploying automated scoring systems with validity in mind: Methodological design decisions. *Applied Measurement in Education*, *31*(3), 191–214.

Sakaguchi, K., Heilman, M., & Madnani, N. (2015). *Effective feature integration for automated short answer scoring* (pp. 1049–1054). Proceedings of the 2105 Conference of the North American Chapter of the Association of Computational Linguistics: Human Language Technologies.

Sarker, A., Klein, A. Z., Mee, J., Harik, P., & Gonzalez-Hernandez, G. (2019). An interpretable natural language processing system for written medical examination assessment. *Journal of Biomedical Informatics*, *98*, 103268.

Sennrich, R., Haddow, B., & Birch, A. (2015). Neural machine translation of rare words with subword units. *arXiv*. https://doi.org/10.48550/arXiv.1508.07909

Shermis, M. (2014). State-of-the-art automated essay scoring: Competition, results, and future directions from a United States demonstration. *Assessing Writing*, *20*, 53–76.

Shermis, M., & Burstein, J. (Eds.). (2006). *Automated essay scoring: A cross-disciplinary perspective*. Lawrence Erlbaum Associates.

Shermis, M., & Lottridge, S. (2019). *Communicating to the public about machine scoring: What works, what doesn't*. Paper presented at the annual meeting of the National Council on Measurement in Education.

Shermis, M., Mao, L., Mulholland, M., & Kieftenbeld, V. (2017). Use of automated scoring features to generate hypotheses regarding language-based DIF. *International Journal of Testing*, *17*(4), 351–371.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(56), 1929–1958.

Sun, Z., Yu, H., Song, X., Liu, R., Yang, Y., & Zhou, D. (2020). MobileBERT: A compact task-agnostic BERT for resource-limited devices. *arXiv*. https://doi.org/10.48550/arXiv.2004.02984

Sundararajan, M., Taly, A., & Yan, Q. (2017). *Axiomatic attribution for deep networks*. Proceedings of the 34th International Conference on Machine Learning.

Taghipour, K., & Ng, H.-T. (2016). *A neural approach to automated essay scoring* (pp. 1882–1891). Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing.

Uto, M., Xie, Y., & Ueno, M. (2020). *Neural automated essay scoring incorporating handcrafted features* (pp. 6077–6088). Proceedings of the 28th International Conference on Computational Linguistics.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *arXiv*. https://doi.org/10.48550/arXiv.1706.03762

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv*. https://doi.org/10.48550/arXiv.1804.07461

Wang, S., Li, B., Khabsa, M., Fang, H., & Ma, H. (2020). Linformer: Self-attention with linear complexity. *arXiv*. https://doi.org/10.48550/arXiv.2006.04768

Wang, Z. (2021). *Comparison of human rater and automated scoring of test takers' speaking ability and classification using item response theory*. Paper presented at the annual meeting of the National Council on Measurement in Education, Online.

Wang, Z., & von Davier, A. (2014). *Monitoring of scoring using the e-rater automated scoring system and human raters on a writing test (RR-14-04)*. ETS Research Report Series.

Williamson, D., Xi, X., & Breyer, F. J. (2012). A framework for the evaluation and use of automated scoring. *Educational Measurement: Issues and Practice*, *31*(1), 2–13.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., & Rush, A. (2019). HuggingFace's transformers: State-of-the-art natural language processing. *arXiv*. https://doi.org/10.48550/arXiv.1910.03771

Wolfe, E. (2020). Human scoring with automated scoring in mind. In D. Yan, A. Rupp, & P. W. Foltz (Eds.), *Handbook of automated scoring: Theory into practice* (pp. 49–68). CRC Press.

Wood, S. (2020). Public perception and communication around automated essay scoring. In D. Yan, A. Rupp, & P. W. Foltz (Eds.), *Handbook of automated scoring: Theory into practice* (pp. 133–150). CRC Press.

Woods, B., Adamson, D., Miel, S., & Mayfield, E. (2017). *Formative essay feedback using predictive scoring models*. Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Xi, X., Higgins, D., Zechner, K., & Williamson, D. (2008). *Automated scoring of spontaneous speech using SpeechRater ® v1.0 (RR-08-62)*. ETS Research Report Series.

Yan, D., & Bridgeman, B. (2020). Validation of automated scoring systems. In D. Yan, A. Rupp, & P. W. Foltz (Eds.), *Handbook of automated scoring: Theory into practice* (pp. 297–318). CRC Press.

Yao, L., Haberman, S., & Zhang, M. (2019). Penalized best linear prediction of true test scores. *Psychometrika*, *84*, 186–211.

Zhang, M., Chen, J., & Ruan, C. (2016). *Evaluating the advisory flags and machine scoring difficulty in the e-rater automated scoring engine (ETS RR-16-30)*. ETS Research Report Series.

Zhou, Z., Wu, J., & Tang, W. (2002). Ensembling neural networks: Many could be better than all. *Artificial Intelligence*, *137*(1–2), 239–263.

3

Speech Analysis in Assessment

Jared C. Bernstein and Jian Cheng

1. Introduction

Assessments that elicit and evaluate spoken responses have been used for many centuries – in job interviews, medical diagnosis, and in formal schooling. We present methods that enable automatic scoring of spoken responses in testing and instruction, along with notes on the current limits on their accuracy and application. Much of the content of a spoken response (word sequences, propositions, some hedges, and some clues to sentiment) would be found in a text transcription of the response, but many speech-borne cues have no conventional orthographic representation. This chapter describes the methods currently available for extracting the lexical content and other information from spoken responses in the context of assessments that are used in selection and qualification, as well as in controlling adaptive instruction and in providing formative guidance to learners.

First, we present an overview of the information carried by a spoken response to a test item or an interview question. This includes both the information that would be found in an orthographic transcript of the response and the extra-linguistic information that is conveyed in the vocal performance of the bare text that would be found in a transcript.

Next, we describe the technologies used to extract the text transcript in the response. We intend to explain them as clearly as we can so that a reader can form an intuition about their operation and performance, giving citations to accessible primary sources.

Then, we present two applications of the technologies in education: the design, development, and evaluation of a fully automated assessment of second language listening and speaking, and one test of basic reading.

Finally, we summarize the state of the art in some domains where spoken language processing (SLP) has been successfully applied and review the technical and social circumstances that have slowed adoption in assessment, including the need for transparency in assessment scoring.

DOI: 10.4324/9781003278658-4
This chapter has been made available under a CC-BY-NC-ND 4.0 license.

2. Information in Speech

Usually, the predominant and most important information carried by a speech signal is exactly the sequence of words in the signal. These words, if they conform reasonably well to common patterns of a known language and if they are accurately transcribed, yield a text that a reader or a natural language processing (NLP) system can analyze to derive propositional content, sentiment, and contextual meanings. Apart from this text-borne content, consider what you might infer from a recording of a person speaking a language that you don't know at all. You might infer (correctly or not) certain traits such as the person's size, sex, and age. Additionally, you might infer aspects of the speaker's state at the time of speaking, such as the speaker's level of arousal or psychomotor coordination. If you know the language spoken, you might infer much more about the speaker's traits and states. For example, you might hear evidence that the speaker was from a particular location, had a certain level of vocabulary or education, had fluent facility with the topic at hand, or had struggled to compose utterances on this topic at this time. You might also hear other qualities like amazement or loathing that could be inferred from a combination of lexical valence and tone of voice.

For the purposes of this chapter, we divide the information in speech into three categories:

1. Linguistic – sequences of lexical and other units, such as words, phrases, clauses, utterances.
2. Paralinguistic – aspects of spoken performance that reflect speaker states.
3. Indexical – aspects of spoken performance that reflect speaker identity or other stable speaker characteristics.

This chapter will focus on the linguistic content of spoken responses, as the paralinguistic and indexical aspects of spoken test responses are not yet widely used in assessment or instruction.

2.1 Linguistic Information

Starting with content, we can examine the speech recognition output for a nonnative speaker who is describing a silent video.

Some information is evident immediately – even from the waveform shown in Figure 3.1. The test-taker starts to talk soon after the video starts, and he pauses several times for 1 to 2 seconds while speaking, but he keeps speaking almost until the end of the video. We might make a tentative inference that the test-taker does not seem reticent, and he probably understood that he was expected to speak.

If we consider the words that he says, we can easily calculate his *speaking rate* (in words per minute) and his *articulation rate* (in words per second of speech time). The target construct for this 'describe the silent video' item was *proficiency in spoken English*, and the timing and the appropriateness of the linguistic content produced in a response to this item provides good information about a test-taker's language-related skills in the spontaneous word-retrieval and English sentence formation. Note, however, that the *rate of speech* will also be affected by the

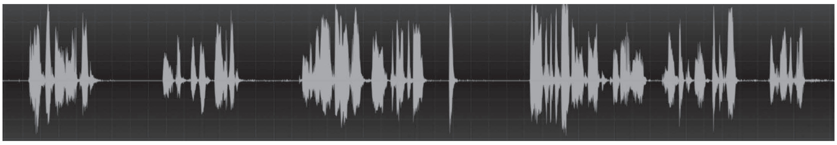


Figure 3.1 The waveform of a nonnative speaker's 22-second spoken response when asked to describe what he sees depicted in a silent 25-second video clip.

pace of action in the video, by the person's quickness in understanding the setting, the characters, and the activities in the silent video, as well as by the test-taker's enduring traits such as timidity or bravery, as modulated by his current level of psychomotor activation. These personal and time-variable characteristics of the test-taker are conventionally considered construct-irrelevant with respect to proficiency in spoken English. However, articulation rate is less affected by the construct-irrelevant influences on spoken performance in responding to this real-time description task.

A machine transcription of this response is:

<SIL 1.0s> a man's walking in a park
<SIL 1.7s> in a uh he's wearing a hat
<SIL 1.6s> and another lady was <@ 0.5s> picking up a ball
<SIL 0.6s> oh
<SIL 2.0s> the lady pick up the ball and return that to the man
<SIL 0.4s> man thanked him uh thanked her
<SIL 0.8s> and uh walked away <SIL 2.0s>

In this transcription, a 1-second silence is shown as <SIL 1.0s>, and one-half second of unintelligible speech is represented by <@ 0.5s>. The utterance, from the start of the first word to the end of the last word, lasts 22 seconds, during which time 42 words were spoken. Thus, the test-taker's *speech rate* in this response is about 112 words per minute, which is well within the range of rates found for native speakers of English doing this task. If we subtract the durations of the utterance-internal pauses, then we can calculate this test-taker's articulation rate to be about 164 words per second of speech time, which again is within the range of articulation rates found for fluent native speakers.

Examination of the linguistic content of the machine transcription will yield more evidence that can be used to extract a scale value of the test-taker's speaking proficiency and/or a diagnostic profile of the test-taker's strengths and deficits in spoken English. Most of the recognized spoken material in this 22-second response is quite good. The test-taker uses words and constructions that are found in native and in high-proficiency nonnative responses to this video clip. There are also indications that the test-taker, though fluent, uses some forms that are outside the range of common native English speech – consider the use in this context of 'another', 'pick', and 'man thanked him'.

2.2 Paralinguistic Information

Knowing that a test candidate sounds anxious or relaxed can be used to personalize the presentation of item material and even the selection of item material during automated interactive assessment. Paralinguistic aspects of the speech signal have not yet often been used in assessment or instruction, although one can easily imagine their use in adapting the substance or manner of presentation coming from the machine's side to suit a test-taker's emotional state or level of psychomotor activation. Automated speech emotion recognition has been an active field since the 1990s, with systems using combinations of linguistic content, paralinguistic features, and purely acoustic features to estimate the emotional state of a speaker.

Note that well-documented methods for detecting sentiment (emotion and attitude toward topic) from the words and statements in written material can be applied directly to the word sequences that comprise a spoken response and will account for a large portion of the variance found in human judgments of speaker emotion. Sentiment analysis is reviewed in Jurafsky and Martin (2021, chapters 4, 20). Early machine learning approaches to speech emotion recognition (e.g., Rosenfeld et al., 2003) often relied on face-valid features of the speech signal, such as response latency, speech rate, average pitch and pitch variability, or pause duration. More recently, most emotion recognition systems operate on acoustic features that have little obvious relation to traditional psychological concepts. For reviews of recent methods, see B. Schuller (2018) or Schuller and Schuller (2020), or Swain et al. (2018).

2.3 Indexical Information

Machine leaning methods may yield indexical information that is *demographic* – for example, that the current student or test-taker is male or female, very old or very young, or from a particular linguistic or geographic background. This chapter will not address demographic identification by voice. Here we limit ourselves to individual identity – a particular person did or did not produce this speech signal. Note, however, that identification of some demographic categories can be achieved by direct adaptations or extensions of the techniques described next for identification of individuals.

In assessments, the indexical traits of a spoken response can be important for security –for example, to monitor if there is more than one speaker in a recorded response, or to verify that a particular person was the respondent in a previously unsecured test performance. Automatic speaker verification (ASV), or speaker recognition, is a subfield of spoken language processing that has available benchmark datasets and well-established machine learning methods that yield accuracies sufficient for many uses in assessment. Human listeners often recognize familiar persons by prosodic and lexical characteristics of the person's style. For example, a person speaks quickly, with a flat affect, and often uses certain uncommon words or idiosyncratic pause fillers. These are often the characteristics that mimics or impersonators use to great effect for entertainment. However, human listeners are also very sensitive to acoustic-phonetic aspects of speech that signal a speaker's identity, and that may be evident in a laugh or in another nonverbal vocal production, as well as in normal speech. ASV systems focus on these acoustic features that depend primarily on anatomical properties of a person's vocal tract (mouth, pharynx, larynx, lungs). These acoustic features of speech are well represented in the 40 mel-frequency cepstral coefficients that represent acoustic spectra as used in automatic speech recognition (see the description of Figure 3.3 in the 'Operation and Development' section).

A general use case for automatic speaker verification has a set of known talkers and a speech signal that might have come from one of the known talkers or from anyone. ASV operates using an acoustic model of the speech of each of the known talkers and a background acoustic model of all possible talkers, called a universal background model (UBM). These acoustic models may both be simply Gaussian mixture models, with each known-talker model having been built by modifying the all-talker model with samples of that known-talker's speech. The question at hand is whether or not a new incoming signal has come from one of the known talkers, and that question can be answered with high accuracy by calculating which model yields a higher probability to have produced this new signal. Reynolds et al. (2000) introduced this UBM method, and it remains a benchmark standard that newer methods are often evaluated against. Hansen and Hasan (2015) give an excellent and accessible review of ASV technology in several contexts and compare its operation and performance to human skill. More recent deep learning approaches to ASV are reviewed in Irum and Salman (2019) and in Bai and Zhang (2021).

Finally, note that there is an evolving tension, or arms race, between ASV technology and fake-a-talker or voice-spoofing technologies, which work against each other like feuding siblings. More accurate fake detection leads to more effective fakes, which in turn incorporate each new fake detection technique in training. Das et al. (2020) cover some of these recent technologies and countermeasures.

The next two sections introduce automatic speech recognition (ASR) technology and then describe some 'current' (as of 2022) applications of ASR in assessment. For each topic, we cover basic operation, development, and evaluation. Then we provide a review of recent progress and an annotated list of tools and resources.

3. Automatic Speech Recognition in SLP

Automatic speech recognition (ASR) is the conversion of a speech signal into a text file. An ASR system takes in speech signals and produces text. The output text file may be orthographic, producing a best estimate of what the ASR system infers the speaker would want as a printable representation of the spoken material. Orthographic ASR output would include capitalization and punctuation, and it typically suppresses false starts and repeated words. That is, some commercial ASR systems are designed to produce an acceptable orthographic output, which is closer to an expected text representation of the speaker's inferred intention. Such an orthographic ASR system might convert a spoken snippet like (a) into a text output like (b).

(a) '. . . home they wen(t) – <SIL 0.4s> they drove to the . . .'
(b) '. . . home. They drove to the . . .'

In this chapter, we take ASR to mean a system that produces an augmented text file, more like (a) in the example just mentioned. Beyond silences and false starts, augmented text output can include an aligned fundamental frequency (pitch) track and mark unintelligible sections of the signal. The augmentation may include start and end times for each phoneme and for each word, along with confidence or likelihood scores for each word and phoneme. The text may also be augmented with a constituent or dependency parse that indicates the likely groupings of adjacent words into longer units.

Spoken language processing (SLP) is the integration of ASR output with natural language processing (NLP) to extract meaning from speech signals. Meaning is taken here to be information from the speech that is useful in a particular context. For us, this context is assessment. In many applications, including assessment, the meaning a machine takes from the text found in a speech signal is no more than would be taken from a user's selection from among available presented choices or from a user's typed response within a text interaction. Our goal in this chapter is to explain ASR (speech recognition), rather than the larger SLP technology, as the application of NLP in assessment contexts is covered in other chapters. The following section describes what an ASR system does and how it operates.

3.1 Overview

A speech recognizer, or ASR system, accepts a digitized speech signal and produces a word sequence that is most likely to have produced that speech signal. This section describes a feasible speech recognizer that uses a traditional set of parameters and values that stand in for a wide range of actual implementations. At a high level of description, an ASR system first digitizes and analyzes an incoming acoustic signal to produce a sequence of spectra (usually 100 per second of signal), which is fed to a search process that finds the single path in a compiled language model (a directed graph with word nodes) that has the greatest likelihood to have produced that sequence of spectra.

Many currently available systems operate roughly in this manner, but with implementations of the processes that are not neatly separated, as in Figure 3.2. This section will describe features of a typical, traditional (1990–2015) ASR system, to provide some intuition into the

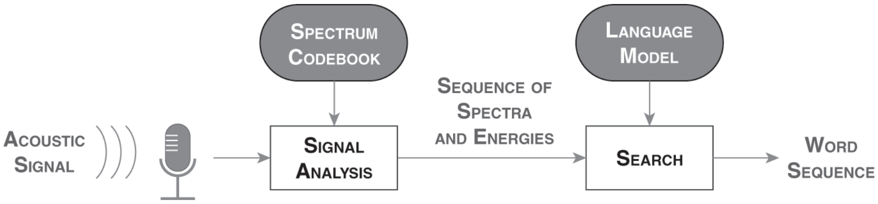


Figure 3.2 Schematic view of the operation of an ASR system, which accepts an acoustic speech signal captured by a microphone and returns an (augmented) sequence of words.

logic and limits of ASR systems generally. The most recent research systems as of this writing (in 2022) attempt fuller, end-to-end implementations that should further improve accuracy, but which have not yet settled into a new standard design suitable for description. The reader can refer to Chapter 9 of the second edition of Jurafsky and Martin's *Speech and Language Processing* book (Jurafsky & Martin, 2009) for a much fuller description of the kind of ASR system merely outlined here. The newer, third edition describes some more recent approaches, but as of March 2023, it was only available in draft form at Jurafsky's Stanford website (https://web.stanford.edu/~jurafsky/slp3). The fundamentals of the ASR methods described here are covered in Jelinek (1998).

3.2 Operation and Development

The acoustic speech signal is sampled 16,000 times a second to yield 16,000 sixteen-bit values per second. This sampling resolution allows the calculation of the acoustic spectrum of the speech signal in the range from 0 to 8,000 Hz. A high-fidelity speech signal that is band-limited to this narrower frequency range is sufficient for excellent word recognition by a human listener. Figure 3.3 shows a 1.5-second signal waveform sampled at 16,000 samples per second. Just below it and aligned with it is the corresponding 8 kHz spectrogram for that digital signal file. The file contains the isolated utterance 'two pills', with about 100 msec. of silence before and after the speech. Time is represented from left to right. The white ticks in the middle of Figure 3.3 occur every 10 milliseconds, so there are 100 ticks per second of signal. The short inter-tick intervals of signal are called *frames*, and the ASR system calculates about 40 mel-frequency cepstral coefficients that represent the spectrum and amplitude of the signal in each frame, typically using a 20-millisecond time-window centered at that frame position.

A speech recognition system typically operates by finding the highest likelihood path for an observed sequence of spectra to traverse a network of possible word sequences. For our illustrative purpose, we can posit a number of words and nonlexical acoustic events like silences and mouth noises and nonlexical pause fillers (e.g. 'uh', 'mm', etc.). This exposition will just focus on the words for simplicity.

First, there is a lexicon, which lists the words and other nonlexical events that the system knows. This lexicon may have hundreds of thousands of word-form entries. Each word-form has an orthographic form associated with one or more phonemic forms, constructed from a phonemic alphabet (for English) with 35–45 different phonemes. The phonemic forms are typically a sequence of phoneme labels and may include syllabic or stress markings – all much like a conventional dictionary entry without grammatical information and without a definition, just the pronunciation field. For example, an entry might be simply *TWO => /T U/ or PILLS* => /P IH L Z/. The CMU Pronouncing Dictionary (CMU, 2022) is a well-established, open-source lexicon with over 130,000 words developed at Carnegie Mellon University and

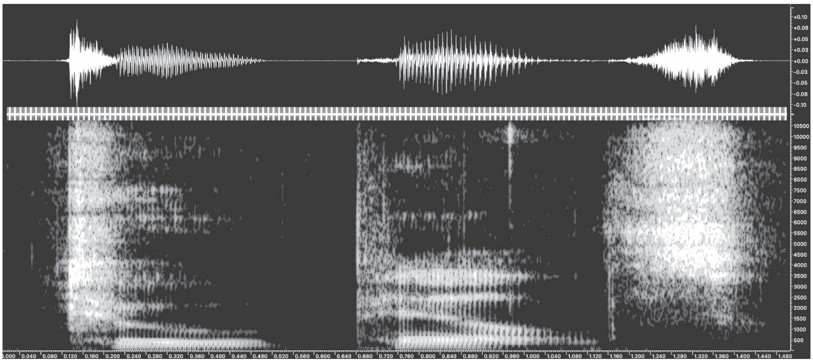


Figure 3.3 The waveform and spectrogram of ‘two pills’, spoken slowly in isolation.

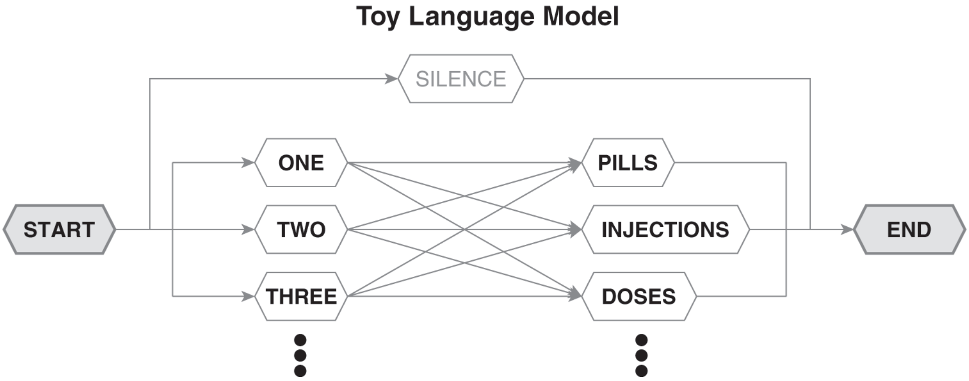


Figure 3.4 Toy language model that might be used to answer a test question about medications.

Note that even at this grossly simple level, the graph is missing the singular forms ‘pill’, ‘injection’, and ‘dose’, and thus cannot match ‘one pill’ or ‘one dose’.

used in many research and commercial ASR systems. It gives common North American pronunciations in ARPABET form.

Second, a Language Model (LM) is developed that arranges the possible words in a graph with word nodes and with probabilities on the arcs between the word nodes. At the top level, this LM graph represents the probability of each word occurring at a given serial position within any sequence of words. In developing an LM, the goal is to construct the lowest entropy (most constrained) LM graph that yields the highest ASR accuracy *for the application at hand*. A very simple LM is presented in Figure 3.4 that might be trained to recognize a test question about a medication dosage. LMs for assessment tasks are discussed later in this chapter.

Next, each word node in the graph is replaced by (unpacked as) a phoneme-level graph with paths that represent the possible pronunciations of the word (often just one) as found in the lexicon, and these are put in the graph as connected sequences of phoneme nodes. Each phoneme node is further unpacked as three *state* nodes – one state for the early, one for the middle, and one for the late portions of the signal that correspond to that phoneme. A very simple example of these embedded graphs is shown in Figure 3.5, with three state nodes inside each of the phoneme nodes, which are inside the word node for PILLS.

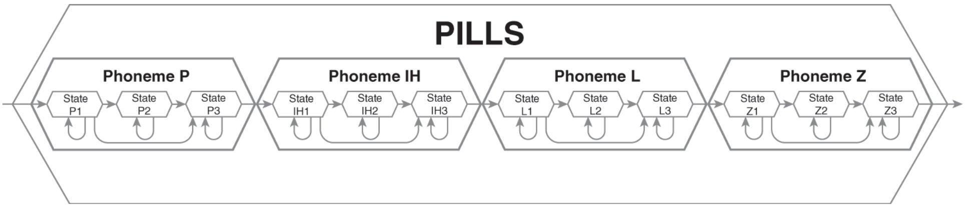


Figure 3.5 Schematic view of the word-node 'pills', unpacked as one pronunciation with four phoneme models, each with three internal states. 'Pills' compiles down into a sequence of 12 different state models: P1, P2, P3, IH1, IH2, IH3, L1, L2, L3, Z1, Z2, Z3.

As the system searches the graph, we can expect that, on average, a phoneme will last for about 10 frames or about 3 or 4 frames per state. The 10 frames per phoneme can be very roughly estimated as:

approx. 600 phonemes/minute <= (approx. 150 words/minute) x (approx. 4 phonemes/word), and
approx. 10 frames/phoneme <= (6,000 frames/minute)/(600 phonemes/minute).

Note, however, that some phonemes can be assigned to just one frame because the state graphs inside a phoneme node often have skip-arcs that may permit a whole phoneme to be aligned with only a single frame. The very clear, slow utterance of 'two pills' shown in Figure 3.3 spans about 1,300 milliseconds of the file and has only six phonemes, which thus average about 22 frames per phoneme, or about 7 frames per state.

As stated earlier, the basic *operation* of an ASR system involves finding the most likely path through the compiled language model where every frame of an utterance is aligned with one state node. In the traditional model here described, the likelihoods are of three types: word-arc, state-arc, and frame-state. Each arc in the LM has an associated likelihood that is estimated from training examples, and each state-to-state arc in the phone models also has a trained likelihood, including the self-loop arcs and the skip-arcs. Each frame has a likelihood that it will occur in each state in the compiled model. The likelihood of a frame, given a state, can be modeled in many ways, but two common methods have been Gaussian mixtures and, more recently, deep neural nets. The process of training an ASR system is mainly focused on estimating the likelihoods for arcs in the LM and the likelihoods of frames given a state within a phoneme model. Figure 3.6 shows general schema for training an ASR system within an assessment and then verifying its performance with reference to a correlation coefficient *r*. For example, a sample of students respond to sets of items presented on a Pilot Test Platform, and their responses are scored by human raters. The item responses are also sent to a Computer Scoring system, and a machine learning process repeatedly updates the Computer Scoring to minimize the difference, 'DIFF', between the set of computer scores and the corresponding set of human reference scores. The machine learning process is then assessed or validated by comparing the final Computer Scoring algorithm to a new set of human scores on a new set of responses from a new student sample. A correlation coefficient, *r*, is often used to measure the correspondence of score sets.

3.3 ASR Evaluation

The standard ASR evaluation metric is Word Error Rate (WER), which is the percentage of recognition word errors in a body of spoken material. WER is calculated by computing the minimum edit distance between the sequence of words spoken and the sequence of words

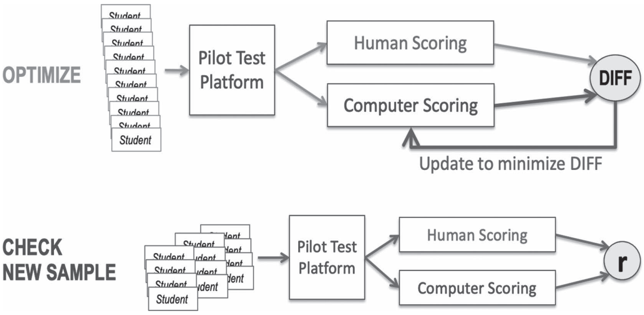


Figure 3.6 Training and evaluating ASR-based scoring in a test.

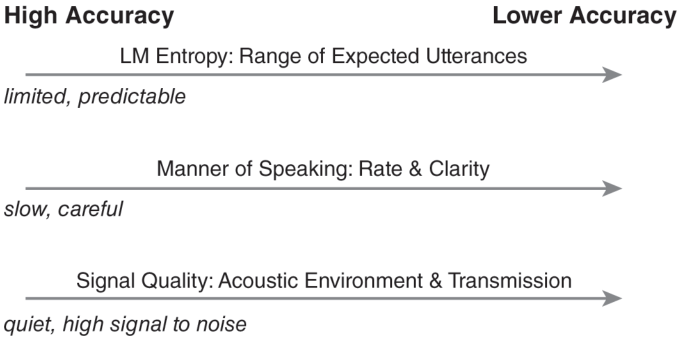

High Accuracy
Lower Accuracy
LM Entropy: Range of Expected Utterances
limited, predictable
Manner of Speaking: Rate & Clarity
slow, careful
Signal Quality: Acoustic Environment & Transmission
quiet, high signal to noise

Figure 3.7 Some factors that affect speech recognition accuracy.

returned by the ASR system. Errors are of three types: insertions, deletions, and substitutions. The WER formula is:

$$\%\text{WER} = 100 \times (\text{insertions} + \text{deletions} + \text{substitutions}) / (\#\text{of words spoken})$$

Current ASR systems vary greatly in WER. Beyond the system's accuracy in isolation, as shown in Figure 3.7, the determinants of WER include the quality of the incoming signal, the manner of speaking, and the uncertainty (entropy) of the LM needed to handle the expected incoming utterances.

3.4 State of the Art

Jurafsky and Martin's online draft (2021, pp. 26–34) reports a range of WER values for several vintage-2020 ASR systems. The best performance is 1.4% WER for audiobook recordings, extending through 11% WER for telephone conversations between family members, and hitting 81% WER for dinner-party conversation recorded from a single microphone at a distance from the talkers. Note that these example WERs are likely for an ASR system with a general (high-entropy) LM that has not been optimized to the topic of the spoken material. Note that a WER of 5% represents, on average, one word error every 20 words, and a WER of 2% is an average of one word error every 50 words.

It is important also to understand which words are most likely to be misrecognized. In general, the missed words are short and often may not have an important impact on the evaluation of an answer to a question within an assessment. A 2017 paper by a group at Microsoft (Xiong et al., 2018) reported ASR performance that reached parity (6% WER) with human transcriptions on the Switchboard Corpus of American English telephone conversations between strangers. Not only did the Microsoft ASR system match professional human transcriptions, but the most frequent kinds of errors committed by human listeners and the Microsoft system were very similar.

Most Common Substitutions:

Human transcripts: *oh/um*, *was/is*, *a/um*, *in/and*, *the/a*, *that/it*
Machine transcripts: *oh/um*, *was/is*, *a/um*, *I/uh in/and*, *the/a*

Most Common Deletions:

Human transcripts: *I*, *and*, *it*, *a*, *that*, *you*, *the*, *to*, *oh*, *yeah*
Machine transcripts: *it*, *I*, *a*, *that*, *you*, *and*, *have*, *oh*, *are*, *is*

These substitutions and deletions will not have a strong impact on the automatic scoring of the correctness of the content of a spoken answer to many kinds of assessment items.

4. Examples of Current Applications

Assessment often involves measuring a person's ability to demonstrate propositional knowledge or performance skill. For example, how well does someone answer probes such as 'What's the difference between a solution and an emulsion?' or 'Name the model of each foreign aircraft in this sequence of images'. The solution-emulsion question might be scored on the answer's propositional content alone, or possibly in combination with the quality of the rhetorical organization with which the content is expressed. On the other hand, the aircraft ID probe would probably be scored on a combination of speed and content accuracy because time is often critical in the imagined use domain. A third type of probe is designed to elicit a response that primarily holds evidence of psychomotor, cognitive or emotional states such as vigilance or memory or mood.

Various test-taker populations are assessed to measure different constructs, and test scores can be used in a range of contexts for various purposes. The population may be school children or job applicants or psychiatric outpatients, and the scores may be used for selection or qualification, or to monitor status or progress, or to diagnose mastery levels across a set of skills. When the measured performance is manifest in speaking, the task that elicits the spoken response may be presented in speech, or drawn figures, or animations, or text or video, or in sequences and/or combinations of these. A test item may elicit a spoken response in many ways – for example, an animation with a voice-over, or an exchange with the test-taker over several conversational turns, with or without synchronized text presented with the machine's spoken turns. Figure 3.8 provides a high-level view of how a speech recognition system works within an assessment that elicits spoken responses.

Here we describe two applications of ASR in assessment, These applications are a diagnostic assessment of early oral reading fluency for students in Grades K-5 (Moby.Read), and a listening/speaking test of Spanish as a second language (Versant Spanish Test). Another recent application example for adult populations (general and psychiatric) includes a fully automated assessment of attention or executive function (a Stroop test) described by Holmlund et al. (2023).

The ASR-based technologies used in these applications have been available for many years. For scoring segmental pronunciation, see Bernstein et al. (1990), Franco et al. (2000), Neumeyer et al. (2000), and Witt and Young (2000). For approaches to scoring prosodic quality, see Cheng (2011) and Slaney et al. (2013).

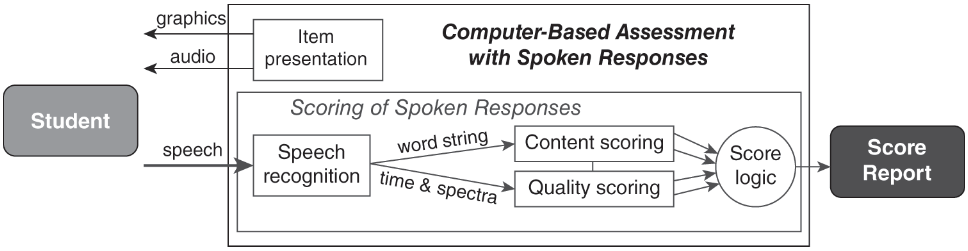


Figure 3.8 A schematic view of speech recognition and response scoring within a computer-based assessment with items that elicit spoken responses.

4.1 Oral Reading Fluency (Example: Moby.Read)

For in-school measurement of early reading skill, oral reading fluency (ORF) has been the favored construct/method to measure reading ability and track reading progress in grades K-4. In the United States, since the year 2000, reading instruction in schools has largely followed patterns that are informed by a model of reading that was set forth in a report from the National Institute of Child Health and Human Development (NICHD, 2000). Oral reading fluency is defined as *the ability to read texts aloud for comprehension with speed, accuracy, and proper expression*. ORF testing starts after students have sufficient phonemic awareness to start learning to read. After students are introduced to the most common sight words (e.g., *the*, *in*, *can*, *no*, *how*), and to the basics of phonics, the ‘instruction loop’ shown in Figure 3.9 gets started.

When the instruction loop starts, teachers direct students to read from leveled text material, and if students perform at about the level expected for their age and grade, the instruction loop continues for 3 to 5 years until students can independently read grade-appropriate academic text to learn new material and read efficiently enough to understand and enjoy age-appropriate stories and books.

However, for struggling readers who perform below expectation on benchmark tests, a teacher may administer a set of formal tests and score the student’s reading performance to determine which component reading skills may be slowing the student’s reading development. Quarterly ORF benchmark tests monitor student progress. Traditional ORF benchmark tests are administered individually by a teacher who listens to the student read aloud and scores the test for accurate reading rate (words correctly read per minute). This process takes up teacher and student time, while producing unaudited scores. Cheng (2018) described how ASR-based systems can accurately score several aspects of dysfluent readings from early readers.

Note here that the main scoring task is recognition of words spoken by a student with reference to a known text. On a short sample of leveled passage text, the range of expected spoken responses is relatively small, so the ASR’s language model is very low entropy and recognition output is accurate. We describe *Moby.Read* as an example of a very direct scoring of basic reading skill. The ASR that runs inside the Moby.Read service was built at Analytic Measures Inc. (AMI) and tuned for children’s speech. Most importantly, it was optimized for recognizing the passages in Moby.Read.

The Moby.Read system is an instrument designed, built, and validated in 2016–2019 at Analytic Measures Inc. (AMI). The Moby.Read service was first introduced commercially in January 2019. Figure 3.10 presents an overview of Moby.Read features as presented on the AMI website. Moby.Read is designed for benchmarking early reading performance three times a year. Inside the Moby.Read system, an automated speech recognizer (ASR) was optimized for children’s readings and their spontaneous speech in retelling text passages, and an augmented

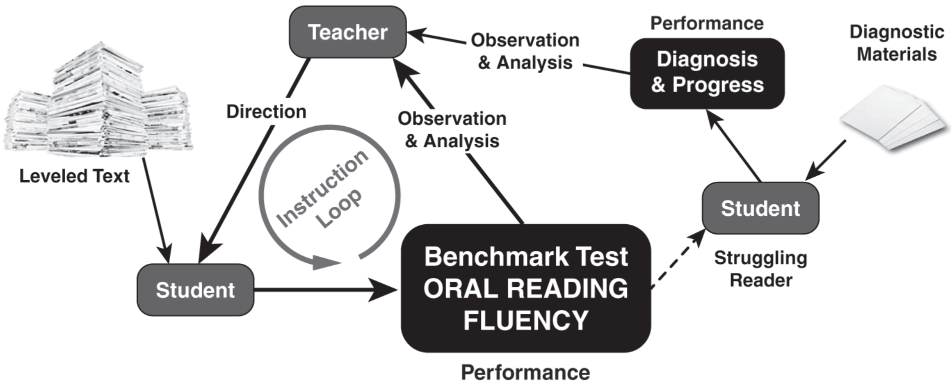
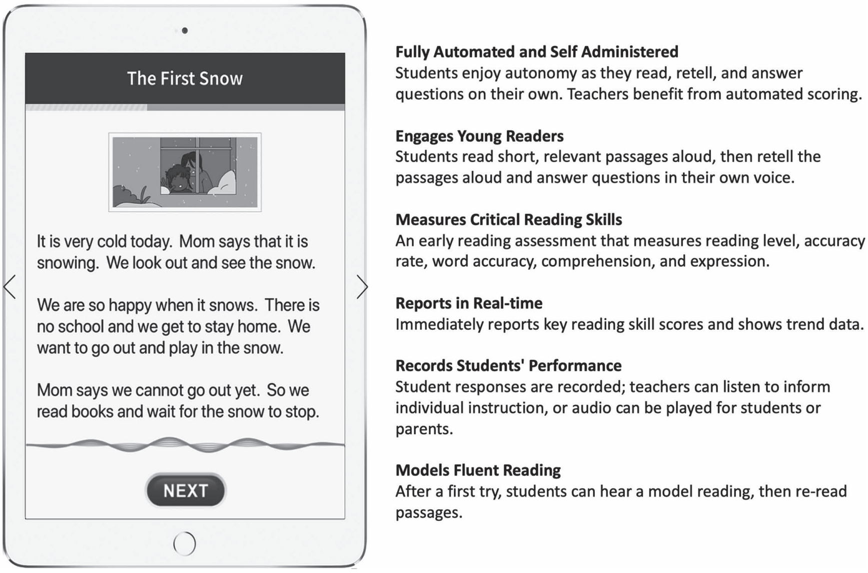


Figure 3.9 A typical reading instruction cycle in grades 1–3. The dashed line shows an added path for students who are identified as struggling readers.



It is very cold today. Mom says that it is snowing. We look out and see the snow.

We are so happy when it snows. There is no school and we get to stay home. We want to go out and play in the snow.

Mom says we cannot go out yet. So we read books and wait for the snow to stop.

Fully Automated and Self Administered
Students enjoy autonomy as they read, retell, and answer questions on their own. Teachers benefit from automated scoring.

Engages Young Readers
Students read short, relevant passages aloud, then retell the passages aloud and answer questions in their own voice.

Measures Critical Reading Skills
An early reading assessment that measures reading level, accuracy rate, word accuracy, comprehension, and expression.

Reports in Real-time
Immediately reports key reading skill scores and shows trend data.

Records Students' Performance
Student responses are recorded; teachers can listen to inform individual instruction, or audio can be played for students or parents.

Models Fluent Reading
After a first try, students can hear a model reading, then re-read passages.

Figure 3.10 A sample item-presentation page with a listing of features and functions.

Source: www.analyticmeasures.com/moby-read.

NLP module enables immediate score reporting of five key oral reading fluency skills: Reading Level, Accuracy, Accurate Reading Rate (in words correct per minute), Comprehension, and Expression. During the assessment, students read passages aloud, summarize the passage content, and respond to short answer questions using their own voice. Moby.Read embeds model readings and opportunities for students to ‘go back’ and re-read a passage to hone their oral reading skills (see Figures 3.10 and 3.11). An assessment takes about 12 minutes and runs in Chrome or as a native app on iPads.

As of this writing, one can take a demo version of the current Moby.Read assessment on the AnalyticMeasures.com website, or one can watch a demonstration of the test administration at https://youtu.be/_V6_7agY5tc. In each test administration session, the Moby.Read ORF assessment collects 18 spoken responses from each student. The responses are elicited in the sequence shown in Figure 3.11.

Of the 18 responses, only 12 are scored – the second, third, and fourth passages, and, with each, its associated retelling and student answers to two direct questions about the content of the passage. The reported scores for accurate rate, word accuracy, and expression are based on the three readings (each about 60 to 90 seconds long), and the comprehension score is based on the spoken retelling of the passage and the spoken answers to the two questions.

Moby.Read automatically scores and reports the reader's Accuracy, Accurate Reading Rate, Comprehension, and Expression with high accuracy from a student's reading of three short passages, as shown in Figure 3.12.

Moby.Read augments the fluency measures with comprehension measures based on spoken retellings and constructed answers to short questions. AMI has developed SLP modules and scoring algorithms that have achieved excellent accuracy in measurement of oral reading rate, accuracy, and expression. The SLP modules have been combined with AMI's NLP algorithms to measure reading comprehension from spontaneous spoken retellings and spoken answers to comprehension questions.

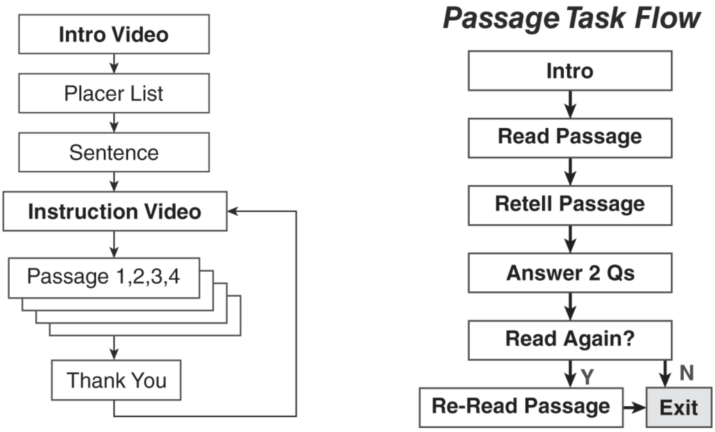
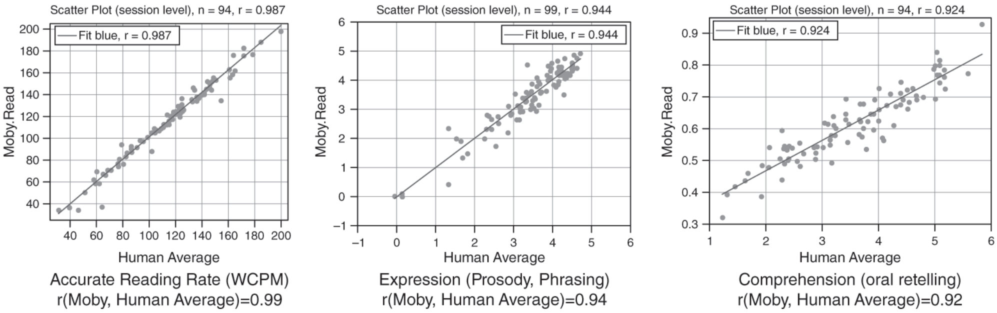


Figure 3.11 The left-hand figure shows the flow of a test session, in which a student sees an introduction video, then reads a list of words and a single sentence aloud. Following this, the student sees a video with more specific instructions on the passage-reading parts of the test and then reads four passages. The task flow for each passage reading is shown in the right-hand figure: passage intro, read aloud, retell aloud, answer two questions aloud, optional re-read.



Accurate Reading Rate (WCPM)
r(Moby, Human Average)=0.99

Expression (Prosody, Phrasing)
r(Moby, Human Average)=0.94

Comprehension (oral retelling)
r(Moby, Human Average)=0.92

Figure 3.12 Accuracy of Moby.Read vs. human scoring on three primary oral reading measures.

Moby.Read automatic scoring covers accurate reading rate, expression, and comprehension, with accuracy sufficient to match the reliable information in human scores. Moby.Read and other recent automated ORF assessments (e.g. Balogh et al., 2012; Bernstein et al., 2020) have simplified the quarterly benchmarking of grade school reading achievement. However, these scores are summative and serve mainly to indicate which students are struggling and may need extra help with reading. Many struggling students (perhaps 10–20% of the total cohort) perform below expectation on the benchmark tests and go into the right-hand diagnostic path of the instruction cycle shown in Figure 3.9. The struggling readers then take one or more specialized diagnostic test batteries to guide reading remediation, adding expense and delaying needed instructional interventions.

Administering diagnostic tests occupies student time, may require specialist time, and may involve other logistic costs or publisher fees. Emerging extensions to automated ORF testing can remove some of the cost of diagnostic testing, by extracting specific diagnostic information from student performances on the benchmark ORF tests. Now that the rich information produced by students reading out loud during automated ORF benchmark assessments is being recorded, it can be analyzed to guide instruction, which should reduce the need for follow-on rounds of diagnostic tests to guide reading remediation.

Skilled teachers and reading specialists who observe and listen to a young student reading passages aloud can hear the reader's skills and infer a profile of likely reading difficulties to guide remediation. Augmented ASR output includes the events in oral reading that lead experienced reading teachers to make these skill and deficit inferences, so supervised machine learning can associate event pairs (such as {recorded readings, expert judgments}) to automate skill diagnosis from oral readings of passage text. When we can automatically derive a profile of reading strengths and difficulties from recordings of a student reading several short passages aloud, we can greatly reduce the diagnostic testing burden and help teachers focus on those reading difficulties their students encounter when reading grade-level prose. Figure 3.13 shows the initial development process that AMI applied to train models to extract accurate performance skill profiles from oral passage readings. Known passages, leveled but not designed for diagnostic use, were read aloud by students and were analyzed by NLP routines. The NLP produced augmented representations of the passage texts (text analytic data) that included a

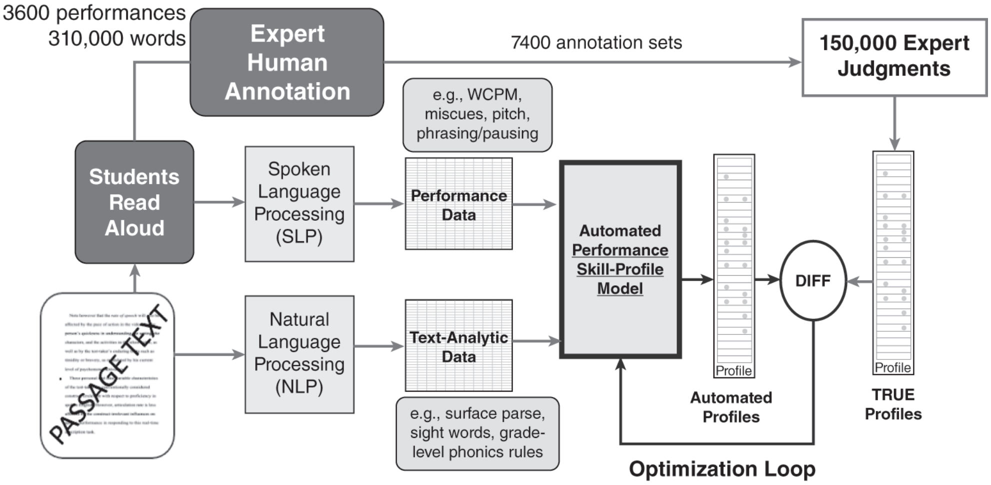


Figure 3.13 Processes that produce data to feed the optimization of a scoring model to support diagnostic measurement of reading subskills from an oral reading of a set of passages.

surface parse that grouped the words into phrases, and a vector associated with each word that identified the phonics rules and morphological rules a student would need to know to decode the word's pronunciation from the letter sequence. The text-analytic data also indicated, when appropriate, the grade level at which the word would be considered a 'sight word' that readers are expected to recognize without decoding the letter sequence.

In Figure 3.13, the optimization loop is shown with red arrows. In this case, the skill profiles distilled from the expert human annotations of the student readings are the target truth that the optimization is designed to match, minimizing the difference (DIFF) between a set of automated skill profiles and the TRUE skill profiles.

Going back in time, the first large-scale, ASR-based ORF-test scoring was in the *Fluency Addition to NAAL* (FAN) which was conducted in 2003. *NAAL* is the National Assessment of Adult Literacy, conducted by the National Center for Education Statistics (NCES) of the U.S. Department of Education. The NAAL population was a representative sample of the U.S. adult population, and the FAN test was administered to 20,000 of the lowest-performing participants in the NAAL survey. The core of FAN was an ORF test, which was scored using ASR. Balogh et al. (2012) audited the ASR scoring of these struggling adult readers for accuracy and for bias. Analysis showed that the scores were consistently accurate and did not show significant bias against the speakers of AAVE (African American Vernacular English) or against the native speakers of Spanish in the sample. A similar ASR-based system scored an NCES special study of oral reading as part of the fourth-grade reading section of the 2018 NAEP (National Assessment of Education Progress), and again validation analyses confirmed high accuracy for the ASR scoring (White et al., 2020) and revealed important new insights into the characteristics of 'Below Basic' fourth-grade readers.

4.2 Second Language Speaking Ability (Example: Versant Spanish Test)

The ability to understand and speak a language is sometimes a key factor in the qualification or selection of candidates for employment or for entry to educational programs. This is often the case when the language of the business or the school is different from the first or principal language of an applicant. Two well-known examples are the TOEFL and the TOEIC tests from the Educational Testing Service. Traditional tests often focused on listening, reading, vocabulary, and knowledge of prescriptive syntax, in part because these skills were simplest to test in a paper-and-pencil format. Without ASR technology, measurement of speaking skill posed logistical problems and/or significant expense.

The first available automated assessment of second-language speaking ability was the Versant English Test. The Versant English Test was originally offered in 1999 under the name PhonePass, and the format and scoring logic of the original PhonePass English speaking test has been applied to build Pearson's Versant-branded tests in Spanish, Dutch, Arabic, Chinese, and French, and has been extended to build the Pearson Test of English Academic (see www.pearsonpte.com).

The overall construct of the Versant tests is *facility* with the spoken language, defined as *the ability to understand spoken Spanish and speak appropriately in response at a native-like pace on everyday topics*. This definition of the facility construct adds timing (*at a native-like pace*) to the more traditional 'speaking proficiency' construct (Bachman, 1990), and it limits the intended range of vocabulary and specialized usages (*everyday topics*), although analysis of the Versant validation data suggests that all the reliable variance in a spoken language proficiency assessment will be predictable from a measure of facility.

Facility should be closely related to successful participation in native-paced discussions, and so it includes both listening and speaking skills, emphasizing the test-taker's facility (ease, accuracy, fluency, alacrity) in responding to decontextualized linguistic material (material the test-taker cannot anticipate) constructed from common conversational vocabulary in common phrase and clause structures. The Versant tests focus on core linguistic knowledge (phonology, lexicon, morphology, phrase structure, and clause structure), and the basic psycholinguistic abilities: speech comprehension and production.

In this chapter, we describe the *Versant Spanish Test* (Pearson Education, 2011) for two reasons:

1. It has been validated against concurrent human scores in several different ways.
2. It was completed, validated, and fielded for high-stakes use in 2004, and the accuracy of speech recognition has increased significantly since that time.

The release date of the Versant Spanish Test (VST) and the PhonePass English Test (1999) is important because by 1999, these tests already returned highly accurate, construct-relevant scores on the speech of nonnatives transmitted over random limited-bandwidth (nominally 300–3,300 Hz) telephone connections. If a reasonably good ASR system is used, then there should be no need for trepidation in deploying a high-stakes language test that returns scores based wholly, or in part, on speech recognition. Since these early tests were developed and fielded, other ASR-based spoken language tests have been published online, including Pearson's *TELL* assessment (Bernstein et al., 2013; Pearson Education, 2014, 2022) that works via omnidirectional microphones in a classroom setting for children as young as 4 years old who are learning English as a second language, and the AZELLA test (Cheng et al., 2014), which was validated for administration to young children over speaker-phone connections.

4.3 Versant Spanish Test – Design and Development

The VST presents a series of spoken prompts in Spanish at a conversational pace and elicits oral responses in Spanish. A test administration takes about 13–17 minutes to complete and yields 52 or 54 recorded spoken responses from a test-taker, which typically contain 2 to 5 minutes of actual test-taker speech. The voices for the prompts are from native Spanish speakers from different countries, providing a range of native accents and speaking styles. As shown in Figure 3.14, the VST has seven sections: Reading, Repeats, Opposites, Short Answer Questions, Sentence Builds, Open Questions, and Story Retelling. For each item type, Table 3.1 gives a text version of an example item.

All items in the first five sections elicit relatively short responses (fewer than 20 words) that are analyzed automatically, with each providing multiple, fully independent measures of skills that underlie facility with spoken Spanish. For example, the production of each linguistic element in a repeated sentence yields information about intelligibility, phonological fluency, receptive speech processing, vocabulary, and pronunciation of rhythmic and segmental units. Conversely, because more than one task type contributes to each subscore (pronunciation, fluency, sentence mastery, vocabulary), there are many hundreds of performance measures available in the response recordings of the use of multiple item types that maximize score reliability.

VST items were designed to be region neutral so that both native speakers and proficient nonnative speakers would find the items easy to understand. Each VST item is independent of the other items and presents unpredictable spoken material in Spanish. Context-independent material is used in the test items because:

1. Context-independent items measure the most basic meanings of words, phrases, and clauses on which context-dependent meanings are based (Perry, 2001).
2. When language usage is relatively context-independent, task performance depends less on construct-irrelevant characteristics and more on the test-taker's facility with the language.
3. Context-independent tasks maximize response density, so the test-taker spends more time speaking in responses and less time developing background cognitive frames for items.

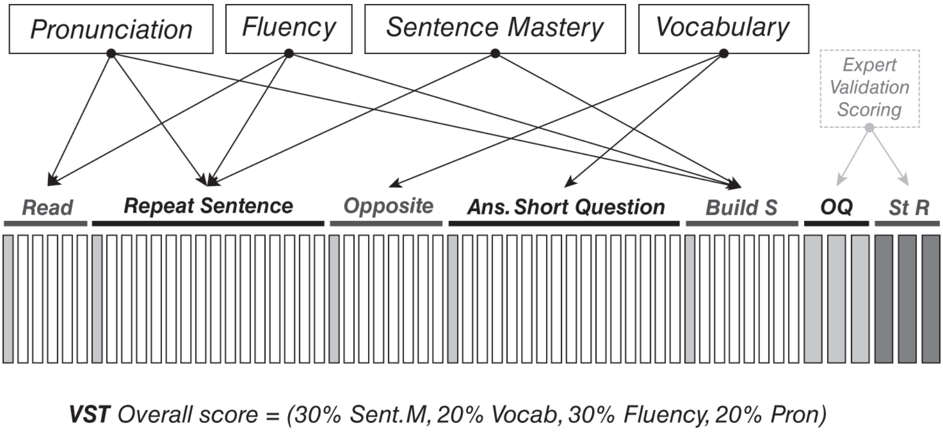


Figure 3.14 Sequence of item types and item presentations in the original Versant Spanish Test. Each tall rectangle is an item presentation. Unscored items are filled in gray. The last item type (Story Retell) was not machine scored in the 2004 form, but since 2007, two Retells are presented and machine scored. The remainder of the VST has not changed since 2004.

Table 3.1 An Example Item for Each of the Seven Item Types in the Versant Spanish Test.

Item TYPE	Example Item
Reading	*Un día, al no encontrarla, creyó que se la habían robado.*
Repeats	*Le gusta cantar canciones románticas.*
Opposites	*subir*
Short Answer Qs	*¿Cuántas patas tiene un perro?*
Sentence Builds	*de la mesa/el plato/recogió*
Open Question	*¿Prefiere usted vivir en la ciudad o en el campo? Por favor explique su elección.*
Story Retell	*Tres niñas caminaban a la orilla de un arroyo cuando vieron a un pajarito con las patitas enterradas en el barro. Una de las niñas se acercó para ayudarlo, pero el pajarito se fue volando, y la niña terminó con sus pies llenos de barro.*

Note: For more detail on the items, see the VST validation summary (Pearson, 2009).

The VST was originally developed for the U.S. government to screen applicants and qualify employees for work that requires fluent, accurate understanding of spoken Latin American Spanish, but later releases of the VST include item voices from Spain and have been validated with Spanish speakers from Spain.

The process flow in Figure 3.15 starts with the *Test Spec* and finishes with two paths leading to Validation. Starting in the lower left of Figure 3.15, the test content (item recordings, timings, scripted presentation sequences) is prepared by a group of native developers, and then uploaded to form a test that runs from the Versant Test database. The vocabulary used in the test items and responses was restricted to forms of the 8,000 most frequent words in the Spanish Call Home corpus (LDC, 1996). The 8,000 most common lexemes (headwords or lemmas) were used to create a base lexicon, and a small number of other related words were included for completeness. VST items were drafted by two Argentine item developers. The items were designed to be independent of social nuance and higher cognitive operations. Draft items were reviewed to ensure that they conformed to current colloquial Spanish usage according to reviewers in Chile, Colombia, Ecuador, Mexico, Puerto Rico, Spain, and Venezuela. The changes proposed by the different reviewers were then reconciled and edited accordingly.

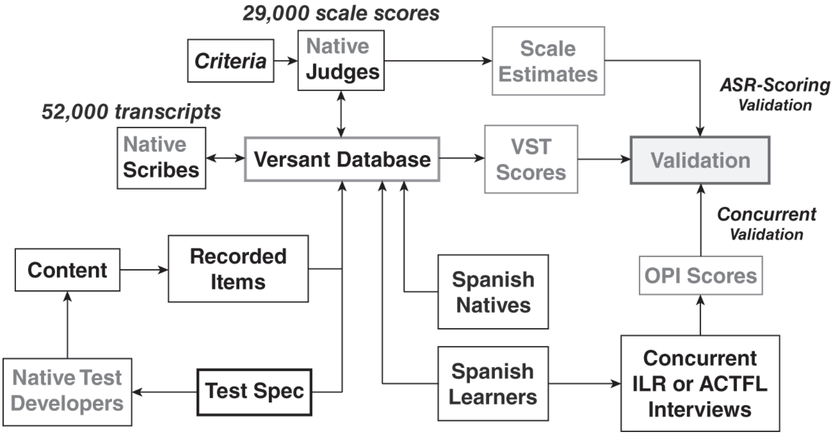


Figure 3.15 The development flow and validation points of the Versant Spanish Test.

Expert judgment was used initially to define correct answers to Short Answer Question items. Many of the items have multiple answers that are accepted as correct. Native speakers from six different Spanish-speaking countries recorded the spoken materials. Instructions were given in an 'examiner voice' that was quite distinct from the item voices. All questions were pre-tested on diverse samples of native and nonnative speakers. For an item to be retained in the test, it had to be understood and responded to appropriately by at least 80% of a reference sample of educated native speakers of Spanish. The PhonePass English test items had reached a 90% correct threshold for inclusion.

Of the 58 or 60 presented items in an administration of the Versant Spanish Test, 49 or 51 responses are used in the automatic scoring. The Versant Spanish Test returns an overall score and four subscores, each of which is reported in the range from 20 to 80. Versant Overall scores have a confidence interval of about ±3 points, which suggests that scores range over about 20 meaningful levels.

4.4 VST Validation

The validation of this ASR-based test hinges on comparison with expert human scoring. In the VST case, the expert listeners make judgments with respect to defined rubrics, which are then compared to corresponding machine scores from the same test-takers. Over 1,000 test-takers participated in a series of validation experiments.

Before that comparison, we can simply observe cumulative VST score distributions showing that native Spanish speakers (male and female) are clustered at the high end of the score scale, whereas Spanish language learners (male and female) are distributed across a wide range of scores.

As shown in Figure 3.16, the distribution of the native speakers clearly distinguishes the natives from the nonnative sample. For example, fewer than 5% of the native speakers score below 75, and only 10% of the nonnative speakers score above 75. Note that underlying scale scores above 80 are reported as 80, so scores of 80 and above are shown with a gray overlay. This finding suggests that the VST has high discriminatory power among learners of Spanish as a second or foreign language, whereas native speakers obtain near-maximum scores. Further analysis has shown that the same patterns hold true regardless of gender or age, and across national dialects, including dialects that were not included in the training datasets.

VST reliability supports validity. The split-half reliability of the Overall score of the Versant Spanish Test is 0.97 (*N* = 267, with a standard error of 2.6 points). The reliability values

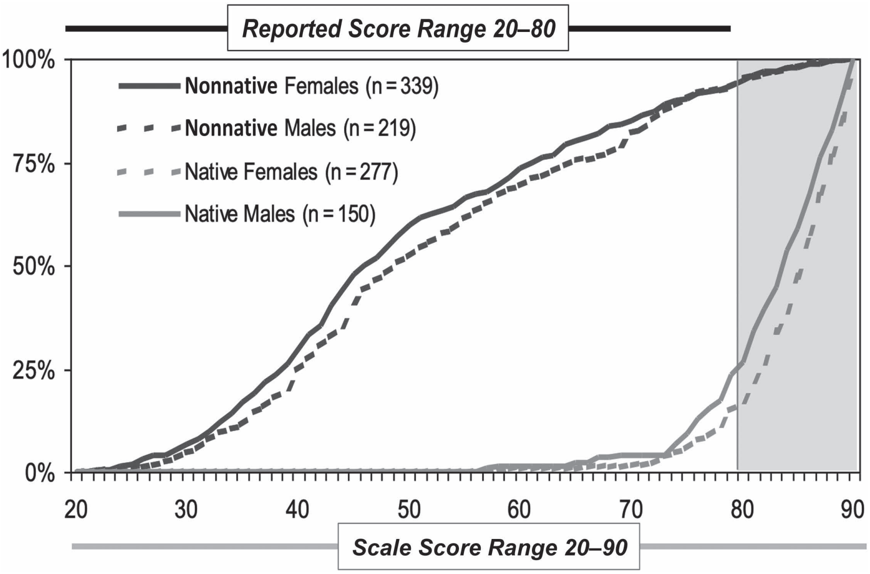


Figure 3.16 Cumulative distributions of VST scores for native and nonnative speakers.

Table 3.2 Split-Half Reliability for Versant Spanish Test Scores ($N = 267$) Shown for Native and Nonnative Group Performance

VST Score	Reliability
OVERALL	0.97
Vocabulary	0.91
Pronunciation	0.94
Sentence Mastery	0.95
Fluency	0.93

are corrected for split-half underestimation. Table 3.2 lists these reliabilities for each reported VST score.

The principal goal of the validation studies was to understand the relation of Versant Spanish Test scores to the scores obtained using well-documented human-mediated measures of oral proficiency and expert human estimates of test-taker performance using the well-established language proficiency scales. To accomplish this, machine-generated VST scores were compared with scores from other well-accepted human-rated assessments of spoken Spanish and with scores assigned by sets of expert raters after listening to recorded speech samples from the VST itself. Test-takers did not take a practice test prior to taking the VSTs administered in the validation studies. In any event, unpublished research indicated that VST performance is not improved by taking a practice test.

4.5 Concurrent Score Data

Three sets of participants were required for the validation studies: a group of Spanish native speakers (as test-takers), a group of nonnative speakers (as test-takers), and trained human raters to assess recorded speech samples and to conduct Oral Proficiency Interviews (OPIs).

Adult native Spanish speakers (18 years old or older) were recruited for norm-referencing and test validation. Native speakers were roughly defined as individuals who spent the first 20 years of their lives in a Spanish-speaking country, were educated in Spanish through college level, and currently reside in a Spanish-speaking country. Samples were gender balanced when possible. Four hundred twenty-two candidates constituted the native Spanish speaker sample: 135 from Argentina, 36 from Colombia, 217 from Mexico, 21 from Puerto Rico, and 13 from other Latin American countries. In addition, 153 native Spanish speakers from Spain were recruited for further validation, bringing the total to 575 native speakers.

For the *nonnative Spanish speaker* sample, the Versant Test Development team contacted a number of Spanish departments at universities in the United States asking them to have students take the Versant Spanish Test and, if possible, an official Spanish OPI certified by the American Council on the Teaching of Foreign Language (ACTFL). Students/universities were remunerated for their participation and for the ACTFL test fee. In addition, test-takers were recruited from the military and other institutions. A subset of each group took one of two oral interview tests (ACTFL OPI or SPT-Interview-ILR). A total of 574 nonnative speakers participated in the experiment.

Expert human raters. The validation experiments called for several groups of human raters to perform the proficiency interviews and to analyze spontaneous speech files collected from the last two tasks (Story Retellings and Open Questions). Two groups of raters conducted two types of oral interviews:

1. Raters from Language Testing International (LTI) administered all the certified Oral Proficiency Interviews (OPI) for the American Council on the Teaching of Foreign Languages (ACTFL). The ACTFL OPI interviews were all conducted by telephone. LTI does the official ACTFL testing. (The scores from these interviews will be referred to as *ACTFL OPI*.)
2. Two government-certified raters (contractors to the FBI) administered telephone Oral Proficiency Interviews according to the Spoken Proficiency Test (SPT) procedure, with scores reported on the ILR scale. Both raters had experience administering official SPTs in Spanish, and both were female – one from Peru and one from Colombia. (Scores from these interviews will be identified as *SPT-Interview-ILR*.)

In addition, human raters were recruited to analyze a total of six 30-second recorded responses elicited by the Open Questions and Story Retellings at the end of each Versant Spanish Test. These six recordings are referred to as the 30-second response samples. Human ratings of the 30-second response samples were supplied by three rater sets.

Three native speakers of Spanish were selected to listen to the 30-second response samples and assign level descriptors to them based on the Common European Framework (CEFR). All had degrees from universities in South America. Two of the raters were certified Spanish translators/interpreters. Raters received training in the CEFR level rubrics prior to engaging in the rating tasks and were tracked during the rating process to ensure that they were following defined rubrics. Training included rating subjects not used in the main study until a predetermined level of agreement was reached. (The scores generated from these raters will be referred to as *CEFR Estimates*.)

Four government-certified ILR interviewers listened independently to the six 30-second response samples from each of 166 test-takers. Two raters were active in testing at the Defense Language Institute (DLI) in Monterey, California, and provided estimated ratings based on the ILR scale descriptors. (These ratings will be identified as *ILR-Estimate/DLI*.) The other two

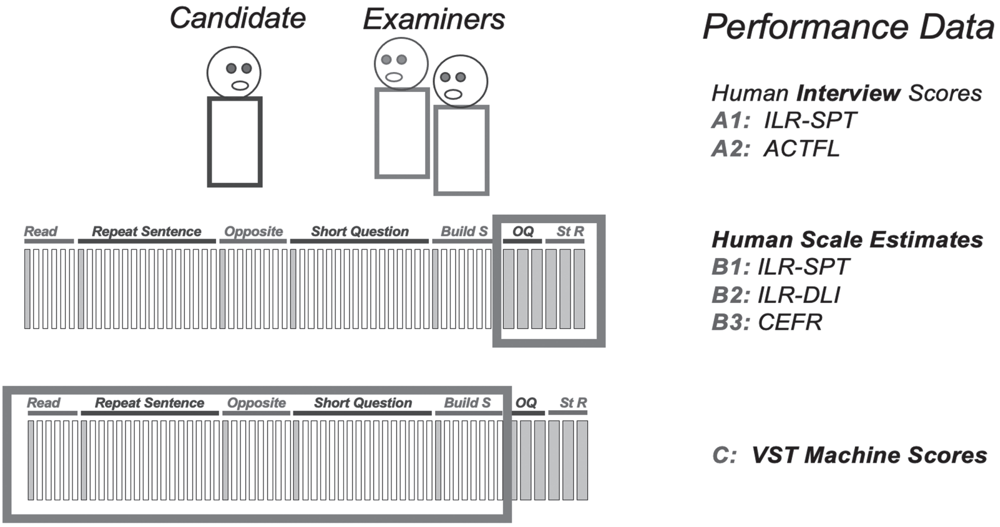


Figure 3.17 Three performance types and a key to the scores that are compared in Table 3.3.

government-trained raters were the same two people who administered the SPT-Interview-ILR. After a pause of two weeks, these raters also listened independently to the 30-second response samples from the Versant Spanish Test administrations and provided estimated ratings based on the ILR scale. (These ratings will be referred to as *ILR-Estimate/SPT*.) These 996 (6 x 166) 30-second response samples included the samples from the 37 test-takers that they had interviewed.

As shown in Figure 3.15, both native Spanish speakers and the nonnative speakers took the Versant Spanish Test. The VST data consisted of all automatically generated scores and subscores, along with recorded 30-second response samples from the Open Questions and Story Retellings. The VST Overall scores for native Spanish speakers were compared with the scores of nonnative speakers and were correlated with several sets of human scores and human ratings. These sets of pairs were taken from one of three kinds of performances: an interview score, a scale estimate from the open question and story retelling recordings, or a machine score from the responses to the first five item types. For any participant in the validation studies, these three performances are disjoint, as shown in Figure 3.17. The figure also gives labels to the datasets that are compared; for example, A1 is a score from a Spoken Proficiency Test interview scored on the ILR scale, and B2 is an ILR-scale score estimated from six 30-second recordings from the final two item types in a VST administration.

The A1 and A2 scores are ‘gold standard’ scores based on a carefully specified interview that typically lasts between 25 and 35 minutes, and which is based on scores by the two interlocutors who conduct the interview, with reference to time-tested rubrics that describe spoken performance at several levels in each of several dimensions. The B1, B2, and B3 scores are scale scores estimated from 12 scores, which comprise one score from each of two human raters to each of the six 30-second spontaneous responses to the three Open Questions (OQ) and the three separate Story Retellings (StR) – scored independently in different random orders. The C scores are the VST machine scores derived from the responses to the first five item types (read, repeat, opposite, short question, sentence build), which on average comprise about 3 minutes of speech from the test-taker.

The human scores shown in Figure 3.18 are:

ACTFL OPI – For the ACTFL interviews, the Versant Test Development team coordinated with universities to have their students take an ACTFL interview within a day of the Versant Spanish Test. The standard ACTFL interview was administered with at least two official ACTFL ratings per interview. ACTFL submitted 52 scores, one for each of the 52 participants.

SPT-Interview-ILR – For the SPT interviews, the candidates were asked to take the Versant Spanish Test within one day of the SPT interview. The raters followed the ILR level descriptions that appear in the Test Manual of the Speaking Proficiency Test developed by the Federal Language Testing Board. Each rater independently provided ILR-based proficiency level ratings for each of the 37 candidates, for a total of 74 ratings.

CEFR-Estimate ratings – CEFR ratings were based on the Common European Framework level descriptors. For the CEFR Estimates, 30-second response samples from 572 test-takers were rated. On average, the three raters together provided 11 independent scores for each test-taker, resulting in a total of 6,125 ratings.

ILR-Estimate/DLI ratings – These ratings were based on the ILR/OPI rubrics. Nine hundred ninety-six 30-second response samples from 166 test-takers were scored. The two raters provided a total of 1,978 ratings, with an average of 12 independent scores for each test-taker.

ILR-Estimate/SPT ratings – These ratings were based on the rubrics in the Tester manual of the Speaking Proficiency Test developed by the Federal Language Testing Board. Nine hundred ninety-six 30-second response samples from 166 test-takers were scored. On average, the two raters provided 19 independent scores for each test-taker, resulting in 2,798 ratings total.

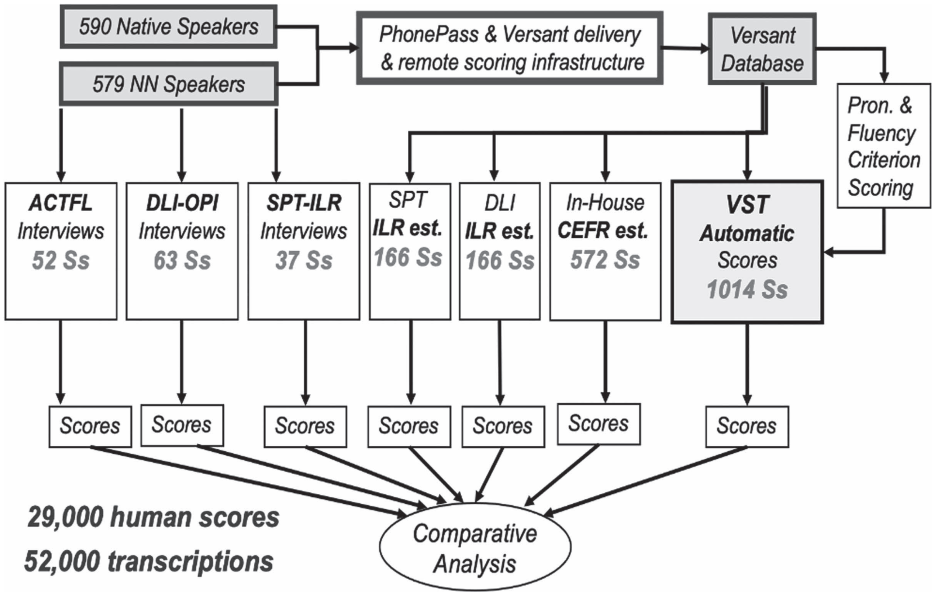


Figure 3.18 The array of scores gathered and compared in the VST Validation studies (Ss are students).

Table 3.3 Correlations Between Different Measures of Oral Proficiency

	Versant Spanish Test	ACTFL OPI	SPT-Interview-ILR	CEF Estimate	ILR-Estimate/DU	ILR-Estimate/SPT
Versant Spanish Test		0.86	0.92	0.90	0.88	0.89
ACTFL OPI	0.86		–	–	0.87	0.89
SPT-Interview-ILR	0.92	–		–	0.92	0.94
CEF Estimate	0.90	–	–		0.91	0.91
ILR-Estimate/DLI	0.88	0.87	0.92	0.91		0.96
ILR-Estimate/SPT	0.89	0.89	0.94	0.91	0.96	

4.6 Results

Table 3.2 summarizes all the Machine-Human correlation results for the ratings described previously, in addition to correlations between human-rated scores from different raters, on different material, and with reference to different rubrics. Some of the comparisons are not possible because the data do not overlap – for example, no respondent participated in two oral proficiency interviews, such as the ACTFL OPI and the SPT-Interview-ILR.

The range of correlation coefficients between the VST test and other assessments of oral proficiency is from 0.86 to 0.92. These are all statistically significant correlations. In addition, all the correlations between the oral proficiency interviews and the VST test are nearly identical to the interview's correlation with human-rated estimates of other measures. For example, in Table 3.3, the VST scores correlate with the ILR scores from the SPT-Interview-ILR with a correlation coefficient of 0.92, while the ILR estimates from the two pairs of certified ILR interviewers correlated with the actual SPT-Interview-ILR scores with coefficients of 0.92 and 0.94. Thus, the VST procedure elicits sufficient spoken language behavior on which to base a reasonably accurate human judgment of practical speaking and listening skills. The VST test is also producing results that are similar to results from human raters.

The same group at Pearson designed, constructed, and validated Versant tests in several languages. A report by Bernstein et al. (2010) summarized the validation process and results across languages. Correlation results for Spanish, Dutch, Arabic, and English are shown in Table 3.4.

5. Transparency and Bias

Although there is often no precise understanding of the mental process by which human judges rate a spoken performance for correctness of content or quality of exposition, assessment stakeholders accept the human judgments if they are given with reference to clear scoring rubrics and they show reasonable agreement between judges working independently (cf. Nisbett & Wilson, 1977). Stakeholders in 2022 are sometimes skeptical of assessment scoring based on ASR analysis mixed with undocumented end-to-end optimized scoring processes, and correlation with gold-standard human scoring is not sufficient to put skeptics at ease. One possible response to 'black box' skeptics is to build up scores from specific rubric-based component scores, each one of which is validated separately before their combination into reported scores.

Taking this extra step can support a clearer, more understandable explanation of the scoring logic and its validity argument. For example, the *overall* score returned by a test of spoken

Table 3.4 Validation Data for Automated Second Language Tests

Dataset	Number of test-takers	Test-taker sample	Automated test	Automated test administrations	Human test	Human Test administrations (total human scores)	Observed human test score range	Correlation	Citation
1	37	Convenience sample	Versant Spanish	1	ILR-SPT interviews	1 (2)	ILR 0+ to 4	0.92	Balogh and Bernstein (2007)
2	228	Netherlands immigrants, near CEFR AI	Toets Gesproken Nederlands (TGN)	2	CEFR OPI and career interviews	2 (3)	CEFR AI – and AI	81% classification consistency	De Jong et al. (2009)
3	118	Convenience sample	Versant Arabic	2	ILR-OPI interviews	2 (4)	ILR 0 to 4	.87	Pearson (2009), Cheng et al. (2009)
4	151	Students in Adult Education ESL Classes	Versant English	2	Best+ tests	2 (2)	BEST Plus 337 to 961	0.81–0.86	Present-Thomas & Van Moere (2009)
5	130	TOEFL test-takers in Iran	Versant English	1	IELTS interviews	2 (4)	IELTS 2.75 to 9	0.77	Farhady (2008)

Source: From Bernstein et al. (2010).

language proficiency can be formed as a composite and may be justified with reference to an explicit combination rule or weighting that operates on more specific, elemental performance attributes. The intermediate subscores for these more specific attributes can then each be validated against specific human judgments of separable performance elements that have specific rubrics and separable manifestations in speech signals. A composite overall spoken proficiency score might be built up from individually validated elemental scores, as schematized in Figure 3.19.

The idea behind Figure 3.19 is that a test publisher/provider can describe the variables that the machine uses to derive the Proficiency Element scores and publish the rubrics and rater qualifications for human ratings that serve as concurrent scores to validate machine scores for each element. Finally, if the rules used to combine the Proficiency Elements into reportable score are made public, then the process seems more open to audit than many commonly used human scoring regimens.

There is also evidence that some of the most commonly used ASR systems in 2019 (Apple, Google, Amazon, Microsoft, IBM) are more accurate in recognizing speech from some demographic groups than in recognizing speech from members of other groups (see, for example, A. Koenecke et al., 2020). This result is not necessarily surprising, given significant disparities in the datasets used in the Koenecke et al. paper to test the systems. It is altogether possible that these same systems can reach intergroup parity after retraining with more representative speech data and a more uniform test data collection procedure, as was found by Balogh et al. (2012) on a large sample of relatively uneducated, native and nonnative North American speakers of English. Demographic group bias is avoidable in ASR technology and in its applications for assessment.

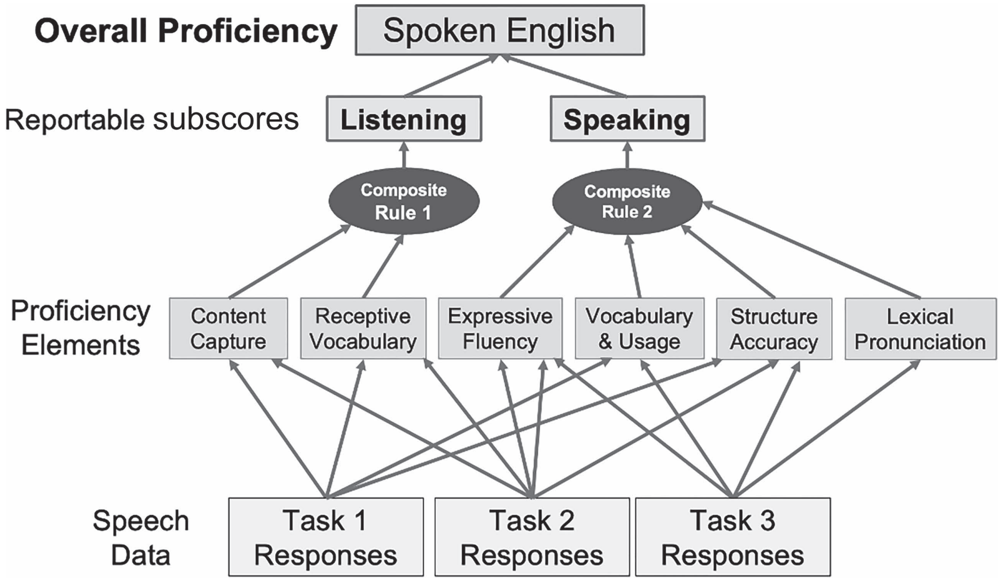


Figure 3.19 A hypothetical example of a spoken language proficiency score that is assembled from validated elemental scores; thus, it may provide more transparency into logic and process.

References

Bachman, L. (1990). *Fundamental considerations in language testing*. Oxford University Press.

Bai, Z., & Zhang, X. (2021). *Speaker recognition based on deep learning: An overview*. Elsevier.

Balogh, J., & Bernstein, J. (2007). Workable models of standard performance in English and Spanish. In Y. Matsumoto, D. Y. Oshima, O. R. Robinson, & P. Sells (Eds.), *Diversity in language: Perspective and implications* (pp. 20–41). Center for the Study of Language and Information Publications.

Balogh, J., Bernstein, J., Cheng, J., Van Moere, A., Townshend, B., & Suzuki, M (2012). Validation of automated scoring of oral reading. *Educational & Psychological Measurement*, *72*(3), 435–452.

Bernstein, J., Cheng, J., Balogh, J., & Downey, R. (2020). Artificial intelligence for scoring oral reading fluency. In H. Jiao & R. W. Lissitz (Eds.), *Applications of artificial intelligence to assessment* (pp. 51–76). Information Age Publishing.

Bernstein, J., Cohen, M., Murveit, H., Rtischev, D., & Weintraub, M. (1990). *Automatic evaluation and training in English pronunciation* (pp. 1185–1188). Proc. ICSLP-90: 1990 Int. Conf. Spoken Lang. Proc. Kobe, Japan.

Bernstein, J., Todic, O., Neumeyer, K., Schultz, K., & Zhao, L. (2013). *Young children's performance on self-administered iPad language activities* (pp. 24–25). Proceedings of SLaTE 2013.

Bernstein, J., Van Moere, A., & Cheng, J. (2010). Validating automated speaking tests. *Language Testing*, *27*(3), 355–377.

Cheng, J. (2011). *Automatic assessment of prosody in high-stakes English tests* (pp. 1589–1592). *Proceedings of Interspeech*.

Cheng, J. (2018). *Real-time scoring of an oral reading assessment on mobile devices* (pp. 1621–1625). *Proceedings of Interspeech*.

Cheng, J., Bernstein, J., Pado, U., & Suzuki, M. (2009). Automatic assessment of spoken modern standard Arabic, 1–9. 10.3115/1609843.1609844.

Cheng, J., Zhao D'Antilio, Y., Chen, X., & Bernstein, J. (2014). *Automatic spoken assessment of young English language learners* (pp. 12–21). Proceedings of the Ninth Workshop on Innovative Use of NLP for Building Educational Applications.

CMU (2022). *The CMU pronouncing dictionary*. www.speech.cs.cmu.edu/cgi-bin/cmudict

Das, R. K., Tian, X., Kinnunen, T., & Li, H. (2020). The attacker's perspective on automatic speaker verification: An overview. *arXiv*. https://arxiv.org/abs/2004.08849

De Jong, J. H. A. L., Lennig, M., Kerkhoff, A., & Poelmans, P. (2009). Development of a test of spoken Dutch for prospective immigrants. *Language Assessment Quarterly*, *6*(1), 41–60.

Farhady, H. (2008). *Human operated, machine mediated, and automated tests of spoken English*. Paper presented at the American Association of Applied Linguistics.

Franco, H., Neumeyer, L., Digalakis, V., & Ronen, O. (2000). Combination of machine scores for automatic grading of pronunciation quality. *Speech Communication*, *30*(2–3), 121–130. https://doi.org/10.1016/S0167-6393(99)00045-X

Hansen, J. H., & Hasan, T. (2015). Speaker recognition by machines and humans: A tutorial review, *IEEE Signal processing magazine 32*(6), 74–99.

Holmlund, T., Cohen, A., Cheng, J., Foltz, P., Bernstein, J., Rosenfeld, E., Laeng, B., & Elvevåg, B. (2023). Using automated speech processing for repeated measurements in a clinical setting of the behavioral variability in the stroop task. *Brain Science*, *13*(3), 442. https://doi.org/10.3390/brainsci13030442; https://www.mdpi.com/2076-3425/13/3/442

Irum, A., & Salman, A. (2019, February). Speaker verification using deep neural networks: A review. *International Journal of Machine Learning and Computing*, *9*(1).

Jelinek, F. (1998). *Statistical methods for speech recognition*. MIT Press.

Jurafsky, D., & Martin, J. (2009). *Speech and language processing*. Pearson.

Jurafsky, D., & Martin, J. (2021). *Speech and language processing* (3rd ed.). https://web.stanford.edu/~jurafsky/slp3/

Koenecke, A., Nam, A., Lake, E., Nudell, J., Quartey, M., Mengesha, Z., Toups, C., Rickford, J., Jurafsky, D., & Goel, S. (2020). Racial disparities in automated speech recognition. *Proceedings of the National Academy of Sciences*, *117*(14), 7684–7689.

LDC (1996). *(A) CALLHOME Spanish speech LDC96S35, (B) CALLHOME Spanish transcripts LDC96T17, (C) CALLHOME Spanish Lexicon LDC96L16*. https://catalog.ldc.upenn.edu/

Neumeyer, L., Franco, H., Digalakis, V., & Weintraub, M. (2000). Automatic scoring of pronunciation quality. *Speech Communication*, *30*(2–3), 83–93. https://doi.org/10.1016/S0167-6393(99)00046-1

NICHD (2000). *National reading panel: Teaching children to read*. www.nichd.nih.gov/sites/default/files/publications/pubs/nrp/Documents/report.pdf

Nisbett, R., & Wilson, T. (1977). Telling more than we can know: Verbal reports on mental processes. *Psychological Review*, *84*(3).

Pearson (2009). *Versant Arabic test – test description and validation summary*. Pearson. https://www.pearson.com/content/dam/one-dot-com/one-dot-com/english/SupportingDocs/Versant/ValidationSummary/Versant-Arabic-Test-Description-Validation-Summary.pdf

Pearson Education (2011). *Versant™ Spanish test: Test description and validation summary*. www.versanttest.com/technology/VersantSpanishTestValidation.pdf

Pearson Education (2014). www.pearsonassessments.com/content/dam/school/global/clinical/us/assets/TELL-CERA-Paper-New-Technologies-to-Assess-English-Learners.pdf

Pearson Education (2022). www.pearsonassessments.com/content/dam/school/global/clinical/us/assets/TELL-Overview-Brochure.pdf

Perry, J. (2001). *Reference and reflexivity*. CSLI Publications.

Present-Thomas, R., & Van Moere, A. (2009). *NRS classification consistency of two spoken English tests*. Paper presented at the East Coast Organization of Language Testers 3.4 Conference (ECOLT), WDC.

Reynolds, D. A., Quatieri, T. F., & Dunn, R. B. (2000). Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, *10*(1–3), 19–41.

Rosenfeld, E., Massaro, D., & Bernstein, J. (2003). *Automatic analysis of vocal manifestations of apparent mood or affect* (pp. 5–8). Proceedings of the 3rd international workshop on Models and Analysis of Vocal Emissions for Biomedical Applications, Firenze Press.

Schuller, B. (2018). Speech emotion recognition: Two decades in a nutshell, benchmarks, and ongoing trends. *Communications of the ACM*, *61*(5), 90–99.

Schuller, D., & Schuller, B. (2020, January). A review on five recent and near-future developments in computational processing of emotion in the human voice. *Emotion Review*, *13*(1), 44–50.

Slaney, M., Shriberg, E., & Huang, J.-T. (2013). Pitch-gesture modeling using subband autocorrelation change detection. In *Procedings of Interspeech*.

Swain, M., Routray, A., & Kabisatpathy, P. (2018). Databases, features and classifiers for speech emotion recognition: A review. *International Journal of Speech Technology*, *21*, 93–120. https://doi.org/10.1007/s10772-018-9491-z

White, S., Sabatini, J., Park, B., Chen, J., Bernstein, J., & Li, M. (2020). *The 2018 NAEP oral reading fluency study*. National Center for Education Statistics. https://nces.ed.gov/nationsreportcard/studies/orf

Witt, S. M., & Young, S. J. (2000). Phone-level pronunciation scoring and assessment for interactive language learning. *Speech Communication*, *30*(2–3), 95–108. https://doi.org/10.1016/S0167-6393(99)00044-8

Xiong, W., Wu, L., Alleva, F., Droppo, J., Huang, X., & Stolcke, A. (2018). *The Microsoft 2017 conversational speech recognition system* (pp. 5934–5938). 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada. doi: 10.1109/ICASSP.2018.8461870.

4

Assessment of Clinical Skills

A Case Study in Constructing an NLP-Based Scoring System for Patient Notes

Polina Harik, Janet Mee, Christopher Runyon, and Brian E. Clauser

We begin this chapter with an overview of the evolution of automated scoring of text since Page's early work more than half a century ago (Page, 1966). As that review makes clear, NLP-based scoring systems have made it possible to augment or replace human ratings in scoring written responses (e.g., Burstein et al., 2001; Landauer et al., 2000; Monaghan & Bridgeman, 2005). In many cases this research focuses on the usefulness of these computer-based systems for approximating human scores. Details of the scoring algorithms are often viewed as secondary or, in some cases, they have been intentionally withheld as proprietary intellectual property. Even when details are provided, it is rare that researchers describe how each component – or module – within the system contributes to accuracy. In this chapter, in addition to describing our system, we provide a detailed evaluation focusing on the incremental improvement in accuracy associated with each component of the system. This type of evaluation is commonly referred to as ablation study. Results of this evaluation may have important practical implications for researchers developing similar applications for use in other assessments.

1. Automated Scoring of Written Responses

The history of automated scoring for written responses dates back at least to Page's work in the 1960s (Page, 1966, 1967). A complete historical overview is well beyond the scope of this chapter; instead, we will focus on three trends that have been evident since Page's work was published:

1. Early work on automated scoring of written responses focused on scoring form rather than content. Over time, it has been recognized that in many contexts, it is critical to score content.
2. Typically, the early systems used measures that might be described as surrogates or proxies rather than direct measures of the quality of the written response. Again, in many assessment contexts, there are advantages to replacing these surrogates with more direct measures of the quality of the response.

DOI: 10.4324/9781003278658-5
This chapter has been made available under a CC-BY-NC-ND 4.0 license.

3. These early systems also typically were designed to model or approximate human ratings. It has become apparent that in some contexts, it is advantageous to model what content experts say raters *should* do rather than approximating what they *actually* do (Bennett & Bejar, 1998; Margolis & Clauser, 2020).

From a historical perspective, these changes in approach for automated scoring appear to be trends, but they might better be seen as variations in how automated scoring is implemented. In some contexts, scoring should focus on form, in others on content. In some contexts, surrogates will be both efficient and sufficient; in others, more direct measures will be needed. In some contexts, modeling what human raters do will be appropriate; in others, modeling what raters should do will be preferred. In what follows, we consider each of these trends or variations. Additionally, we place the scoring procedure we developed in the context of other recent state-of-the-art efforts.

1.1 Form vs. Content

Page's work focused on the form of the essay rather than the content (Page, 1966, 1967). A test-taker may well have received an excellent score even if the topic of the scored essay had little discernible relationship to the assigned topic. This was typical of many early efforts. Over time, researchers have tackled the problem of scoring that includes – or is entirely motivated by – the appropriateness of the content.

The historical development of automated scoring systems does not reflect an unequivocal trend away from scoring form to scoring content; in many instances (e.g., K-12 essays), form is central to the proficiency the assessment is intended to measure. There has, however, been a recognition that scoring written responses based solely on the form of the writing may be unacceptably limited. The testing context that we focus on in this chapter is an extreme example: the test-takers are asked to document the important information that they collected in interviewing a patient and completing a physical examination. Scoring is based almost exclusively on the presence (or absence) of critical information (referred to as *key essentials* or *key features*). Provided the writing is understandable, issues of form – such as complete sentences and standard punctuation – are relatively unimportant. In another assessment setting, for example, a middle school student might be asked to write an essay describing the causes of the First World War. That assessment might be scored based on both the presence of relevant and accurate historical information and on the use of appropriate grammar and structure. Finally, in some instances, an essay may be scored (almost) exclusively based on form, but an evaluation of content may be necessary to verify that the essay is 'on topic'.[1]

Two major approaches to scoring content have been developed. The first of these is commonly referred to as latent semantic analysis (Landauer et al., 1998). It provides a general measure of the extent to which the words in the response are similar to those in previously scored responses. The second approach explicitly attempts to match strings of words in the response to concepts delineated in the key.

With latent semantic analysis (and variations on this theme), a large corpus of relevant text is identified. In the case of scoring for a specialized content area – such as medicine – the corpus must be matched to the content area because it is essential that the vocabulary in the responses is included in the corpus. The corpus is then analyzed to produce a multidimensional semantic space in which every word (and document) in the corpus can be represented by a vector of numbers. The corpus will typically include hundreds of thousands of paragraphs of text, and the semantic space will have hundreds of dimensions. A set of previously scored essays can then be located in this semantic space. The associated vectors for new essays can then be compared to those of previously scored essays by using a similarity metric such as the cosine of the angle between the new essay and each previously scored essay. The score for the most similar essay, or the average score for some set of similar essays, can be assigned to the new essay.

Latent semantic analysis has been demonstrated to be useful in a variety of settings. This approach has been incorporated into the topic analysis module in e-rater, the essay scoring system developed by the Educational Testing Service (Burstein et al., 2001). E-rater is designed for use in contexts where content contributes to an overall score but is not the driving focus of the assessment. Swygert et al. (2003) also demonstrated the usefulness of this procedure for scoring written summaries produced by medical students after interacting with standardized patients. Latent semantic analysis has also been incorporated into a system intended to provide writing instruction (Streeter et al., 2011) and has been used in both high-stakes (Shermis, 2014) and low-stakes (Landauer et al., 2000) writing assessments. Latent semantic analysis has generally been less successful in scoring shorter written responses where human raters identify highly specific concepts that can be matched to an answer key (LaVoie et al., 2020; Willis, 2015).

This limitation of latent semantic analysis was one of the motivations for developing alternative procedures intended to match concepts more directly in the key to text in a response. A range of related approaches exists. At the conceptually simple end of this range is an approach described by Yamamoto et al. (2017), which was designed for multilingual assessments. Their procedure is based on exact matches to a key or dictionary and works on the assumption that within a sample of examinee responses, the number of unique responses will be substantially smaller than the total number of responses. With this approach, any time content experts agree on the score to a specific response, that response will be added to the key (or dictionary), and the associated score will be assigned to all other responses that exactly match the scored response. Yamamoto et al. showed that this procedure is suitable for international assessments that include many languages, but relatively small sample sizes per language, and can substantially improve the efficiency of scoring for PISA administrations.

More complex approaches that make greater use of NLP technology include c-rater – also developed by Educational Testing Service (Leacock & Chodorow, 2003; Liu et al., 2014; Sukkarieh & Blackmore, 2009) – and INCITE, the procedure that is the focus of this chapter (Sarker et al., 2019). Again, the approaches used in these systems are designed to identify specific scorable concepts in the response. To implement these approaches, content experts create a key that contains the scorable concepts. Typically, content experts also review and annotate a sample of test-taker responses. The annotation identifies the words or phrases in the response that the annotator believes reflect the scorable concept. Then, using a variety of NLP techniques, the system is constructed to identify the concepts in future responses that may or may not exactly match previously scored responses.

Another example of the general approach of matching text to a key was presented by Willis (2015). With that system, content experts score a sample of responses; NLP technology is then used to develop rules that match the pattern of correct/incorrect judgments produced by the judges. The human judges can then edit the rules. The rules reflect the presence of specific terms in the response and the relative position of those specific terms. The system works iteratively so that human judgments are used to develop scoring rules; the rules can then be used to score additional responses, and humans can be included in the process to score response patterns not seen previously. These new judgments are then used to create additional rules. Numerous variations on the general approach of matching to a key exist (e.g., Cook et al., 2012; Jani et al., 2020).

1.2 Use of Surrogates in Scoring

Longer essays that include longer sentences, more sophisticated vocabulary, and more complex punctuation (e.g., semicolons) are not inherently better essays, but the presence of these characteristics tends to correlate positively with scores. Early automated scoring procedures took advantage of these relationships. The demonstration that it was possible to use an automated system to produce scores that correlated well with human ratings represented an important breakthrough. And if all that is needed is an efficient means of providing an appropriate rank ordering of a set of essays, Page's early approach was useful as well (Page, 1966). There are, however, contexts in which this approach might be viewed as inadequate.

First, for formative assessment of writing, it is important to provide actionable feedback. Scores from this type of scoring procedure are likely to fall short. Telling a student to write longer essays using more sophisticated vocabulary is not likely to be particularly helpful.

Second, for tests used to make high-stakes decisions (graduation, admission, certification), scoring based on surrogates may create an opportunity for test-takers to game the system. If test-takers know that longer responses with more sophisticated vocabulary produce higher scores, they can use that knowledge to artificially inflate their scores. A short essay may be copied and pasted into the response interface multiple times; lists of sophisticated, but irrelevant, vocabulary can be memorized and inserted into the essay (Bejar, 2013; Bridgeman et al., 2012; Higgins & Heilman, 2014).

Third, scoring systems that make substantial use of indirect measures of the quality of an essay lack transparency (sometimes referred to as *traceability*). With systems that use surrogates – or other indirect measures – it is unlikely that stakeholders of the testing process will be able to understand how a specific performance resulted in an associated score.[2] Content experts are likely to be skeptical if they cannot see a relationship between what is taught and what is scored. In the case in which scores are used to make high-stakes decisions, test-takers are likely to believe that knowing how the test is scored is a prerequisite for fair testing.

Transparency was an important consideration in developing the INCITE system. The approach used by the system to match content from the response to a scorable key ensures transparency. The appropriateness of the key can be questioned, and the reliability of the matching process must be empirically evaluated, but the process itself is open to transparent evaluation.

1.3 Modeling Human Scores

Related to the use of correlation as the basis for scoring is the explicit intention to model – or predict – human scores. In some sense human scores are an obvious criterion, given that the automated system is often developed to replace human scoring. At the same time, it must be recognized that human scores are often unreliable. Without constant monitoring and feedback, humans tend to either systematically diverge from identified criteria or to apply criteria inconsistently. Cianciolo et al. (2021) reported on the development of an automated system to score diagnostic justification essays written by medical students. They note, 'Faculty ratings were insufficiently reliable for training machine scoring algorithms, so trained research assistants were employed to re-rate the essays using a more rigorous process' (p. 1027). In this case, the original faculty ratings had inter-rater reliabilities ranging from .13 to .33.

The example provided in the previous paragraph makes it clear that human ratings have practical limitations as a criterion either for modeling automated scores or for evaluating the quality of such scores. Although some researchers (e.g., Cianciolo et al., 2021) have continued to attempt to improve the quality of the ratings, the trend in educational measurement has been to recognize that these ratings may be both practically and theoretically limited. If, for example, content experts agree that an optimal justification for a specific diagnosis would cite four specific patient characteristics (identified through the history and physical examination) and include no additional inaccurate or irrelevant information, the criterion for evaluating the automated scoring system might be based on the accuracy of identifying these scorable features within a set of essays. This approach requires a more detailed description and justification of the scoring criteria and explicitly excludes vaguely defined expert judgment, but it brings the automated scoring process into line with more contemporary principled approaches to test construction such as evidence-centered design (Mislevy et al., 2006).

In the next section we describe the context in which the system was to be deployed operationally. We then provide a description and evaluation of the system.

1.4 Context

Between 2004 and 2020, the USMLE Step 2 Clinical Skills Examination (Step 2 CS) was part of the sequence of assessments required for allopathic medical practice in the United States.[3] This live simulation was administered to approximately 30,000 candidates each year at five locations across the United States. The examination was designed to measure patient-centered clinical skills. For each administration, examinees rotated through a sequence of twelve 25-minute encounters with actors trained to play patients with specific medical problems. Examinees had up to 15 minutes to interact with the standardized patient: taking a focused history, performing a physical examination, and discussing their findings with the patient. During the remaining 10 minutes, the examinee documented the encounter in a *patient note*. These patient notes consisted of two sections: (1) the *data gathering* section required test-takers to document the pertinent findings from the patient history and the physical examination; (2) in the *data interpretation* (DI) section, test-takers were instructed to produce an ordered list of up to three potential diagnoses, provide pertinent evidence from the data gathering section to support each diagnosis, and identify initial diagnostic studies that would be warranted. Examinee patient notes were assigned to physicians trained to rate the notes using case-specific algorithms that mapped patterns of performance onto a rating scale for both the data gathering and data interpretation subcomponents.

In order to pass the examination, it was necessary to receive a passing score on each of three separate components: *the Integrated Clinical Encounter*, *Spoken English Proficiency*, and *Communication and Interpersonal Skills*. The latter two scores were provided by the standardized patients. The *Integrated Clinical Encounter* score consisted primarily of the physician ratings of the data gathering and data interpretation subcomponents.[4]

The motivation for developing an automated scoring system for this examination was much the same as that for other assessments – to reduce the cost of scoring and improve reliability. The specifics of the plan for implementing this system were, however, somewhat different from those for most other large-scale tests. Because the examination was scored pass/fail and no numeric score was reported, minor differences between the scores produced by trained physicians and those produced by the automated system could be ignored for test-takers whose proficiency level was far above the cut score. This allowed for a scoring approach in which all test-takers could be scored by the automated system. Test-takers receiving scores well above the cut score would have a *pass* decision reported based on the computer-generated score. All other test-takers would then be re-scored by physician raters. Preliminary results indicated that this would cut the number of required human ratings by nearly 50% without impacting classification accuracy, which translated into eliminating the need for approximately 200,000 human ratings per year. This would result in substantial savings in time and money.

As we noted, the examination was in place from 2004 to 2020. USMLE Step 2 CS was discontinued during the COVID-19 pandemic because it was unsafe for test-takers to travel to the test sites, and because close interaction between the standardized patients and the test-takers represented a risk to both groups. The rollout of the automated scoring system was scheduled to begin within days of when the examination was terminated, so the system was never used operationally. Nonetheless, the development of the system was completed, and as part of the development process, we evaluated how different components of the system contributed to the usefulness of the system for correctly identifying targeted concepts in the written responses. The remainder of this chapter describes the system and reports on our evaluation of the accuracy of the system for the data gathering section of the patient note.

2. The INCITE System

INCITE is a system designed to identify scorable concepts described as key essentials. The individual concepts can be expressed in many ways depending on the test-taker's choice of words. The number of variations can also grow substantially because identifiable misspellings are also considered correct responses. The system gives priority to high precision over high recall – that is, it gives priority to correctly identifying true matches (minimizing false-positive decisions) over maximizing the number of concepts identified. The system was designed in this way because of the operational context – as noted previously, it was developed for use in high-stakes testing and our intention was to use computer-based scores only for those test-takers with a proficiency level well above the cut score. Test-takers with INCITE-based scores at or below the cut score would be re-scored by human raters. This meant that giving credit for a concept that was absent from the response was a more serious error than failing to give credit for a concept that was present in the response.

The INCITE system attempts to match each key essential to the text of an individual note sequentially until a match is made or until the sequence is completed without a match. The processing sequence includes the following conceptual steps.

2.1 Preprocessing

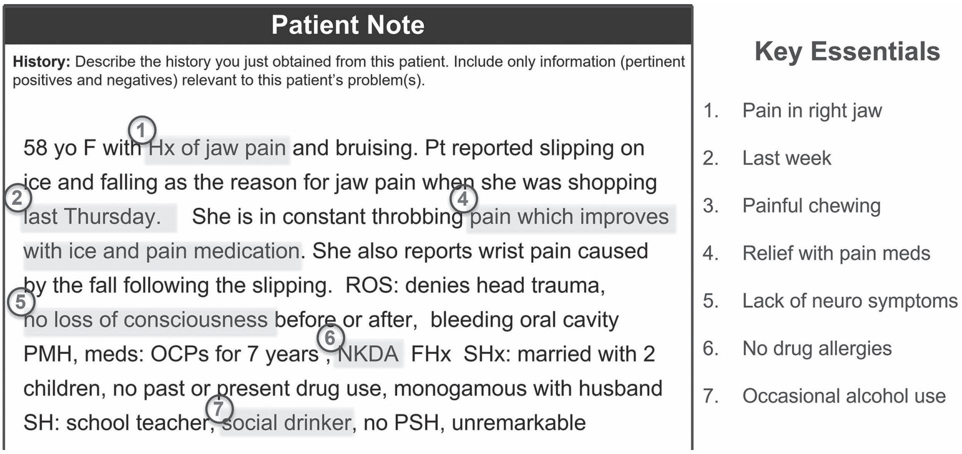
The preprocessing step removes unnecessary characters and converts all text to lowercase. The common steps of stemming and stop-word removal[5] are not performed at this stage because the system relies on exact text matches.

2.2 Annotation

The annotation process was intended to identify specific strings of words in the patient notes that could be mapped to the key essentials. Two annotators annotated each case. They began by annotating three 'training' notes. For each of these notes they discussed what did and did not count as a match. For example, they considered whether 'abdominal pain for some time' would be considered a sufficient representation of 'LLQ abdominal pain x 2 weeks'.[6] Once the common annotation rules were established, the annotators were each given 22 notes; 12 of these were unique to each annotator. Five of the notes annotated in common by both annotators were used to cross-validate the results produced by INCITE and will be discussed in the evaluation section.

The annotations consisted of strings of text found in patient notes that conceptually matched case-specific key essentials. They included a wide range of lexical representations of each key essential: synonyms, misspellings, medical abbreviations, and alternative expressions. The example in Figure 4.1 shows a section of a patient note in which the concepts 'Relief with Pain meds' is documented as 'pain which improves with ice and pain medication' and 'No drug allergies' is recorded as 'NKDA'.

The annotated notes were used to develop the model used for identifying key essentials in the text. As this process proceeded, it became clear that strings of words identified by one



Patient Note

History: Describe the history you just obtained from this patient. Include only information (pertinent positives and negatives) relevant to this patient's problem(s).

58 yo F with Hx of jaw pain and bruising. Pt reported slipping on ice and falling as the reason for jaw pain when she was shopping last Thursday. She is in constant throbbing pain which improves with ice and pain medication. She also reports wrist pain caused by the fall following the slipping. ROS: denies head trauma, no loss of consciousness before or after, bleeding oral cavity PMH, meds: OCPs for 7 years , NKDA FHx SHx: married with 2 children, no past or present drug use, monogamous with husband SH: school teacher, social drinker, no PSH, unremarkable

Key Essentials

1. Pain in right jaw
2. Last week
3. Painful chewing
4. Relief with pain meds
5. Lack of neuro symptoms
6. No drug allergies
7. Occasional alcohol use

Figure 4.1 An example of mapping key essentials concepts to patient note text.

annotator did not consistently agree with the decisions made by the other annotator. To minimize the impact of these inconsistencies, the annotators reconciled all instances in which two annotators matched the same string of words to different key essential concepts. This was typically the result of a clerical error on the part of one of the annotators.

2.3 Exact Matching

The initial matching step includes a search for exact matches to the key essentials as well as matches to variations on the key essentials (different wording and misspellings) represented in several dictionaries. The first dictionary, referred to as the *global dictionary*, was developed without the use of case-specific annotations. The second dictionary – *dictionary-A* – was compiled from the annotations for notes annotated by the two annotators. The third dictionary – *dictionary-B* – included augmentation of the annotations provided by staff involved in fine-tuning the system in addition to the information in dictionary-A. These augmentations were created by combining information from individual annotations. For example, if the annotators had identified the following phrases as representing a specific key essential – 'acetaminophen helps reduce pain' and 'pain is controlled with Tylenol', the augmentation would be: 'Tylenol helps reduce pain' and 'pain is controlled with acetaminophen'.

2.4 Fuzzy Similarity and Dynamic Thresholding

The number of potential variations on each key essential far exceeded the number of variants contained in the dictionaries, so a fuzzy similarity matching module was included in the system. The fuzzy matching used a sliding window to evaluate strings of words. The size of the window depended on the length of the key essential. For each key essential, four windows were used: (1) a window one word shorter than the length of the key essential; (2) a window equal to the length of the key essential; (3) a window one word longer than the key essential; and (4) a window two words longer than the key essential (e.g., a key essential with four words would be evaluated with windows of three to six words).

To evaluate the similarity between the key essentials and the string in the window, we used the Levenshtein Ratio Method. With this approach, the distance between two strings is represented by the number of deletions, insertions, or substitutions that are required to transform one string to the other. The Levenshtein Ratio equals the sum of the length of the two strings minus the distance between the strings, divided by the sum of the length of the strings:

$$Levenshtein\ Ratio = \frac{Length - Distance}{Length}.$$

If the two strings match exactly, the ratio equals one. As the distance between the strings increases, the ratio approaches zero.

Once the Levenshtein Ratio is calculated, a string of words can be classified as a match if the ratio exceeds an identified threshold. Evaluation of the classification accuracy for varying key essentials at different thresholds suggested that a fixed threshold would be suboptimal because the key essentials significantly varied in length. For shorter key essentials, a very high threshold is needed to ensure that, for example, the term ‘contusion’ is not matched to the very different concept ‘concussion’. Decreasing the threshold for shorter key essentials would result in a large number of false positive matches. On the other hand, a high threshold would result in a large number of false negatives for a longer key essential, such as ‘traveled abroad two weeks ago’, where ‘traveled to Kenya a few weeks back’ and ‘international travel 2–3 weeks ago’ are matches that can be detected only with a threshold around 0.6. Experimentation on a pilot set of cases led us to select an approach that uses a dynamic threshold where longer key essential entries have a proportionally lower threshold, compared to shorter entries. The dynamic threshold was defined as

$$DT = T_i - \frac{k \times Length}{100}$$

where T_i is an initially set static threshold, Length is the length of the sliding window, and k is an index that determines the magnitude of the threshold change (for a detailed description, see Sarker et al., 2019).

2.5 Set Overlap and Intersection

Fuzzy matching is effective for identifying many of the variants of the key essentials that are not already included in the dictionaries. However, the approach is ineffective if the text in the note and the key essential use a substantially different word order. Consider the phrase ‘Antibiotics taken in recent times for his symptoms – negative’. This clearly captures the same concept as the key essential ‘Negative for recent antibiotics’, but it would not be identified with the INCITE fuzzy-matching algorithm. To allow for matching under this scenario, a wider search window is employed with a bag-of-words approach.[7] The system searches for strings of words that overlap or intersect with the words in the key essential or alternative versions of the key essential that appear in the dictionaries. To account for misspellings, fuzzy matching with a high threshold is also incorporated in the matching used for the bag-of-words approach.

3. Evaluation of the INCITE System

In what follows, we provide analyses that report how different parts of the system impact the accuracy of identifying key essential concepts in the patient note text. We first examine the usefulness of exact matching of text to the key essentials without and then with the various dictionaries. We then present related results that show the change in the performance of the system when the fuzzy-matching and bag-of-words approaches are incorporated into the system. The primary metric used for reporting these results is the F1 score. This metric is a commonly used index of accuracy in machine learning (Han et al., 2012). It represents the harmonic mean of the precision and recall.

$$\mathrm{F1} = \frac{2}{\frac{1}{Recall} \times \frac{1}{Precision}}.$$

The value can also be expressed as

$$\mathrm{F1} = \frac{True\,Positive}{True\,Positive + .5\left(False\,Positive + False\,Negative\right)}.$$

The F1 score takes on a value of one when the proportion of non-identified (false negatives) and wrongly identified (false positives) concepts are both zero, and a value of zero when no concepts are identified correctly.

3.1 Data

The dataset used for both developing the system and for the subsequent evaluation included patient notes written by examinees who took the Step 2 CS Examination between September 2017 and August 2019. For this study, a curated set of 18 cases representing the range of the exam's content was selected. For each case, a committee of physicians defined the essential concepts (key essentials) expected in a patient note produced by a competent physician. These key essentials reflected information about the patient that should be collected as part of a focused history and physical examination. The number of key essentials per case ranged from 10 to 20. The results presented in the next section are based on the five notes that were annotated by two annotators. These five notes were intended for independent cross-validation and so were not used in developing the system.

3.2 Results

Table 4.1 presents the F1 scores for identifying key essential concepts using exact matching for each of the 18 cases. For the results in this table, a concept was considered present if *either* of the annotators identified it in the note. The second column shows the F1 scores when the matching was implemented using only the key essential definition. The third column shows the same score when the global dictionary is added. The fourth and fifth columns present results associated with adding dictionary-A and dictionary-B, respectively.

Not surprisingly, the F1 scores were relatively low when only the key essential definition was used. Adding each of the three dictionaries improved the scores to a varying degree. Adding the global dictionary increases the mean F1 score by approximately .12. Dictionary-A, which reflects the results of annotation, additionally increases the mean F1 score by .31. Including the augmented annotations in dictionary-B additionally increases the F1 scores by approximately .09. Table 4.2 provides analogous information to that in Table 4.1, but for these results, a key essential was considered present in the note only if it was identified by *both* annotators. The results are similar to those in Table 4.1, reflecting similar incremental improvements as the dictionaries are added (.13, .30, and .08).

Table 4.3 presents F1 scores for the full INCITE system using dictionaries A or B (unlike the results reported in Tables 4.1 and 4.2, these results include exact matches as well as those produced by the fuzzy-matching and bag-of-words procedures). The inclusion of the fuzzy-matching and bag-of-words procedures substantially improve the performance of the system. For dictionary-A, the improvement results in an increase in the mean F1 score of between .19 and .20, depending on whether the criterion was identification of the key essential by annotator 1 or 2 or both annotators 1 and 2. With dictionary-B (which includes the augmented annotations), adding the additional matching procedures increases the mean F1 scores by between

Table 4.1 System Performance (F1 Scores) Against Combined Annotations, Exact Matching Only

	F1 Scores Exact Matching Only			
Case	KEs Only	KEs + Global Dictionaries	KEs, Global Dictionaries, Dictionary-A	KEs, Global Dictionaries, Dictionary-B
1	0.44	0.59	0.75	0.80
2	0.27	0.36	0.71	0.75
3	0.32	0.34	0.68	0.78
4	0.34	0.51	0.65	0.68
5	0.20	0.39	0.74	0.74
6	0.20	0.36	0.69	0.83
7	0.00	0.18	0.55	0.55
8	0.11	0.11	0.48	0.73
9	0.18	0.28	0.53	0.73
10	0.36	0.56	0.72	0.79
11	0.23	0.36	0.77	0.86
12	0.32	0.55	0.68	0.74
13	0.13	0.13	0.67	0.85
14	0.19	0.35	0.80	0.87
15	0.42	0.45	0.82	0.91
16	0.24	0.45	0.68	0.68
17	0.33	0.36	0.68	0.78
18	0.23	0.34	0.70	0.77
Mean	0.25	0.37	0.68	0.77
SD	0.11	0.14	0.09	0.08

Table 4.2 System Performance (F1 Scores) Against Matching Annotations, Exact Matching Only

	F1 Scores Exact Matching Only			
Case	KEs Only	KEs + Global Dictionaries	KEs, Global Dictionaries, Dictionary-A	KEs, Global Dictionaries, Dictionary-B
1	0.44	0.59	0.75	0.80
2	0.27	0.37	0.73	0.77
3	0.32	0.35	0.68	0.78
4	0.36	0.53	0.68	0.71
5	0.21	0.41	0.75	0.75
6	0.21	0.38	0.71	0.81
7	0.00	0.19	0.57	0.57
8	0.12	0.12	0.52	0.75
9	0.20	0.31	0.52	0.70
10	0.39	0.61	0.76	0.79
11	0.25	0.35	0.79	0.84
12	0.33	0.57	0.67	0.71
13	0.13	0.13	0.67	0.84
14	0.20	0.37	0.79	0.84

(*Continued*)

Table 4.2 System Performance (F1 Scores) Against Matching Annotations, Exact Matching Only *(Continued)*

	F1 Scores Exact Matching Only			
Case	KEs Only	KEs + Global Dictionaries	KEs, Global Dictionaries, Dictionary-A	KEs, Global Dictionaries, Dictionary-B
15	0.44	0.47	0.82	0.87
16	0.25	0.47	0.71	0.71
17	0.36	0.39	0.68	0.78
18	0.25	0.36	0.69	0.76
Mean	0.26	0.39	0.69	0.77
SD	0.12	0.14	0.09	0.07

Table 4.3 System Performance (F1 Scores) for Exact, Fuzzy and Bag-of-Words Matching

	F1-Score			
Case	A1OrA2 and INCITE-A	A1AndA2 and INCITE-A	A1OrA2 and INCITE-B	A1AndA2 and INCITE-B
1	0.89	0.89	0.91	0.91
2	0.86	0.85	0.86	0.85
3	0.87	0.88	0.90	0.91
4	0.94	0.97	0.94	0.97
5	0.88	0.88	0.88	0.88
6	0.88	0.91	0.95	0.94
7	0.89	0.87	0.89	0.87
8	0.89	0.88	0.96	0.93
9	0.86	0.85	0.91	0.87
10	0.85	0.88	0.91	0.90
11	0.88	0.86	0.94	0.89
12	0.86	0.86	0.91	0.89
13	0.83	0.82	0.96	0.93
14	0.93	0.93	0.95	0.93
15	0.89	0.89	0.95	0.92
16	0.84	0.85	0.87	0.87
17	0.86	0.83	0.91	0.88
18	0.90	0.88	0.93	0.91
Mean	0.88	0.88	0.92	0.90
SD	0.03	0.04	0.03	0.03

Table 4.4 Counts of Classifications for Cross-Validation Samples Using INCITE

	Classification			
Case	False Negative	False Positive	Ture Negative	True Positive
1	9	1	29	51
2	11	6	31	52
3	7	2	26	40

Case	Classification False Negative	False Positive	Ture Negative	True Positive
4	3	1	15	31
5	11	1	31	42
6	5	1	21	58
7	6	5	19	45
8	4	0	20	46
9	7	1	15	42
10	8	2	35	50
11	5	1	14	50
12	6	2	26	41
13	2	3	17	58
14	4	2	31	53
15	2	3	20	50
16	7	7	26	45
17	6	2	34	43
18	4	3	28	50
Mean	5.9	2.4	24.3	47.1

.13 and .15. Again, the use of dictionary-B provides modestly better overall performance than dictionary-A.

To provide a more detailed evaluation of the classification accuracy for the system, Table 4.4 presents counts of false-negative, false-positive, true-negative, and true-positive classifications for the cross-validation sample for each case using the full INCITE system with classifications made by annotators *A and B* as the criterion. Consistent with our intentions in designing the system, the number of true-positive and true-negative classifications is high, and when classification errors occur, false-negative error rates (failing to identify a concept) were substantially higher than false-positive error rates (giving credit for a concept that was not present). The ratio of these errors is in excess of two to one.

4. Discussion

Numerous papers cited in this chapter have reported results showing that automated systems are capable of accurately scoring text. Depending on the context, these systems have been used in conjunction with human raters or independently. As we noted, previous (unpublished) results indicated that the INCITE system could reduce the number of human ratings required by half with no change in classification accuracy. Such information is important because the primary reason for introducing computerized scoring is to improve efficiency. Although these results provide encouragement about the usefulness of automated scoring of text-based responses, they provide little guidance for researchers hoping to develop new scoring systems. Often there is little detail about the specifics of the system; it is even rarer that information is provided about how different components of a system improve the accuracy of the scoring. The results presented in this chapter represent a step towards filling that gap.

Results of the type presented in this chapter can fill a number of needs. First, they provide information about the relative contribution of different scoring components. Introducing each component will have a cost. That cost might be in: (1) the human effort required to develop the module (e.g., programing time); (2) the human effort required to implement the component (e.g., annotation time); or (3) the computer time required to implement the module for each response that must be scored. The results reported in this chapter reflect primarily on the second of these costs, although the third is important as well.

When we consider the human effort required to implement the INCITE system, there are two separate aspects of that effort to consider. The first of these is, to what extent is the system improved by customizing the scoring for each case? The primary effort required for this customization is the work done by the annotators. Setting aside the notes required for the cross-validation, customizing the system for each case required each of two annotators to annotate 20 notes. This is not a trivial amount of work, but it represents hours – not days – of effort for each annotator in each case. The return on this investment is represented by the increase in the F1 scores presented in the third and fourth columns of Tables 4.1 and 4.2. On average, that increase is approximately 0.30, which is substantial.

A second, separable effort involved in preparing the case-specific algorithms is represented by the augmentation step. The augmentation process requires careful review and comparison of the terms produced as part of annotation. Again, the effort per case is represented by hours of work, not days. The payoff of this effort is shown by comparing the fourth and fifth columns of Tables 4.1 and 4.2 or by comparing the INCITE-A and INCITE-B columns in Table 4.3. This improvement is meaningful, but more modest than that associated with the original annotation.

The third largely separable component of the scoring system is represented by the fuzzy-matching and bag-of-words modules. Again, these procedures substantially improve the matching accuracy: mean increases of .19 to .20 were observed when they were applied to dictionary-A and .13 to .15 when they were applied to dictionary-B. These results suggest that some of the benefit associated with the augmentation process could be achieved simply by introducing the fuzzy-matching and bag-of-words modules, without augmentation. Nonetheless, the full system including augmentation continues to outperform the system without augmentation.

Adding these NLP-based matching procedures (fuzzy matching and bag of words) clearly enhances the system. It provides this enhancement without additional human review and intervention. That said, it is certainly not without cost. In addition to the programming time, experimentation was necessary to identify optimal search windows and thresholds for both the fuzzy-matching and bag-of-words modules. Introducing these procedures also makes the system more computationally intensive.[8]

The results make it clear that each component of the system adds to the accuracy of the identification of key essentials. These same results also suggest that the benefits are not strictly additive.[9] We have already commented that a proportion of the incremental matches resulting from the augmented annotations would have been produced by instituting the fuzzy-matching and bag-of-words procedures without including the augmented annotations. In this context, it is worth examining results for individual cases. As represented in Tables 4.1 and 4.2, case 7 stands out. For this case, exact matching based on the key essentials is essentially worthless. Using the variants represented in the dictionaries similarly results in the lowest F1 scores for any of the 18 cases. However, after including the fuzzy-matching and bag-of-words procedures, the case is no longer an outlier. Cases 1 and 15 represent the opposite pattern. These cases have the highest F1 values for exact matches both without and with the various dictionaries; the F1 scores for these cases are high after including the fuzzy-matching and bag-of-words procedures in the processing, but they are no longer the highest scores. This general pattern is confirmed by examining the standard deviations (across cases) for the scores reported in Tables 4.1 through 4.3. The standard deviations for the scores in Tables 4.1 and 4.2, reflecting variability in F1 scores for exact matching, range from 0.07 to 0.14. This indicates a moderate level of variability across cases. In Table 4.3, which reports F1 scores for the full INCITE system including the fuzzy-matching and bag-of-words procedures, the variability across cases is reduced to between 0.03 and 0.04. This suggests that these more computationally intensive procedures are, relatively speaking, more useful when the exact-matching procedures are less useful.

The results reported in this chapter reflect a reasonably high level of accuracy for the INCITE system, but the scores are not perfect. As we have already mentioned, although the INCITE system was targeted for operational use at the time the Step 2 CS examination was discontinued, the system has continued to evolve. We are currently making two enhancements to the system that we expect will result in incremental improvements in performance. The first of these will allow us to introduce unique/customized thresholds for applying the Levenshtein Ratio for each key essential. The current form of INCITE uses thresholds that are a function of the length of the key essential. This approach proved to work better than using a single fixed threshold, but it ignores the fact that some key essentials are inherently less likely to produce false-positive matches and so can be associated with a lower threshold – presumably leading to more true-positive matches. The second enhancement to the system will record the specific position in the text where the match was made. This will allow for evaluation of the specific text that resulted in each false-positive match. This type of evaluation will both support the identification of an optimal threshold for individual key essentials and will provide a basis for identifying other aspects of the system that could be modified.

With regard to identifying optimal thresholds, it is worth returning to the results reported in Table 4.4. As we noted, in the context of the intended application, priority was given to precision over recall, and the results in the table reflect this choice; the false-negative rates are more than double the false-positive rates. This suggests that it might be possible to increase the overall accuracy – as reflected in the F1 scores – by using a lower threshold.

One final issue is worth mentioning in interpreting the results presented in this chapter. Although the decisions made by the annotators have been treated as truth, those decisions are not error free. The mean F1 scores reported for the full INCITE system in Table 4.3 (labeled INCITE-B) varies from .90 to .92, depending on whether the criterion is defined by concepts identified independently by both annotators *or* by concepts identified by at least one annotator. This difference reflects the less-than-perfect agreement between the annotators. This is admittedly a small difference, but it is, nonetheless, a meaningful consideration as we attempt to improve the accuracy of the system beyond the current level.

5. Conclusion

In this chapter we have described the INCITE system, an NLP-based system for computerized scoring of patient notes. The emphasis has been on the specifics of the system and how each component contributes to overall accuracy. In interpreting these results or in adopting aspects of the system for use in another context, it is important to remember the specifics of the context in which the system was developed. First, because it is used to score patient notes, it uses a specialized vocabulary. Scoring responses that use a different vocabulary may be more or less challenging, depending on the specifics. A second consideration is that we constructed the system to support transparency. This resulted in excluding some widely used approaches to evaluating text. Additionally, our focus was limited to scoring content. This decision will certainly impact the applicability of a system like the one we described for use in other contexts. Finally, we decided to prioritize precision over recall. This decision may have impacted the overall accuracy of the system and may be inappropriate in some other settings.

Notes

1 There are a number of reasons a test-taker might choose to diverge from an assigned topic. At the extreme, this might include memorizing a well-written essay that the individual test-taker would have been unable to write. Identifying this sort of effort to game the system may require a fairly minimal evaluation of the content, but that evaluation could be critical for appropriate scoring.

2 Although systems that use surrogates are likely to lack transparency, other methods that use indirect measures of the quality of the response, such as latent semantic analysis, also have this limitation.

3 Practice for physicians with an MD degree.

4 These ratings were then combined with scores from the standardized patient that indicated whether the examinee correctly completed important components of the physical examination, referred to as the *physical exam* score.

5 Stemming and stop-word removal are common preprocessing steps for NLP systems. Stemming is a process in which words are reduced to their root or stem by eliminating suffixes. Stop words are common words in English (e.g., articles, prepositions, pronouns, conjunctions). They are typically removed because they tend to carry relatively little information that can be used in NLP.

6 This, in fact, would not be considered a match.

7 Bag of words refers to matching based on the number of words in one sample that are also found in a second sample, without regard to word order.

8 One aspect of the INCITE system was not included in our evaluation, but nonetheless warrants comment. In our description, we noted in passing that the process is sequential. The efficiency of the system – in terms of the time required for processing a note – was maximized by sequentially moving to more and more computationally intensive steps. Each note must be searched for each key essential associated with the case. The sequence for each search begins with exact matching to the key essential, followed by exact matching to the various dictionaries. This is followed by the more computationally intensive fuzzy-matching procedure and the bag-of-words procedure. Whenever a match occurs, the search is terminated. As Tables 4.1 and 4.2 suggest, although exact matching is in itself insufficient, these less computationally intensive procedures identify a substantial proportion of the variants.

9 We also note that the F1 scores do not represent an additive (or equal interval) scale. It is reasonable to interpret higher F1 scores as representing more accurate matching than lower F1 scores. It is not appropriate to interpret an increase from .50 to .55 as being equivalent to a change from .95 to 1.00.

References

Bejar, I. I. (2013, April). *Gaming a scoring engine: Lexical and discourse-level construct irrelevant response strategies in the assessment of writing*. Paper presented at the annual meeting of the National Council on Measurement in Education.

Bennett, R. E., & Bejar, I. I. (1998). Validity and automated scoring: It's not only the scoring. *Educational Measurement: Issues and Practice*, *17*(4), 9–16.

Bridgeman, B., Trapani, C., & Attali, Y. (2012). Comparison of human and machine scoring of essays: Differences by gender, ethnicity, and country. *Applied Measurement in Education*, *25*, 27–40.

Burstein, J., & Leacock, C., & Swartz, R. (2001). *Automated evaluation of essays and short answers*. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=4d7766c71bc1d03dc5b53511cac4ca947b017034

Cianciolo, A. T., LaVoie, N., & Parker, J. (2021). Machine scoring of medical students' written clinical reasoning: Initial validity evidence. *Academic Medicine*, *96*, 1026–1035.

Cook, R., Baldwin, S., & Clauser, B. (2012, April). *An NLP-based approach to automated scoring of the USMLE® step 2 CS patient note*. Paper presented at the annual meeting of the National Council on Measurement in Education.

Han, J. W., Kamber, M., & Pei, J. (2012). *Data mining concepts and techniques* (3rd ed.). Morgan Kaufmann Publishers.

Higgins, D., & Heilman, M. (2014). Managing what we can measure: Quantifying the susceptibility of automated scoring systems to gaming behavior. *Educational Measurement: Issues and Practice*, *33*(3), 36–46.

Jani, K. H., Jones, K. A., Jones, G. W., Amiel, J. B., & Elhadad, N. (2020). Machine learning to extract communication and history-taking skills in OSCE transcripts. *Medical Education*, 1–12.

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Process*, *25*, 259–284.

Landauer, T. K., Laham, D., & Foltz, P. W. (2000). The intelligent essay assessor. *IEEE Intelligent Systems*, *15*(5), 27–31.

LaVoie, N., Parker, J., Legree, P. J., Ardison, S., & Kilcullen, R. N. (2020). Using latent semantic analysis to score short answer constructed responses: Automated scoring of the consequences test. *Educational and Psychological Measurement*, *80*, 399–414.

Leacock, C., & Chodorow, M. (2003). C-rater: Automated scoring of short-answer questions. *Computers and the Humanities*, *37*(4).

Liu, O. L., Brew, C., Blackmore, J., Gerard, L., Madhok, J., & Linn, M. C. (2014). Automated scoring of constructed-response science items: Prospects and obstacles. *Educational Measurement: Issues and Practice*, *33*, 19–28.

Margolis, M. J., & Clauser, B. E. (2020). Automated scoring in medical licensing. In D. Yan, A. A. Rupp, & P. Foltz (Eds.), *Handbook of automated scoring: Theory into practice* (pp. 445–467). Taylor and Francis.

Mislevy, R. J., Steinberg, L. S., Almond, R. G., & Lucas, J. F. (2006). Concepts, terminology, and basic models of evidence centered design. In D. M. Williamson, R. J. Mislevy, & I. I. Bejar (Eds.), *Automated scoring for complex tasks in computer-based testing* (pp. 123–167). Lawrence Erlbaum Associates.

Monaghan, W., & Bridgeman, B. (2005, April). *E-Rater as a quality control on human scores*. ETS R&D Connections, ETS.

Page, E. B. (1966). *Grading essays by computer: Progress report*. Notes from the 1966 Invitational Conference on Testing Problems, Educational Testing Service.

Page, E. B. (1967). The imminence of grading essays by computer. *Phi Delta Kappan*, *47*, 238–243.

Sarker, A. D., Klein, A. Z., Mee, J., Harik, P., & Gonzalez-Hernandez, G. (2019). An interpretable natural language processing system for written medical examination assessment. *Journal of Biomedical Informatics*, *98*.

Shermis, M. D. (2014). State-of-the-art automated essay scoring: Competition, results, and future directions from a United States demonstration. *Assessing Writing*, *20*, 53–76.

Streeter, L., Bernstein, J., Foltz, P., & DeLand, D. (2011). *Pearson's automated scoring of writing, speaking, and mathematics* (White Paper). http://kt.pearsonassessments.com/download/PearsonAutomatedScoring-WritingSpeaking-Math-051911.pdf

Sukkarieh, J. Z., & Blackmore, J. (2009). *C-rater: Automatic content scoring for short constructed responses* (pp. 290–295). Proceedings of the Twenty-Second International FLAIRS Conference, Association for the Advancement of Artificial Intelligence.

Swygert, K., Margolis, M., King, A., Siftar, T., Clyman, S., Hawkins, R., & Clauser, B. (2003). Evaluation of an automated procedure for scoring patient notes as part of a clinical skills examination. *Academic Medicine (RIME Supplement)*, *78*(10), S75–S77.

Willis, A. (2015). *Using NLP to support scalable assessment of short free text responses* (pp. 243–253). Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications, Association for Computational Linguistics.

Yamamoto, K., He, Q., Shin, H. J., & von Davier, M. (2017). *Developing a machine-supported coding system for constructed-response items in PISA (ETS RR-17-47)*. Educational Testing Service.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Part II

Item Development

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

5

Automatic Generation of Multiple-Choice Test Items from Paragraphs Using Deep Neural Networks

Ruslan Mitkov, Le An Ha, Halyna Maslak, Tharindu Ranasinghe, and Vilelmini Sosoni

1. Introduction

Multiple-choice tests are a popular form of objective assessment where respondents are asked to select only one answer from a list of choices and are extensively used in teaching, learning, and training but also market research, elections, and TV shows. Against the background of ubiquitous digitalization, there is a pressing need to find a more efficient way to automate the development and delivery of multiple-choice test items. With the manual construction of such tests being a time-consuming and labor-intensive task, the objective of this chapter is to find effective alternatives to the lengthy and demanding activity of constructing multiple-choice test items by employing state-of-the-art natural language processing (NLP) and deep learning (DL) techniques.

Mitkov and Ha (2003) pioneered the development of an NLP methodology for generating multiple-choice tests from educational textbooks. Their original methodology employed various NLP techniques including shallow parsing, automatic term extraction, sentence transformation and semantic distance computing, and language resources such as corpora and ontologies. The automatic construction of multiple-choice tests included the identification of important concepts in the text, the generation of questions about these concepts, as well as the selection of semantically close-to-the-correct-answer distractors. The tool developed on the basis of this methodology offered the option to post-edit the automatically produced test items in a user-friendly interface. The evaluation reported that in assisting test developers to construct test items in a significantly faster and expedient manner without compromising quality, the implemented tool saved both time and production costs.

While Mitkov and Ha's study has been well received by the field, it also has limitations, one of them being that each multiple-choice test item is generated from a single sentence. Our next challenge was to develop a methodology and tool capable of generating the questions of multiple-choice test items from the information contained in a paragraph of several sentences, not just a single sentence. In this chapter, we report on our experiments with the latest DL techniques seeking to achieve this objective. The generation of multiple-choice questions (MCQs) from more than one sentence is a significant breakthrough given that all previous related work had attempted generation from one sentence only.[1]

DOI: 10.4324/9781003278658-7
This chapter has been made available under a CC-BY-NC-ND 4.0 license.

The rest of the chapter is structured as follows. The next section outlines the related work on this topic. The section 'Data Preparation' discusses the data compiled and employed in this study. The section 'Methodology' details the methodology adopted, and the section 'Performance Evaluation' presents the evaluation results. The chapter finishes with a summary of the envisaged future research.

2. Related Work

Mitkov and Ha (2003) pioneered the generation of MCQs automatically. A few years later, they reported the generation of multiple-choice test items from medical text using rapid item generation (RIG) and the UMLS thesaurus (Karamanis et al., 2006). A medical textbook served as the source texts, while a much more extensive collection of MEDLINE texts was used as the reference corpus (RC). In another study, Mitkov et al. (2006) employed various NLP techniques including automatic term extraction, shallow parsing, sentence transformation, and computing of semantic distance as well as corpora and ontologies. In contrast to the aforementioned studies which benefited from the data of a specific domain, Papasalouros et al. (2008) described a domain-independent approach employing specific ontology-based strategies and Web Ontology Language (OWL).

Several years later, Singh Bhatia et al. (2013) proposed a methodology which selects sentences by retrieving existing test items on the Web as well as a technique for creating named entity distractors from Wikipedia, while Alsubait et al. (2014) used OWL ontologies to generate multiple-choice test items and proposed a psychologically based theory to control the question difficulty.

Afzal and Mitkov's (2014) system for generation of multiple-choice tests employed an unsupervised dependency-based approach to identify the most important named entities and terms and define semantic relations between them. Their approach did not use any prior knowledge about the semantic types of the relations but was based on a dependency tree model. The results were evaluated in respect of their readability, usefulness of semantic relations, relevance, acceptability of questions and distractors, and general usability of multiple-choice test items.

More recently, the focus of the research community shifted towards the use of neural networks for NLP tasks and applications, including the generation of multiple-choice tests. Liang et al. (2018) investigated how machine-learning models – in particular feature-based and neural net (NN)–based ranking models – can be used for distractor selection. Gao et al. (2019) proposed a hierarchical encoder-decoder framework to generate question items for reading comprehension questions from real examinations. Susanti et al. (2018) investigated methods for automatically generating distractors for MCQs on English vocabulary by employing semantic similarity and collocation information, and Shin et al. (2019) used a topic modeling procedure, machine learning, and NLP to generate distractors based on students' misconceptions.

All of the aforementioned approaches generate MCQs (questions or distractors) automatically. Furthermore, all questions are generated on the basis of a single sentence only. To the best of our knowledge, the study we report in this chapter is the first instance of such questions being produced from paragraphs rather than single sentences.

It is worth noting that there has recently been an increasing amount of work on question generation as part of the Question Answering (QA) NLP application (Lee et al., 2020; Qi et al., 2020; Xiao et al., 2020). In particular, question generation has followed the recent trend in DL models for NLP to use generic text-to-text models for a variety of tasks. The models are first pretrained on large amounts of texts using creative unsupervised objectives (such as predicting the masked segments, or predicting the order of the texts). It is hoped that by doing this, the models will learn the knowledge (or at least uncover the correlations) needed to solve downstream tasks. Then, the models are fine-tuned for the specific task at hand, by training them on the task-specific objectives such as generating questions from a pair of text-answer. There is evidence that handcrafted knowledge can be substituted by large-scale models. Models which perform well in the task of generating questions include ENRIE-GEN, ProphetNet, and Info-HCVAE. All of these models are text-to-text generation models whose differences are mainly in the architectures and the pretraining objectives.

However, there are several drawbacks when using these DL question generation models. The first one is that they are essentially black boxes: we do not know how or why they generate the questions and, as a result, cannot quickly fix some issues when raised. The second one is that at the moment, they mostly generate questions from single sentences rather than several sentences.

3. Data Preparation

The method for finding similar paragraphs was inspired by our recent research on retrieving fuzzy matches in a translation memory (TM) system (Ranasinghe et al., 2020). The idea was to compile and operationalize a multiple-choice test items corpus (MCTIC) as ‘source text’ and at the same time to build and benefit from a larger reference corpus (RC) which could play the role of the TM. To the best of our knowledge, no similar corpus exists or is publicly available which would be suitable for this study. Further, all MCTIC test items are constructed to cover multiple sentences. In other words, questions cannot be successfully answered based on the information in one sentence only.

For this study we chose the European Union (EU) law domain, the rationale being that MCQs are used to assess knowledge of job seekers applying for positions at EU institutions. Furthermore, MCQs are frequently used during exams at law schools and universities.

3.1 The MCTIC Corpus

More specifically, for the purpose of this study, we compiled a corpus of MCQs based on multiple sentences within the EU law domain. The corpus consists of the following triples: (1) the paragraph from a chosen book on which the question is based; (2) the question associated with the paragraph; and (3) the answer. Before compiling this corpus, we considered using other available resources, including QA datasets such as the Natural Questions corpus (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), HotpotQA (Yang et al., 2018), QuAC (Choi et al., 2018), QASPER (Dasigi et al., 2021), CaseHOLD (Zheng et al., 2021), OpenBookQA (Mihaylov et al., 2018), SQuAD (Rajpurkar et al., 2016), and MultiRC (Khashabi et al., 2018). However, as most of them contain QA pairs derived from Wikipedia articles, and questions were not normally based on multiple sentences, they were not deemed suitable for this study which focuses on textbooks and other teaching materials for real-life classroom scenarios.

We based MCTIC on an electronic textbook (Turner & Storey, 2014) because of the following considerations: first, the contents of the book covered general information about the EU and did not include any specific details regarding specialized legislation cases (for instance, home affairs law, EU immigration and asylum law, commerce, or terrorism are too narrow topics and were not envisaged to be part of the corpus); second, the information presented in the book was mostly written in full paragraphs, did not feature many bullet points, and was not stored in tables which in itself facilitated the processing.

The selected textbook consists of 16 chapters and covers a number of topics from the history and legislation of the EU, including but not limited to the origins and character of EU law, the development from Community to Union; the political and legal EU, the sources of EU law, the legislative process, enforcement of EU law, EU competition law, and the relationship between the EU and member states’ national laws.

The following criteria were applied to compile the corpus.

1. *Paragraphs had to be included for all questions*. Since the experiments were based on paragraph similarity matching, questions without the associated paragraph could not be used. In this study, a paragraph refers to either an actual paragraph from the book or a couple of sentences on which the question is based. The 'paragraph cell' contains only those sentences which are necessary to answer the question. Therefore, on several occasions the actual paragraph from the book had to be edited to include only the sentences needed to answer the question. The length of paragraphs varied from two to five sentences, and one sentence only does not constitute a paragraph. The number of sentences in the paragraph depended on the question to be asked and its answer. Although it was possible to include longer paragraphs in the MCQ corpus, it would have been challenging to produce a question that is based on the information from too many sentences.
2. *All the questions in MCTIC had to be based on the whole paragraph*. That is, it should be impossible to answer the question without reading the whole paragraph. In this sense, yes/no or true/false questions are not possible. The questions should be worded as actual questions rather than incomplete statements.
3. *The answer should consist of one or a few words rather than a long phrase*. The answer cannot consist of a whole sentence given the way distractors are automatically selected by Mitkov and Ha's (2003) system. Although the distractors were not generated for this study, the idea would be implemented in future research to combine this methodology for the generation of multiple-choice tests including distractors.

Following the three selection criteria, the MCTIC corpus was compiled consisting of a total of 200 paragraphs and QA pairs. The corpus was checked with the writing tool Grammarly to spot grammar and spelling mistakes, typos, extra spaces, and other issues. Furthermore, all the questions were revised by a native speaker of English who is a professional proofreader and qualified linguist. The limited size of the corpus due to lack of sufficient resources should be noted as a constraint of this study. Another constraint for the same reason is that the MCQ corpus was not revised by an expert in EU law. The construction of a larger corpus and its validation by an EU law expert is planned as a follow-up study.

3.2 The Reference Corpus

An RC was compiled and based on a textbook (Kaczorowska-Ireland, 2016) covering similar topics as in the MCTIC corpus. The textbook, originally a PDF file, was processed using Python scripts to delete all irrelevant information such as headings, footnotes, images, and tables, and to join hyphenated words at the end of a line. The resulting RC consisted only of extracted paragraphs within the chosen domain. The approach proposed in this chapter is inspired by the functionality of computer-assisted translation (CAT) tools in terms of TM[2] matches. The aforementioned DL models are used to identify similar paragraphs in RC based on the paragraph similarity score match against that of the MCTIC paragraphs. More specifically, the proposed methodology operates as follows: MCTIC features pairs of paragraphs and questions $<P_I,Q_I>$s and for every paragraph P_R from the RC, matches with all paragraphs in the MCTIC will be returned. If P_R ($P_R \in RC$) matches paragraph P_M with $P_M \in MCTIC$, the pair $<P_M,Q_M>$ will be retrieved, and the question Q_M will serve as the template to be post-edited.

The aforementioned methodology resembles the way a TM match is suggested in TM tools for the target language segment. It is assumed that the proposed question will need only minor revision by human editors, since the new paragraph is expected to offer a high fuzzy match score.[3]

For the compilation of the RC, we opted for a textbook on European law. The idea was to identify a book which covers the same topics in the EU law and is available in a machine-readable or easy-to-process format. It was important for the information in the book to be presented in text (paragraphs) rather than in diagrams, charts, tables, or images. In contrast to the MCTIC, the RC corpus had to be of a considerably larger size. While searching for such a suitable book, we came across several challenges. The majority of books on EU law were available only as hard copies. In some cases, if available in PDF format, the printed book was scanned. Therefore, even with the use of the optical character recognition (OCR) software, the processed output would be of very low quality and would require a significant amount of time and human resources to be revised manually.

The book we chose consisted of 1,198 pages and included similar chapters as the one used for the MCQ corpus. The PDF was processed with a Python program in order to write the text into a text file. The information unrelated to the contents of the relevant paragraphs was omitted. For example, the table of contents, page numbers, headings, headers, footnotes, and the list of references, as well as any tables or images, were not included in the RC.

Moreover, in the original file, the words at the end of the line were hyphenated. It is a typographical hyphen that had to be removed from the corpus so that same words with or without hyphens were not considered as different ones. Besides, a new line character followed this typographical hyphen, so the word was split into two parts. After the new line symbol following the hyphen was deleted, the obtained words were checked by the Enchant module (spell-checking library for Python). If such a word was found in the library, it was left intact. Otherwise, the hyphen was removed.

The other issues with processing the file included the fact that many chapters included quotes that constitute a large paragraph. They should not be included in the RC corpus since the MCTIC corpus can contain the same quotations. As already pointed out, for the purposes of this research, we seek to retrieve fuzzy matches instead of the exact matches. Therefore, it would be counterproductive to have exactly the same paragraphs in both corpora. Since such quotes were written in a different font and size, it was possible for the Python program to detect and remove them.

MCTIC and RC were compared in terms of the word count, the number of unique words, the average number of words per paragraph and the most common words. As previously stated, the MCTIC includes 200 multiple-choice test items together with the referencing paragraphs totalling 12,754 words. A word in this context is a sequence of alphanumeric characters separated by a space. On the other hand, the RC amounts to 399,196 words. It encompasses more than a thousand pages of the processed text. Thus, it is more than 31 times larger than the multiple-choice questions corpus.

The average number of words in a paragraph in the MCTIC corpus is 63 words; for RC, this number is 57.

4. Methodology

As stated, our study was inspired by Ranasinghe et al. (2021), who used sentence encoders to improve the matching and retrieving process in TM systems. This study uses a similar approach to find the matching paragraphs. The paragraphs from the MCTIC corpus would correspond to the ‘source language sentence’ in a TM system, whereas the multiple-choice question from that paragraph would correspond to the ‘target language sentence’. Paragraphs in the RC would be equivalent to potential incoming segments in a TM. Our method should retrieve the best match to the incoming segment from the TM.

In line with Ranasinghe et al. (2021), we follow these steps.

1. The embeddings[4] for the paragraphs in MCTIC were generated using a sentence encoder.[5] The generated sentence embeddings were stored in the random-access memory of the computer in order to enable fast access to them.

2. For a paragraph in the RC, the embedding was acquired using the same sentence encoder as the first step. We call this the input paragraph.
3. The cosine similarity between the embedding of the input paragraph and the embeddings of the paragraphs from the MCTIC was computed.
4. The embedding with the highest similarity score from the MCTIC was returned as the best match for that input paragraph.
5. These four steps were repeated for all the paragraphs in the RC.

As the sentence encoders, we used two recently released NLP algorithms; doc2vec (Le & Mikolov, 2014) and SBERT (Reimers & Gurevych, 2019). Both employ neural architectures and have shown promising results in text-similarity tasks.

Doc2vec, proposed by Le and Mikolov (2014), is an extension to Word2Vec (Mikolov et al., 2013) to learn document-level embeddings. Instead of using just words to predict the next word, Le and Mikolov (2014) added another feature vector, which is document-unique. The doc2vec models can be used in the following way. In the training phase, a set of documents is required. A word vector is generated for each word, and a document vector is generated for each document. In the inference stage, a new document is presented, and all weights are fixed to calculate the document vector. We used this document vector to represent the paragraphs in the RC corpus and MCQ corpus. To the best of our knowledge, there are no pretrained doc2vec models available. Therefore, we trained a doc2vec model using the RC corpus and MCQ corpus.

The second sentence encoder we employed in this study is SBERT (Reimers & Gurevych, 2019). SBERT is a modification of the pretrained BERT (Devlin et al., 2019) network that uses Siamese and triplet network structures (Reimers & Gurevych, 2019) to derive semantically meaningful sentence embeddings that can be compared using cosine similarity. Even though the BERT model has achieved state-of-the-art performance in textual similarity tasks, it requires both texts to be fed into the network, which causes a massive computational overhead. Finding the most similar pair in a collection of 10,000 texts requires about 50 million inference computations which will take approximately 65 hours with BERT (Reimers & Gurevych, 2019). SBERT reduces the effort for finding the most similar pair from 65 hours to about 5 seconds. Since efficiency is essential in our application, we used SBERT instead of BERT. Unlike doc2vec, SBERT released several pretrained models.[6] We employed the 'all-MiniLM-L6-v2' model from the available pretrained models. It is the fastest model with reasonable accuracy on the Semantic Textual Similarity (STS) benchmark,[7] producing an 85.29 Spearman correlation coefficient. Furthermore, 'all-MiniLM-L6-v2' is small in size and does not require ample disk space compared to other models.[8]

5. Performance Evaluation

We evaluated the matching performance of both doc2vec and SBERT. The evaluation results show how successful each of these DL models is at identifying semantically similar paragraphs in the RC given a specific paragraph from the MCTIC corpus.

The evaluation results suggest the clear superiority of SBERT over doc2vec. Among the top 10 scores listed in Table 5.1, the highest score of doc2vec is roughly the same as the lowest score of SBERT. Analysis of the results also showed that in some cases, SBERT outperformed human judgment and found the paragraphs with higher similarity scores.

To better demonstrate the performance of the two models we provide the following examples of retrieved matching paragraphs.

Paragraph from the MCTIC corpus:

> Commonly referred to as the Maastricht Treaty, the Treaty on European Union (TEU) was signed at Maastricht in 1992. It also created the concept of European citizenship, although,

Table 5.1 Similarity scores for doc2vec and SBERT

doc2vec	SBERT
Ten Highest Scores	
0.58	0.86
0.57	0.86
0.57	0.85
0.57	0.85
0.57	0.85
0.57	0.83
0.57	0.83
0.56	0.83
0.56	0.82
0.56	0.82
Highest Score	
0.58	0.86
Lowest Score	
0.37	0.57
Average Score	
0.51	0.70

> unlike those rights enjoyed by workers, citizenship did not fall within the legal order; it also created the 'three-pillar' structure of the EU: added to the central pillar of the Communities would be a second pillar to cover co-operation in foreign policy and security, and a third pillar to cover co-operation in justice and home affairs.

The doc2vec result (similarity score: 0.57) was as follows:

> In respect of partners of EU nationals the content of the Directive mirrors the judgment in Case 59/85 Reed, in which the ECJ held that by virtue of the principle of non-discrimination, a Member State cannot refuse a cohabitee of a worker who is an EU national the right to reside with the worker in so far as national law provides this possibility for its own nationals. As a result, Miss Reed, a British national, was allowed to remain in The Netherlands with her English cohabitee of five years.

The SBERT's retrieval was (similarity score: 0.65):

> The Treaty of Maastricht established citizenship of the EU which complements but does not replace national citizenship. Only nationals of the Member States are citizens of the EU. A Member State has exclusive competence to decide who its nationals are and its decision must be respected by other Member States (Case C-369/90 Michelletti; Case C-192/99 Kaur; Case C-200/02 Chen) but this is subject to a proviso that when a situation under consideration is within the scope of the Treaties Member States must pay due regard to EU law (Case C-135/08 Janko Rottmann).

Original question: *Which Treaty created the concept of European citizenship?*

Answer: *The Maastricht Treaty*

Although the doc2vec model found a match in the RC, the content of the retrieved paragraph had little in common with the incoming paragraph from the MCTIC corpus. However, the SBERT result is certainly of better quality since it includes the same key words as the original paragraph and it is possible to answer the question correctly.

The proposed methodology reduces the time and effort needed to create multiple-choice test items manually. In a scenario where many questions need to be created on the basis of the same source text, i.e., the in-classroom textbook, the newly obtained paragraphs provide further ideas to test developers and help avoid word-for-word repetition while formulating questions. Furthermore, the existing questions from the MCTIC corpus can be post-edited to reflect the content of a new paragraph. As proven by the use of translation memory-like matches, making changes to a provided text is faster and more efficient than typing it from scratch. In addition, since the new paragraph shares similar information with the original one, teachers and instructors can rest assured that new multiple-choice questions will not fall out of the scope of the syllabus and students who have mastered the material will be able to answer them.

6. User Evaluation

An expert in the field of EU law and translation was asked to look at the questions and answers that could be generated by the proposed method and evaluate their quality. Specifically, we present the expert with the source paragraph, TM matched paragraphs from the target corpus, the proposed question (that comes with the source paragraph), and the proposed answer (that comes with the source paragraph). The expert is asked to determine:

1. Whether the proposed question (from the source question) can be answered using the information from the matched paragraph.
2. Whether, in case it cannot be used straight away, it can be edited with minimal effort. And if yes, how.
3. If the proposed answer (from the source answer) is the correct answer for the proposed (and potentially minimally edited) question.
4. Whether, in case the answer to the question number 3 is negative, the proposed answer can be minimally edited to become the correct answer, and how.

Out of 96 QA pairs proposed by the engine, the expert indicated that:

Twenty-seven (27%) of the proposed QA pairs can be used straight away, without any editing.
Seven (7%) of the proposed QA can be used with some editing to the answers only.
Nineteen (19%) of the proposed QA pairs can be used with some editing to the questions, but not the answers.
Four (4%) of the proposed QA can be used with some editing to both questions and answers.
In total, 57% of the suggested QA can be used, without, or with some editing.

Given the complexity of the subject matter, it is very encouraging that 57% of the generated questions and answers can be used without or with some editing (which is taken to include modifications such as insertions and deletions, as well as corrections). The temporal effort required in the 30% of cases that require editing (either in the questions, the answers, or both questions and answers) is minimal, while the temporal and technical effort require to produce the questions and/or answers from scratch would be higher. The expert reached this conclusion by comparing the time spent to produce the questions and answers from scratch to the time required to edit the automatically generated questions and answers. Examples of the expert's evaluation are in Table 5.2.

Table 5.2 Qualitative Evaluation by an Expert in EC Law

TM Matched paragraph	Proposed question	Proposed answer	Can proposed question be asked given the matched paragraph?	Can proposed question be edited so that it can be used as a question for the matched paragraph?	How would it be edited?	Is the proposed answer the correct answer for the proposed (and possibly edited) question?	Can the proposed answer be edited to become the correct answer?	How would it be edited?
The Americans supported the idea of political and economic integration in Europe since it would, in the long term, reduce the cost of their obligations and commitments in Europe. Robert Schuman considered that the best way to achieve stability in Europe was to place the production of steel and coal (then two commodities essential to conduct a conventional war) under the international control of a supranational entity. The creation of a common market for steel and coal meant that interested countries would delegate their powers in those commodities to an independent authority.	When did Churchill give his speech in which he suggested the idea of European unity?	In 1946	NO	NO	N/A	N/A	N/A	N/A
In June 1955 in Messina (Sicily), the foreign ministers of the Contracting States of the ECSC decided to pursue the establishment of a United Europe through the development of common institutions, a progressive fusion of national economies, the creation of a common market, and harmonization of social policies. From this materialized two treaties signed in Rome on March 25, 1957. The first established the European Economic Community (EEC) and the second the European Atomic Energy Community (Euratom). The treaties came into force on January 1, 1958. In time they became known as 'the Rome Treaties'.	What was the primary goal of signing the EURATOM and the EC Treaty?	Economic integration and the creation of a Common Market	YES	N/A	N/A	NO	YES	Economic integration, the creation of a Common Market, and harmonization of social policies.

(*Continued*)

Table 5.2 Qualitative Evaluation by an Expert in EC Law (*Continued*)

TM Matched paragraph	**Proposed question**	**Proposed answer**	**Can proposed question be asked given the matched paragraph?**	**Can proposed question be edited so that it can be used as a question for the matched paragraph?**	**How would it be edited?**	**Is the proposed answer the correct answer for the proposed (and possibly edited) question?**	**Can the proposed answer be edited to become the correct answer?**	**How would it be edited?**
The constitutional nature of the founding treaties, as well as the legal implications deriving from their peculiar status under public international law, has been progressively developed by the ECJ. In Case 294/83 Les Verts, the ECJ considered the EC Treaty as the basic constitutional Charter of the Community, and in Opinion 1/91 [Re First EEA Agreement] refused to interpret international agreements in the same manner as the EC Treaty because of the peculiar nature of the EC Treaty.	Which document resembled a kind of constitution for the Community?	The EC Treaty	YES	YES	N/A	YES	N/A	N/A
The founding treaties, as amended, are considered to be the constitutional treaties. The idea that the founding treaties establishing the three Communities are different from classical international treaties was recognized by the ECJ in Case 26/62 Van Gend en Loos, in which the Court held that 'this Treaty is more than an agreement which merely creates mutual obligations between the Contracting States' and that 'the Community constitutes a new legal order of international law', which creates rights and obligations not only for the Member States but more importantly for their nationals 'which become part of their legal heritage'.	Which document resembled a kind of constitution for the Community?	The EC Treaty	YES	YES	N/A	YES	N/A	N/A
The main powers conferred on the Council are defined in Article 16 TEU. This provision states: 'The Council shall, jointly with the European Parliament, exercise legislative and budgetary functions. It shall carry out policy-making and coordinating functions as laid down in the Treaties.'	What are the three main roles of Parliament?	Legislative, supervisory, budgetary.	NO	YES	What are the main roles of the Council?	NO	YES	Legislative and supervisory.

7. Future Work

Future research envisages experiments with high-performing DL models such as the Universal Sentence Encoder (Cer et al., 2018) and LASER (Artetxe & Schwenk, 2019), among others. Furthermore, since our methodology does not rely on language/domain-dependent features, we plan to expand this research to different languages and domains. Future work also includes experiments where questions are post-edited to correspond to the content of the RC paragraph. Automatic evaluation metrics such as traditional edit distance and METEOR score, or more recent ones like BLEURT (Sellam et al., 2020) or BERTScore (Zhang et al., 2020), are to be used to assess the post-editing human effort and success of the DL model used. After the automatic generation of distractors, complete multiple-choice items will be evaluated in terms of their usefulness and difficulty.

Additional future experiments include comparison with the original rule-based approach with the efficiency and quality of the generated multiple-choice tests to be assessed.

Notes

1 This chapter describes the first stage of this project: the generation of the multiple-choice stem in the form of a question. See also Maslak (2021) and Maslak and Mitkov (2021). The next stage of the project will cover the generation of distractors.

2 The concept of TM systems is, essentially, a simple one: the translator has access to a database of previous translations (referred to as a TM database), which he or she may consult, usually on a sentence-by-sentence basis, in order to find something similar enough to the current sentence to be translated. If a suitable example is found, it is used as a model. If an exact match is found, it can simply be cut and pasted into the target text. Otherwise, it can be used as a suggestion for how the sentence in question might be translated. Although the TM system will highlight the ways in which the example differs from the sentence to be translated, it is up to the translator to decide which parts of the target text to change.

3 TM systems are based on the retrieval and reuse not only of identical text fragments (exact matches) but also of similar source sentences and their translations (fuzzy matches). Most commercial TM systems are able quantify the quality of the match with a 'fuzzy score' or 'fuzzy match'. While most systems operate on character-string similarity, some incorporate additional heuristics such as formatting or indicative words. The character-string similarity is also referred to as 'string edit distance', as simply 'edit distance', or more formally as 'Levenshtein distance'. The edit distance is the minimal number of insertions, deletions, or substitutions necessary to change one string of characters into another. For instance, in order to convert 'memory' into 'memories', one deletion (y) and three insertions (i, e, and s) are needed.

4 *Word embedding* is a term used for the representation of words for text analysis, typically in the form of a real-valued vector that encodes the meaning of the word such that the words that are closer in the vector space are expected to be similar in meaning.

5 A *sentence encoder* takes a sentence or text as input and outputs a vector. The vector encodes the meaning of the sentence and can be used for downstream tasks such as text classification and text similarity. In these downstream tasks, the sentence encoder is often considered a black box, where the users employ it to produce sentence embeddings without knowing exactly what happens in the encoder itself.

6 Details about the pretrained models: www.sbert.net/docs/pretrained_models.html

7 The STS Benchmark (http://ixa2.si.ehu.eus/stswiki) contains sentence pairs with the human gold score for their similarity.

8 More details about the model, including the training data: https://huggingface.co/sentence-transformers/all-MiniLM-L12-v2

References

Afzal, N., & Mitkov, R. (2014). Automatic generation of multiple choice questions using dependency-based semantic relations. *Soft Computing*, *18*, 1269–1281.

Alsubait, T., Parsia, B., & Sattler, U. (2014). *Generating multiple choice questions from ontologies: Lessons learnt* (pp. 73–84). Proceedings of the 11th OWL: Experiences and Directions Workshop (OWLED2014).

Artetxe, M., & Schwenk, H. (2019). Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 597–610.

Cer, D., Yang, Y., Kong, S.-Y., Hua, N., Limtiaco, N., St John, R., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., & Strope, B. (2018). *Universal sentence encoder for English* (pp. 169–174). Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations.

Choi, E., He, H., Iyyer, M., Yatskar, M., Yih, W. T., Choi, Y., Liang, P., & Zettlemoyer, L. (2018). *QuAC: Question answering in context* (pp. 2174–2184). Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing.

Dasigi, P., Lo, K., Beltagy, I., Cohan, A., Smith, N. A., & Gardner, M. (2021, June). *A dataset of information-seeking questions and answers anchored in research papers* (pp. 4599–4610). Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of deep bidirectional transformers for language understanding* (Vol. 1, pp. 4171–4186). Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Long and Short Papers).

Gao, Y., Bing, L., Li, P., King, I., & Lyu, M. R. (2019). *Generating distractors for reading comprehension questions from real examinations* (pp. 6423–6430). Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19).

Joshi, M., Choi, E., Weld, D. S., & Zettlemoyer, L. (2017, July). *TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension* (Vol. 1, pp. 1601–1611). Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Long Papers).

Kaczorowska-Ireland, A. (2016). *European Union law*. Routledge.

Karamanis, N., Ha, L. A., & Mitkov, R. (2006). *Generating multiple-choice test items from medical text: A pilot study* (pp. 111–113). Proceedings of the Fourth International Natural Language Generation Conference.

Khashabi, D., Chaturvedi, S., Roth, M., Upadhyay, S., & Roth, D. (2018, June). *Looking beyond the surface: A challenge set for reading comprehension over multiple sentences* (Vol. 1, pp. 252–262). Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Long Papers).

Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., & Toutanova, K. (2019). Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, *7*, 453–466.

Le, Q., & Mikolov, T. (2014). *Distributed representations of sentences and documents* (pp. 1188–1196). International Conference on Machine Learning, PMLR.

Lee, D. B., Lee, S., Jeong, W. T., Kim, D., & Hwang, S. J. (2020). *Generating diverse and consistent QA pairs from contexts with information-maximizing hierarchical conditional VAEs*. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.

Liang, C., Yang, X., Dave, N., Wham, D., Pursel, B., & Giles, C. L. (2018). *Distractor generation for multiple choice questions using learning to rank* (pp. 284–290). Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications.

Maslak, H. (2021). *Automatic generation of multiple-choice test items with deep learning* [Unpublished master's dissertation, University of Wolverhampton].

Maslak, H., & Mitkov, R. (2021). *Paragraph similarity matches for generating multiple-choice test items* (pp. 99–108). Proceedings of the Student Research Workshop Associated with RANLP.

Mihaylov, T., Clark, P., Khot, T., & Sabharwal, A. (2018). *Can a suit of armor conduct electricity? A new dataset for open book question answering* (pp. 2381–2391). Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing.

Mikolov, T., Le, Q., & Sutskever, I. (2013). Exploiting similarities among languages for machine translation. *arXiv*. preprint arXiv:1309.4168. https://arxiv.org/abs/1309.4168

Mitkov, R., & Ha, L. A. (2003). *Computer-aided generation of multiple-choice tests* (pp. 17–22). Proceedings of the HLT-NAACL 03 Workshop on Building Educational Applications Using Natural Language Processing.

Mitkov, R., Ha, L. A., & Karamanis, N. (2006). A computer-aided environment for generating multiple-choice test items. *Natural Language Engineering, 12*(2), 177–194.

Papasalouros, A., Kanaris, K., & Kotis, K. I. (2008). *Automatic generation of multiple choice questions from domain ontologies* (pp. 427–434). Proceedings of the IADIS International Conference e-Learning.

Qi, W., Yan, Y., Gong, Y., Liu, D., Duan, N., Chen, J., Zhang, R., & Zhou, M. (2020). *ProphetNet: Predicting future n-gram for sequence-to-sequence pre-training* (pp. 2401–2410). Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings.

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016, November). *SQuAD: 100,000+ questions for machine comprehension of text* (pp. 2383–2392). Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing.

Ranasinghe, T., Mitkov, R., Orăsan, C., & Caro Quintana, R. (2021). Semantic textual similarity based on deep learning: Can it improve matching and retrieval in translation memory systems? In J. Lavid-López, C. Maíz-Arévalo, & J. R. Zamorano-Mansilla (Eds.), *Corpora in translation and contrastive research in the digital age* (pp. 102–124). John Benjamins.

Ranasinghe, T., Orăsan, C., & Mitkov, R. (2020). *Intelligent translation memory matching and retrieval with sentence encoders* (pp. 175–184). Proceedings of the 22nd Annual Conference of the European Association for Machine Translation.

Reimers, N., & Gurevych, I. (2019). *Sentence-BERT: Sentence embeddings using Siamese BERT-networks* (pp. 3982–3992). Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP).

Sellam, T., Das, D., & Parikh, A. (2020). *BLEURT: Learning robust metrics for text generation* (pp. 7881–7892). Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.

Shin, J., Guo, Q., & Gierl, M. J. (2019). Multiple-choice item distractor development using topic modeling approaches. *Frontiers in Psychology*, *10*, https://doi.org/10.3389/fpsyg.2019.00825

Singh Bhatia, A., Kirti, M., & Saha, S. K. (2013). *Automatic generation of multiple choice questions using wikipedia* (pp. 733–738). Proceedings of Pattern Recognition and Machine Intelligence: 5th International Conference, PReMI 2013.

Susanti, Y., Tokunaga, T., Nishikawa, H., & Obari, H. (2018). Automatic distractor generation for multiple-choice English vocabulary questions. *Research and Practice in Technology Enhanced Learning*, *13*. https://doi.org/10.1186/s41039-018-0082-z.

Turner, C., & Storey, T. (2014). *Unlocking EU law*. Routledge.

Xiao, D., Zhang, H., Li, Y., Sun, Y., Tian, H., Wu, H., & Wang, H. (2020). ERNIE-GEN: An enhanced multi-flow pre-training and fine-tuning framework for natural language generation. *arXiv*. https://arxiv.org/abs/2001.11314

Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W., Salakhutdinov, R., & Manning, C. D. (2018). *HotpotQA: A dataset for diverse, explainable multi-hop question answering* (pp. 2369–2380). Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2020). *BERTScore: Evaluating text generation with BERT*. Proceedings of the International Conference on Learning Representations.

Zheng, L., Guha, N., Anderson, B. R., Henderson, P., & Ho, D. E. (2021). *When does pretraining help? Assessing self-supervised learning for law and the case HOLD dataset of 53,000+ legal holdings* (pp. 159–168). Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law, Association for Computing Machinery. https://doi.org/10.1145/3462757.3466088

6

Training Optimus Prime, M.D.

A Case Study of Automated Item Generation Using Artificial Intelligence – From Fine-Tuned GPT2 to GPT3 and Beyond

Matthias von Davier

1. Introduction

The aim of this chapter is to provide evidence on the state of automated item generation (AIG) using deep neural networks (DNNs). Based on earlier work, a paper that tackled this issue used character-based recurrent neural networks (von Davier, 2018), the current contribution describes an experiment exploring AIG using very large transformer-based language models (Vaswani et al., 2017; Brown et al., 2020; BLOOM: https://huggingface.co/bigscience/bloom).

The chapter provides an overview of a case study that utilizes the latest generation of language models for text generation. In terms of significant stepping-stones, the description is based on the following developments:

a. GPT-2, OpenAI's model, was described in Radford et al. (2018). This chapter explains, among other things, how GPT-2 was retrained using millions of PubMed open access articles for the purpose of generating clinical vignettes. GPT-2 was superseded (in size) by MegatronLM (NVIDIA, 2019).
b. The next step, and one that made not only a huge splash in the media but also resulted in a large number of startups using NLG, was the release of the GPT-3 API, which allowed access to the currently most used transformer model, which clocks in at 175 billion parameters (Brown et al., 2020).
c. GPT-J-6B, a 6-billion-parameter model, and the more recent (February 2022) GPT-neoX with 20 billion parameters, were released. Some examples generated for this chapter are based on GPT-J. Both models are provided by www.eleuther.ai/, a self-described grassroots campaign of a 'decentralized collective of volunteer researchers, engineers, and developers focused on AI alignment, scaling, and open-source AI research, founded in July of 2020'.
d. BLOOM (July 2022), available through the Hugging Face portal (https://huggingface.co/bigscience/bloom) based on the BigScience collaborative open science initiative, is a language model trained on 46 natural languages. It aims at free worldwide access, while OpenAI's GPT models were (despite the name of the organization) proprietary and licensed to individuals and organizations.

DOI: 10.4324/9781003278658-8

This chapter has been made available under a CC-BY-NC-ND 4.0 license.

These most recent developments, BLOOM as well as GPT-neoX, give reason to hope that access and work with these language models is further improving and that researchers who were unable to participate in work due to language, economic, or political barriers will be able to engage in applications of and research on large language models again. Moreover, BLOOM is based on what BigScience calls a Responsible AI Licensing agreement (RAIL: https://huggingface.co/spaces/bigscience/license), which includes a section that restricts the use of this model for purposes that can lead to discrimination, physical or emotional harm, or the dissemination of misinformation. An increasingly important aspect of the applications of AI is the responsible and accountable use of these increasingly potent technologies.

Some of the recent neural network-based language models include more than 175 billion parameters, which is incomprehensible compared to the type of neural networks that were used only a few years back. In the winter semester of 1999–2000, I taught classes about artificial neural networks (NNs) – for example, Perceptrons (Rosenblatt, 1958) or Hopfield networks (Hopfield, 1982). Back then, artificial intelligence (AI) already entered what was referred to as the 'AI winter', as most network sizes were limited to rather small architectures unless supercomputers were employed. On smaller machines that were available to most researchers, only rather limited versions of these NNs could be trained and used, so successful applications were rare, even though one of the key contributions that enabled deep learning and a renaissance of NN-based AI, the long-short-term-memory (LSTM) design (Hochreiter & Schmidhuber, 1997) was made in those years. In 2017, I started looking into neural networks again because I wanted to learn how to program graphical processing units (GPUs) for high-performance computing (HPC) as needed in estimating complex psychometric models (von Davier, 2016). After experimenting with high performance computing for analyzing PISA data (which cut down estimation of IRT models from several hours to 2–3 minutes using the parallel-E parallel-M algorithm developed in 2016), this finally led me to write a paper on using deep neural networks for automated item generation (AIG; von Davier, 2018). AIG is a field that has seen many different attempts, but most were only partially successful, involved a lot of human preparations, and ended up more or less being fill-in-the-blanks approaches such as we see in simple form as MadLibs books for learners.

While I was able to generate something that resembled human written personality items, using a public database that contains some 3,000, several of the (cherry-picked) generated items sounded and functioned a lot like those found in personality inventories (Goldberg, 1999; Goldberg et al., 2006). I was somewhat skeptical whether one would be able to properly train neural networks for this task, given that it would require a very large number of items, and I assumed that each network for that purpose would need to be solely trained on items of the form it is supposed to generate. Part of my concern was that the items that were generated had to be hand-picked, as many of the generated character or word sequences ended up not being properly formed statements. However, those that were selected for an empirical comparison with human-coded items were found to show the same dimensionality (von Davier, 2018) and hence to be fully useful as replacements of human-authored items. Nevertheless, some doubt remained due to the needed handpicking and the limited supply of training material. After all, AI and neural networks have a long history (e.g., Rosenblatt, 1958; Wiesner, 1961) and have been hyped to be the next big thing that may soon replace humans and take our jobs.

As mentioned, items generated using RNNs (von Davier, 2018), then cherry-picked, were passing empirical evaluations and hence functioned a lot like the human-written items in an online data collection. However, many of the generated items were either not properly formed statements that are typical for this domain, or, if the network was trained too long on too little data, they were almost exact copies of what was entered as training material. Therefore, I concluded one would need a lot more data, or an unforeseen qualitative jump in deep learning that I expected to be years away. This was wrong; it turns out that time indeed flies, and the field of deep learning did not rest, and while in the paper published in 2018 I stated that operational use could be years away, I am not so sure anymore that we have to wait that long.

It may well be that we will see automated item generation based on deep learning systems soon in tools that support item writers for developing test questions for high-stakes exams, and that deep neural networks will be used to generate questions or distractors for multiple-choice questions used in test preparation and practice exams much sooner. The reason why I believe this has to do with a graduate student who developed a software tool for programmers based on a product that was released by OpenAI (Radford, 2018). The software that supposedly makes programmer lives so much better is called TabNine (e.g., Vincent, 2019) and it provides context-sensitive (intelligent?) auto-completion based on indexed source code files. The author of the software estimates that TabNine will save programmers at least 1 second per minute by suggesting how lines of program code are completed, or what the most likely next line of code may be, based on the code that the programmer provides and the software uses to improve a predictive model.

The title of the current chapter is a reference to two relevant lines of inquiry. There was an article with the title ‘Doctor A.I.’ (Choi et al., 2015), which described a deep learning approach using generative adversarial networks (GANs) to generate electronic health records (EHRs) that can pass as plausible EHRs, and the other is the recently ignited race around language models that use a specific neural network structure called transformer, which was an obvious trigger for many references to the sci-fi toys and movies. The remainder of this chapter is structured as follows: The next section introduces language models that are based on approaches that can be used to generate the probability of a next word or language token using information about a previously observed sequence of words. The following section outlines potential areas of application and shows select examples of how NN-based language models could be utilized in medical licensure and other assessment domains for AIG.

2. Background and Significance

AIG has been an area of research in the field of employment and educational testing for quite some time (Bejar, 2002). Employing human experts to develop items that can be used in medical licensing and certification is particularly cost-intensive, as expert knowledge is needed to author case vignettes and to develop plausible response options when writing multiple-choice test questions. Any technology that can reduce these development costs by applying machine learning or AI would be a welcomed addition to the toolbox of test developers. AIG often either focused on items that are language free, such as intelligence tests with matrices of graphical symbols that need to be completed by test-takers (Embretson, 1999), or employed methods that amount to something that bears strong similarities to fill-in-the-blanks texts such as the ones found in MadLibs.

The current work builds on and extends a study presented by von Davier (2018), in which a RNN was trained on an open access database of 3,000 items available through the IPIP database (Goldberg, 1999). While this previous study concluded that with existing recurrent network-based models, and with limited item banks, a practical use of AI for AIG would be years away, the development of language models took a quantum leap when researchers did away with recurrence and focused on network architectures built around self-attention (Vaswani et al., 2017). This allowed designing a simple network structure that was easily trained, allowed parallelism in training, and could be pretrained on general corpora of texts and subsequently trained for specific purposes.

Retraining (and re-implementation) of the transformer has led to a variety of applications, including the generation of poems, patent texts, and completion of code in support of software developers. These applications will be references in appropriate sections over the remainder of this chapter. This chapter also describes a similar experiment with the goal to provide a tool for developing medical education test items using deep learning–based language models.

3. Materials and Methods

The basis of these predictive approaches are sequential models that provide the probability of the next word (or other language token such as full stop, newline, etc.) given a number of previous words. These models are not new, I recall my first encounter of this type of model was an article in *Scientific American* before 1985, when I was still a high school student and part-time programmer working for a small educational gaming company located in northern Germany (yes, game-based learning existed back then). This 1980 version actually goes back to the seminal paper by Shannon (1948) and constitutes a primitive language model. This simple model of course did not have the many layers and the complex network architecture of deep learning applications that are nowadays used for machine translations, picture annotations, or automated item generation (von Davier, 2018); rather, it was based on a single layer that connected an input word (previous encounter) to an output word (next encounter). Technically, the basis of this model was a transition matrix, with input (previous) and output (next) words coded as binary vectors, and the model basically implemented the Markov assumption for a model for natural language.

3.1 Markovian Language Models

The model just mentioned is a simple language model that can be viewed as direct translation of the Markov assumption for modeling a sequence of words $w_t \in \Omega_V$ with index $t = 1,\ldots,T$. Here, Ω_V is a finite set of words, the vocabulary of a language, and $S = |\Omega_V| < \infty$ denotes the size of the vocabulary. Let $\omega : \{1,\ldots,S\} \mapsto \Omega_V$ be an index, i.e., a bijective function that maps integers to words. That is, we can obtain an integer that represents a word w_t by applying $i_t = \omega^{-1}(w_t)$, and the associated word can be retrieved from any integer $i_t \in \{1,\ldots,S\}$ through $\omega(i_t)$.

In this most simple case of a language model, we assume that

$$P(w_{t+1} \mid w_t,\ldots,w_1) \equiv P(i_{t+1} \mid i_t,\ldots,i_1) = P(i_{t+1} \mid i_t) \equiv P(w_{t+1} \mid w_t)$$

for any $t \in \{1,\ldots,T-1\}$, namely that the probability of observing a next word w_{t+1} at position $t+1$ of the sequence depends only on the last observed word, w_t, and nothing else. The whole sequence preceding the next-to-last word is ignored in this model. Then, if we assume homogeneity of the transitions, i.e., $P(\omega^{-1}(w_{t+1}) \mid \omega^{-1}(w_t)) = P(\omega^{-1}(w_{u+1}) \mid \omega^{-1}(w_u))$ whenever $w_t = w_u$ and $w_{t+1} = w_{u+1}$, we can define

$$M_{t|t-1} = \left(P(1|1) I P I \ldots P(i|i) I P I \ldots P(S|S)\right),$$

which is a transition matrix that provides a conditional probability distribution for any $i = \omega^{-1}(w)$. If there are no constraints, this transition matrix has $S[S-1] = SS - S$ parameters, i.e., roughly the square of the vocabulary size. The parameters can be obtained by estimating simple sample statistics, or by some more sophisticated methods (e.g., Shannon, 1948).

A more complex language model would consider more than one previous word. This can be implemented as follows. In order to take the previous L words into account, define $n_t = \left(w_t, w_{t-1},\ldots,w_{t-(L-1)}\right) \in \otimes_{i=1}^{L} \Omega_V$, which is an n-gram of length L.

Then assume for $t > L$ that

$$P(w_{t+1} \mid w_t,\ldots,w_1) = P(w_{t+1} \mid n_t) = P\left(w_{t+1} \mid w_t, w_{t-1},\ldots,w_{t-(L-1)}\right).$$

While this is a perfectly sound definition, it has practical implications that may make applications impossible, as soon as the vocabulary contains more than a few handful of words and the length of sequence, L, grows larger than, say, 3. The issue is that the mini-sequence n_t is an element of $\otimes_{i=1}^{L}\Omega_V$, a much larger set, with S^L elements. For a vocabulary of only 100 words and three-word sequences, there are already $100^3 = 1{,}000{,}000$ different elements.

For a transition matrix that contains all conditional probabilities for the next words, given the previous three, we would need to train, estimate, or otherwise obtain $(100-1)\times 1{,}000{,}000 = (S-1)\times S^L$ probabilities. Therefore, most traditional approaches to construct such a large transition matrix have not been pursued, as this would require very large amounts of data.

3.2 Char- and Word-RNNs

One way of circumventing the need to use classical statistical estimation methods, and to be able to ignore some of the more rigorous requirements of these methods, is using NNs for the purpose of language modeling. NNs have been shown to be universal function approximators (e.g., Hornik, 1991; Hanin, 2017). This means that an NN with proper design can be used to plug in an estimate of a function that is otherwise hard to calculate, or hard to specify based on more traditional approximation or estimation methods. This advantage is paid for by having only vague knowledge about the actual form of the function that is being approximated, as NNs operate as black boxes and do not easily reveal how the approximation is achieved.

In order to further reduce demands, one could model the sequence of characters rather than words, as natural languages often contain several thousand words, while alphabetic languages can be expressed using a much smaller character set. Therefore, an alternative to word-based language models using neural networks can be implemented as a character-based language model. A few years ago, Google released TensorFlow (Abadi et al., 2015), a powerful software toolbox to design, train, and sample from neural networks. This triggered implementation of a variety of deep learning approaches using this new tool, among these a character-based deep recurrent neural network (Char-RNN, e.g., Ozair, 2016), and, more recently, other architectures that will be described in this chapter. Obviously, there are many more tools for deep learning, and the models released for further analyses and fine-tuning, as done in the current study, are typically available in more than one framework.

Wikipedia provides a list of neural network tools, specifically, deep learning–oriented tools, at https://en.wikipedia.org/wiki/Comparison_of_deep_learning_software.

4. Attention Is All You Need

Recent language models introduced the concept of attention, as a structure that was part of the neural network architecture aimed at keeping certain concepts more salient. This was initially implemented in addition to the recurrent structures of deep learning models designed for sequence-to-sequence and language modeling. However, Vaswani et al. (2017) proposed an alternative, much simpler structure in which the context and the attention mechanism would replace the sequential structures of RNNs. The title of Vaswani et al.'s article is mirrored in the subsection title, and this article led to multiple language models published in short succession, one of which was recently released by OpenAI and forms the basis of the retrained/fine-tuned model presented in this chapter.

Vaswani et al. (2017) describe the new network structure as consisting only of decoder-encoder layers with multi-headed attention, which provides a distribution of most likely language tokens, given a context of a certain length (say, 1,024 words and information about their position). Psychoanalysts would probably say that transformers simulate some form of free association, noting that this is even called self-attention in the literature. Interestingly, the attention architecture used in the transformer-based models is simpler than what was previously deemed necessary in language models based on recurrent neural networks such as the one used in Ozair (2016) and Brown et al. (2020). This simpler structure allows much faster training, as the transformer architecture allows parallel processing by means of simultaneously using word and position encoding rather than encoding the text sequentially. The drawback is that (currently) only limited lengths of text can be encoded, as the parallel processing makes it necessary to have the sequence to be encoded (input) as well as the output to be present as a whole (for example, sentence-by-sentence), rather than word-by-word.

5. Reincarnations of the Transformers: GPT-2, Transformer-XL, Grover, MegatronLM

The GPT-2 model was trained by a team of researchers at OpenAI (Radford et al., 2018) using four different levels of complexity of the transformer architecture. In an unprecedented move, OpenAI released only the two smallest models, which comprise network weights amounting to 117 million and 345 million parameters, respectively. The larger models are not published due to concerns of malicious use cases and contain up to 1.4 billion (!) parameters. However, this number was recently toppled by NVIDIA, publishing the MegatronLM model that includes more than 8 billion parameters, and making the code available on GitHub (https://github.com/NVIDIA/Megatron-LM). However, the 1.4 billion OpenAI parameter model remains unpublished, as it says on the OpenAI website:

> Due to our concerns about malicious applications of the technology, we are not releasing the trained model. As an experiment in responsible disclosure, we are instead releasing a much smaller model for researchers to experiment with, as well as a technical paper.

All GPT-2 models were trained on what OpenAI called WebText, which is a 40 GB database of text scraped from the World Wide Web, excluding Wikipedia, as OpenAI researchers assumed that Wikipedia may be used by secondary analysts to retrain/fine-tune for specific topics. As the full model is not available, this means that the actual performance of the GPT-2 Transformer model cannot be verified independently, and other researchers can only use and modify (retrain) the smaller models. The examples presented in this chapter are based on experiments with the model that contains 345 million parameters.

Several other transformer-based language models have been under active development and are being made available to researchers for fine-tuning and adaptation to different applications. Among these are the Transformer-XL (Dai et al., 2019), Grover (Zellers et al., 2019), and, most recently, MegatronLM (NVIDIA, 2019). While the NVIDIA model used a corpus called WebText that contains 40 GB of data and was modeled after the corpus used by OpenAI, Grover was trained on 46 GB of real news and can be used to either generate, or detect, fake news.

This ability to both detect and generate is based on the fact that all of these approaches can be viewed as probabilistic models that predict a sequence of new words (fake news, a translation, next poem lines, next syntax line in a software program) based on the previous sentence(s) or lines of code. More formally, we can calculate the loss function

$$H\left(T,\hat{P}\right) = -\frac{1}{T}\sum_{t=1}^{T} log\,\hat{P}\left(w_t \mid n_{t-1}\right),$$

where $\breve{P}\left(w_t \mid n_{t-1}\right)$ is the estimated distribution of word w_t given context (history) n_{t-1}. This is an estimate of the cross entropy, or logarithmic entropy (Shannon, 1948) of the observed sequence $w_0,\ldots,w_t$ given some initial context η_C. This quantity can be used to evaluate generated sequences relative to the distribution of the loss based on true (human-generated) sequences to help distinguish them. The cross entropy is a measure of how well predicted (in terms of expected log-likelihood, e.g., Gilula & Haberman, 1994) an observed sequence is if a certain model $\breve{P}$ is assumed to hold. This loss function is also used during training or fine-tuning in order to evaluate how well the network predicts new batches of data that are submitted to the training algorithm.

It is worth mentioning that while all of these are variations on a theme, the transformer architecture for language modeling has shown great potential in improving over previous designs in terms of performance on a number of tasks (Devlin et al., 2018). In terms of the use for generating test questions, Grover (Zellers et al., 2019) may prove useful in future applications, as it was designed to produce and detect fake news by using 46 GB worth of data based on actual news scraped from the internet. Retraining Grover with targeted assessment materials around a content domain is one of the future directions to take for applied research into automated item generation using NN-based language models.

6. Method and Generating Samples

The applications of deep learning and recurrent neural networks as well as convolutional networks range from computer vision and picture annotation to summarizing, text generation, question answering, and generating new instances of trained material. In some sense, RNNs can be viewed as the imputation model of deep learning. One example of medical applications is medGAN (Choi et al., 2016), a generative adversarial network (GAN) that can be trained on a public database of EHRs and then used to generate new, synthetic health records. However, medGAN can also be considered an ‘old style’ approach, just as the approach I used is for generating personality items (von Davier, 2018), as medGAN was not based on a pretrained network that already includes a large body of materials in order to give it general capabilities that would be fine-tuned later.

Language models as represented by GPT-2 are pretrained based on large amounts of material that is available online. GPT-2 was trained on 40 GB of text collected from the internet but excluding Wikipedia, as it was considered that some researchers may want to use this resource to retrain the base GPT-2 model. These types of language models are considered multi-task learners by their creators, i.e., they claim these models are systems that can be trained to perform a number of different language-related tasks such as summarization, question answering, and translation (e.g., Radford, 2018). This means that a trained model can be used as the basis for further targeted improvement, and that the rudimentary capabilities already trained into the model can be improved by presenting further task-specific material.

7. AI-Based AIG Trained on Workstations With Gaming GPUs

While this should not distract from the aim of the chapter, it is important to know that some considerations have to be made with respect to how and where calculations will be conducted. Software tools used for deep learning are free (Abadi et al., 2015), and preconfigured servers and cloud services exist that facilitate the use of these tools. At the same time, significant costs are involved, and in particular researchers who develop new models and approaches may need multiple times more time and resources compared to standard applications that are used to analyze data. The dilemma is that while most tools for training deep learning systems are made freely available, these tools are worthless without powerful computers. And pointing to the cloud is not helpful, as the cloud is ‘just someone else’s computer’ (as memes and geek merchandise prove): High-performance hardware and algorithms that employ parallelism are needed to train these kinds of networks, either in the form of hardware on-site, in a data center, or rented through the cloud. The training of RNNs as well as transformer-based language models takes many hours of GPU time, which comes at significant costs if the cloud is used. For recent language models of the type of GPT-2 large (1.4 billion parameters), or Grover-Mega, or XLNet, the estimated cost was around $30K–$245K (XLNet) and $25K (Grover-Mega). More details can be found at Sarazen and Peng (2019) as well as in online forums discussing the training and retraining of these models.

Obviously, cloud computing services come at a cost, and while new preconfigured systems pop up daily and prices will decrease due to reduced hardware cost and competition, any more-involved project that requires training specialized systems, or retraining existing large models, will incur significant costs as well. The model used in the current paper was pretrained on several TPUs (specialized Google hardware for tensor computations) for over a week and retraining as well as fine-tuning will take weeks of GPU time in order to produce a system that is useful for a specific purpose. Therefore, building or purchasing a deep learning computer is one of the options that should be carefully considered as well as the use of cloud computing or on-demand GPU time such as Vast.AI. Nowadays, even modest hardware such as gaming

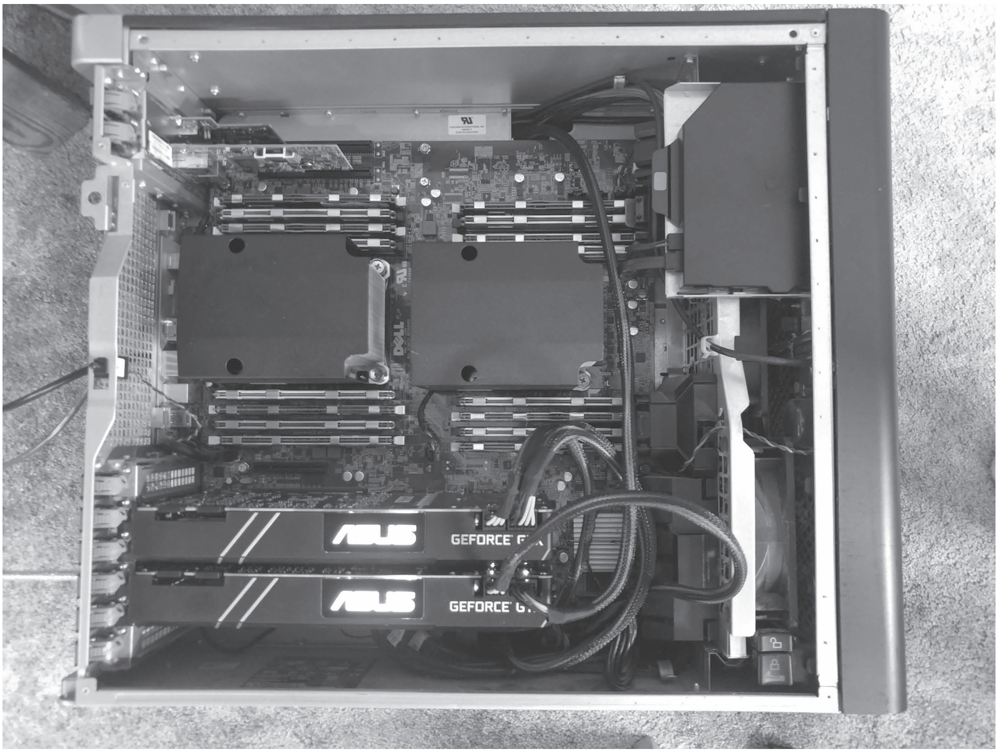


Figure 6.1 Server parts from eBay used to provide the official PISA 2015 data analysis, and now upgraded and re-purposed for automated item generation. All you need is processor cores, RAM, GPUs, and an eBay auction sniper. While cloud computing is an option, the experiments reported here are time-consuming and cloud computing is currently available at an on-demand rate of $0.80/hour ($0.45 pre-ordered) per GPU. Retraining took 6 days on two GTX 1080Ti GPUs obtained and installed in a 2013 T7610 Dell dual Xeon processor workstation.

desktops can be utilized, as most of these contain powerful GPUs for graphical processing, which can be turned into thousands of processing units through toolkits such as CUDA provided by the makers of these graphics cards (e.g., Chevitarese et al., 2012).

The hardware needed for training large NNs can be found at specialized vendors such as Lambda Labs, who often also provide turnkey solutions such as operating system images that include all the common machine learning toolkits such as KERAS, TensorFlow, PyTorch, and others. An alternative is to DIY and to use the many web resources that describe which workstations can be obtained cheaply and how many of the essential GPUs can be housed, with or without modifications. In addition, there are free web resources – for example, Google Colab which is essentially a Jupyter Notebook that anyone with a Google account can use for deep learning and machine learning experiments (free for short-term use), or time-share on-demand GPU services such as Vast.AI can be used for a fee.

Without further digressions, we now turn to how these systems, either purchased fully configured as turnkey solutions, or put together from used parts, can be utilized to produce text that, to a much greater extent than imaginable only two years ago, can facilitate automated generation of assessment materials, including the generation of electronic health record, the production of suggestions for distractor choices in multiple-choice items, and the drafting of patient vignettes based on prompts provided by item writers.

8. Electronic Health Records and Deep Learning

The fact that medicine uses IT for storing and managing patient data brought with it that computer scientists were needed and hired to work on systems for this purpose. At the same time, data on patients, as it is stored in electronic health records (EHRs), is highly sensitive, so developers working in this area looked for ways to use databases that would not directly reflect anyone's real data. One way was to use the same data, carefully anonymized so that individuals cannot be identified. A second approach was to generate health data of nonexistent patients using the regularities found in real health data.

This was the birth of synthetic EHRs, either in the form of expert-generated models (Synthea: Walonoski, 2017) or in the form of deep learning–based models that either predict the next clinical event or generate (plausible) synthetic EHRs based on a training dataset of real EHRs (Dr. AI: Choi, 2015; MedGAN: Choi, 2016). These models can be used to generate data that can be considered a limitless resource, and they are anonymous by design so that concerns about privacy and data protection are alleviated when using these synthetic records in design and analysis work.

A recent systematic review (Xiao, 2018) describes 98 studies that use deep learning in conjunction with EHRs. The studies range from generating synthetic EHRs to enable users to experiment with these data without any privacy issues, to building predictive models that are used to forecast the next clinical encounter based on existing patient history. EHRs are an important source of information and can be used to look at systematic differences in trajectories between patient groups, as well as how different treatments play out by comparing the prevalence of subsequent encounters.

9. Distractor Generation Through Question Answering

The utility of the GPT-2 language model is currently explored by means of retraining for specific purposes. One application that was mentioned in the introduction is the TabNine software that allows the prediction of line completions and programming code continuation. For that purpose, each user provides data of their own project-related code, which can further improve prediction.

Table 6.1 Example Output of the 345M Parameter Model When Used for Question Answering

Model prompt >>> Q: What are the most common side effects of statins? A:

SAMPLE 1: They can lead to a reduction in the effectiveness of your insulin. They can be associated with an increased risk of heart attack and stroke

SAMPLE 2: Common side effects are: nausea, vomiting, diarrhea, weight gain, joint pain . . .

SAMPLE 3: Statins can cause stomach inflammation or stomach ulcers. . . .

SAMPLE 4: There are some common side effects to statins, but these are not always permanent. For example, people with cardiovascular disease tend to have more fatigue.

SAMPLE 5: These can include: dizziness, feeling like your eyes are going to pop out (this is called 'repetitive eye movements'), muscle pain, blurred vision

Table 6.2 Responses to a Question About Symptoms of IBS Using a Network Trained for 3 Days

Model prompt >>> What are the most common symptoms of irritable bowel syndrome?

SAMPLE 1.1 In chronic condition, patients report numerous symptoms including pain, bloating, abdominal pain, diarrhea, constipation, and vomiting. Those patients who have diarrhea, cough, or abdominal pain are often affected by this disorder . . . .

SAMPLE 1.2 In 2014, an estimated 3.9 million U.S. adults reported more than 1 or 2 symptoms of irritable bowel syndrome. About 30 percent reported at least one symptom. In this survey, 63.8 percent IAmericans . . .

SAMPLE 1.3 Symptoms of irritable bowel syndrome tend to be more severe in those with irritable bowel syndrome, which may make them the least likely to seek diagnosis and treatment. Some symptoms that may be present include abdominal In, bloating, . . .

SAMPLE 1.4 Symptoms of irritable bowel syndrome Complications of irritable bowel syndrome Symptoms of irritable bowel syndrome. Control Medication Inflammation Abnormal bowel movements. . . .

SAMPLE 2.1 Symptoms of irritable bowel syndrome No. of Possible Causes 8.2 10.7 Major Common Symptoms – upper and lower gastrointestinal issues. 1. Irritable bowel syndrome with diarrhea (ILS) has been described as . . .

SAMPLE 2.2 It is common for the majority of irritable bowel syndrome patients to experience a variety of symptoms including abdominal pain, cramping, bloating, nausea, abdominal pain, bloating, diarrhea, cramps, bloating, cramping . . .

SAMPLE 2.3 My best friend and I recently purchased a new home to be closer to family and friends, and we felt more comfortable living at home. However, due to some suspicious and unusual events, we were contacted by your physician to take our . . .

SAMPLE 2.4 Symptoms of irritable bowel syndrome are often severe and often require hospitalization. The symptoms of diarrhea (1.2%) and constipation (1.4%) are common in the community. Severe diarrhea and const . . .

The ability to generate distractors for existing multiple-choice items is already given even in the original, not-retrained 345M GPT-2 model. The training material contained a large number of cases in which a question was followed by the prompt 'A:' and an answer. By means of introducing this type of training material, the model was enabled to react to a prompt that ends in 'A:' with a continuation of the text that in many cases represents a meaningful response to a question that was contained in the prompt. Here, we show two examples that were generated using a local copy of GPT-2 on the author's workstation equipped with a GPU and TensorFlow (Abadi et al., 2015), which facilitates sampling from the language model.

The GPT-2 model was trained with material that includes text that has the structure: 'Q: What is X? A: X is a Y.' In other words, the model is prompted to associate a sequence of words that is bracketed in 'Q:' and 'A:' as a question that requires an answer. The next figure shows an example of output generated using the 345M model. Note that these are far from perfect, but they could serve as inspiration for human item writers. The first example (Table 6.1) was generated without any retraining, using the downloadable version of the 345M GPT-2 model.

It is clear that not all of the listed side effects are actual ones patients may experience. However, some overlap with side effects mostly listed in online resources, and some others may be 'plausible enough' to potentially serve as wrong options in a multiple-choice test. The next example (Table 6.2) asks about common symptoms of IBS; the selection of responses were not cherry-picked, and from among two sets of 4 answers, most are on topic.

It is important to note that the responses are based on a general language model that has not been trained specifically to answer questions about medical content. This model is, on top of that, the second-smallest of the GPT-2 models, and contains (by today's standards) only 345 million parameters, while other, larger variants contain much more complex model layers and approximately 1.4 billion parameters (Radford et al., 2018). Again, note that these responses that could potentially be used as distractor suggestions were generated without any retraining of specifically medical assessment materials.

10. Automatic Item Generation

The tests reported in this section are based on the GPT-2 (345M) pretrained language model and roughly 800,000 open access subset articles from the PubMed collection (www.ncbi.nlm.nih.gov/pmc/tools/openftlist/) used for retraining. The data was encoded using the GPT-2 (https://github.com/nshepperd/gpt-2) toolbox for accessing the vocabulary used for pretraining and fine-tuning GPT-2 using TensorFlow. The 800,000 articles roughly equate to 8 GB worth of text from a variety of scientific journals that allow open access to some or all of their articles. Training took 6 days on a Dell T7610 equipped with 128 GB RAM, two 10-core Intel Xeon processors, and two NVIDIA 1080 Ti GPUs using CUDA 10.0 and TensorFlow 1.14, Python 3.6.8 and running Ubuntu 18.04 LTS. It was necessary to use the memory-efficient gradient storing (Gruslys et al., 2016; Chen, 2016) options, as the size of data structures for the 345M model used in the retraining exceeded the 11 GB memory of the GPUs without it.

The amount of training data available through open access (OA) papers that can be downloaded from PubMed repositories is quite impressive: The number of OA articles exceeds 800,000, and the compressed pre-processed databases used for retraining in this study exceeds 8 GB. However, free medical texts are available in abundance, and a 2011 survey (Singh et al., 2011) lists many resources. Language models for data of this size were not able to be processed on customary hardware only a few years ago, while nowadays (with a few tricks), even the medium-size (345 million hyper parameter) GPT-2 model can be retrained on decent gaming GPUs.

Incidentally, during the 6 days of training there is some downtime, which allowed me to find a recent arXiv submission that talks about automated patent application generation using GPT-2 pretrained with (public, granted) patent applications available online (Lee, 2019). Other applications include the syntax completion software TabNine described in the introduction as well as experiments aimed at automatic generation of poems (Gwernnet, 2019). The authors of the GPT-2 patent retraining study used Google Colab, a free online deep learning platform that allows users access to GPUs for NN training for up to 12 hours. This is insufficient for the 8 GB of PubMed data to be fully retrained on GPT-2 medium, so the author of this chapter resorted to upgrading and using a dual GPU workstation.

Table 6.3 shows exemplary results after 2 days of retraining with the 800,000-article PubMed database. Other publicly available medical text and article databases are listed in Singh et al. (2011). While the results are encouraging, they can certainly not be used as is, when produced by the NN. However, some editing and human expert input could use this raw output as inspiration for authoring clinical vignettes. Results should be closer to human-authored item stems using a transformer that was trained on a large number of real medical licensure and certification item stems and distractors, and as larger pretrained transformer models get published. It should be noted that these early examples are certainly not texts that would pass as real items.

Table 6.3 Example 1 (After Only 2 Days of Training on Customary Easily Obtainable Hardware) of Generated Text That Could Potentially Be Used as the Basis for Item Stem Authoring

Model Prompt >>> A 52-year-old man complaining about chest pain and headache is coming to the emergency room.

PubMed (2 days) retrained, GPT-2 based model-generated texts:

SAMPLE 1: His blood pressure is 143/99 (normal). He also has hypertension and urticaria. It is a potentially life-threatening situation. The first piece of information from the patient is that he has on occasion been suffering from chronic abdominal pain, abdominal pain during pregnancy, chronic abdominal pain during childbirth, urinary tract symptoms. All these are related to chronic abdominal pain. He is conscious and willing to go to the emergency room for assessment. However, he is already complaining about abdominal pain when we come to the hospital. My primary concern is the possibility of heart failure due to chronic pain. I don't feel that I am in a position to explain this to the patient, but I want to reassure him that it is a possibility. So, the decision to take him to the emergency room for evaluation is already made. Some would argue that he would be better off in the hospital, but that does not seem to be the case. When it comes to dealing with chronic pain, we can do better with a diagnosis of dysarthria, rather than acute cardiomyopathy. We can give him three days of antibiotics. We can tell him that he will need to have three small intestinal procedures in the hospital, but that he can go home

SAMPLE 2: He also complained of tingling and aching in his hands, feet, legs, and arms. The patient is not in good condition. He had a history of a recent allergic reaction to peanuts. He took 4 mg. of polyphenols: red chip, red, black and white, in combination with 250 mg. of oat milk and 60 mg. of vitamin B12. He immediately experienced tingling sensation in his hands and feet. In addition, he had a mild fever. He did not take medicine for a long time. He then started to complain of a very severe pain in his chest and abdomen. His symptoms became worse. He felt cold, tingling sensation, and headache. He was taken to the hospital, and there a further examination showed that he was allergic to peanuts. A follow-up examination confirmed that he was allergic to peanuts. He was discharged home the same day. A change in his allergy history showed that he had not taken medicine for a long time, and he had a mild fever. He started to take food for a long time. He started to take vitamin B12. He started to take oat milk as well. He started taking 500 mg. of vitamin B12 in combination with 150 mg. . . .

SAMPLE 3: The patient was admitted to the hospital on Friday night with a minor hip fracture. The patient was experiencing abdominal pain, nausea, and fever. He did not report any cough, other than the cough of headache and chest pain. The patient was first treated with antibiotics, then with polyethylene glycol solution (PEG), and finally with chloramphenicol. At the end of treatment, the patient's abdominal pain and abdominal pain continued. The patient told his doctor that he did not feel well, but did not report any cough, cold, or other symptoms. On the following day, the patient developed a fever of 40.2°C and some abdominal pain. On the following morning, he reported no other symptoms, but fever of 40.2°C and fever of 39.2°C. The patient's temperature was recorded as 49.3°C, and his level of red blood cell count was 120/16. The patient was prescribed phenylalanine (PBA), but did not report any other symptoms. The patient's BP was recorded as 118/28, his level of red blood cell count was 126/17. The patient was then referred to an oncology facility for further treatment.

SAMPLE 4: His symptoms were described as sudden onset of chest pain with a brief onset of bradycardia. The patient had a history of transient rapid, transient, and rapid diastolic blood pressure increases. Patient in critical condition for dehydration, 5-day interval between study. Recurrent hypertension and pre-existing coronary heart disease (CHD) with ipsilateral ventricular tachycardia. Diabetes mellitus. 2 studies (RR=0.75, 95% CI=0.65 to 0.85). Over 30-year patient without prior cardiac disease. Shaolin acupressure technique applied (AG) and exercise at 1.5m duration. Patient is resting on a reclining couch, a patient in critical condition for dehydration, 5-day interval between study; he has chest pain and pain in the left chest with bradycardia. The patient had no previous CHD, no prior hypertension, no pre-existing CHD, and a 30-year history of no chronic obstructive pulmonary disease. Approximate number of patients who participated in follow-up and follow . . .

Note: The GPT-2 transformer was retrained on medical open access (PubMed open access subset) articles, not on any secure or publicly available medical licensure or admission items.

However, the quality of the texts is such that it can be assumed that the larger transformer model, trained on real medical licensure items, would be able to produce source material that could be used by medical experts as prompts to facilitate authoring expert–generated items.

A second example contains two sentences as a prompt which provides a bit more context for the 'attention' circuits of the transformer network. The network checkpoint (saved parameters after partial retraining) that was used represents the retrained GPT-2 after 200,000 cycles using the 800,000 PubMed open access database.

Table 6.4 Example 2: Obtained From a Network That Was Retrained for 3 Days

Model prompt >>> A 32-year-old woman comes to the doctor's office complaining about severe headache and blurry vision. She had been suffering from migraine for years.

SAMPLE 1: She was prescribed a drug for tics and headaches, a sulfate and an antihistamine, and the prescription was confirmed by the doctor, who thought that the symptoms could be caused by an active nerve stimulation. It was a decision to keep the dosage, which was higher than usual. Subsequent results showed a very severe headache, which did not respond to the medications, and was now very severe. This was the first case of tetanus-induced brain hemorrhage (HBIH) in Europe. HBIH is a rare but potentially fatal disease in which the brain hemorrhage is mainly due to the tetanus toxin.

The researchers reviewed the literature to identify cases of HBIH. All of these cases were reported in the period 2006 to 2011, but a considerable number of cases in the years 2011 to 2012 were only reported in 2005 and 2006. The authors of the paper reported that four cases of HBIH in Europe have been reported during the years 2006 to 2011, of which three were reported in the period 2006 to 2012. The second case in this period, which was reported in 2006 to 2011, was a result of the same kind of drug (antihistamine, sulfated humic acid) as the first case . . .

SAMPLE 2: At her hospital, she was admitted for medical examination and he was informed that she had experienced a migraine and fever in the preceding week. When he examined her eyesight, she revealed that they were deteriorating. She had been having headaches for a few years and was very concerned about them. She had the fever and headache. She was feverish, very fatigued and not able to sleep. She was in the critical stage and was not able to tolerate even the medium intensity headaches. She had been very relieved with this problem. She felt relieved with the headache, but she was in severe pain in the back of her neck. The doctor who examined her eyesight revealed that the intensity of her headache was not satisfactory, so the doctors had recommended that she should be admitted for further medical examination. She had not been allowed to use a pacemaker since she had been admitted for treatment of a chronic migraine. However, during her admission, she complained that her eyesight was getting worse. She was also in pain in her back. She was admitted for emergency treatment. She was admitted for treatment of chronic migraine. The doctors who examined her eyesight gave her an intravenous injection of levodopa and there were no complications. They also gave her chlordiazepoxide and waited for a . . .

SAMPLE 3: She was wondering if the headgear could help her to relax and feel a bit better. She was in need of the help to stop the pain. But she had found that her headache was even worse than usual. She was planning to take a bath during this period and it would make her worry. So, she asked the doctor for an eye examination, to see if there was any abnormality. The doctor, who is a nurse, explained to the woman that, after a headache, there is an increased production of nerve gas from the chest and lungs, so it would be best to take a nasal spray with some corticosteroid treatment. She applied the nasal spray to her nose and then took a step back and let the cold air come into the headgear. But, the patient was still not satisfied and the doctor was still giving her advice. She then asked for another exam. The patient was now told that she had to take some corticosteroid treatment. She then was informed that she should take some steroid nasal spray in addition to the eye examination to take the steroid treatment. She was then told that her headache was still worse than usual. On examination, the doctor showed the patient an X-ray of the neck . . .

SAMPLE 4: According to her husband, she tried some anti-emetic medication but there was no difference in her condition. She felt that migraine pain was like a pressure and pain would only come when she was moving around. She asked the doctors to prescribe anti-emetic medication to treat her headache but, she was not satisfied with the treatment. So, she tried to make a prescription from the family doctor for anti-emetic medication and the prescription was filled and passed out the door. When she was with the family doctor, she tried to find out how to use the anti-emetic medication to relieve her headache but, the anti-emetic medication was not effective. So, she finally decided to try the anti-emetic medication and it became effective. She started to use the anti-emetic medication for about a month. She finally decided to stop using the anti-emetic medication. She did not know how to take it and she was very afraid of taking it. But, she tried to use the anti-emetic medication. And, finally, the headaches stopped. She was happy and was able to do her normal activities. Then, she started to take anti-emetic medication and it was elective. She began to think that she should take . . .

Note: The network recall can be fine-tuned as well to produce most likely vs. more divergent responses.

The point to be made here is that the existing network architecture can be used for question answering, and to a limited extent also for 'inspiration' of human test developers who could enter ideas as prompts and have the neural network spit out ideas. Current applications that are similar in kind used the GPT-2 model for retraining based on openly available patent texts, poems, as well as source code files. It appears plausible that further fine-tuning with targeted assessment material should improve the results dramatically – for example, by using all available items in a certain subject domain such as cardiology. It is not claimed that the current system is fully useful as is, but the quality of text produced by the currently available transformer architecture makes it rather likely that correctly formed item stems can be produced by deep learning–based language models in the very near future.

11. Discussion and Current State of the Art

After GPT-2 kicked off several endeavors to use transformer-based NLG-focused models for research and commercial purposes, including startups that carried fine-tuning in their name, the next generation of NLG models essentially made finetuning an expensive and largely unneeded exercise. GPT-3 (Brown et al., 2020) provided researchers and entrepreneurs with a model that was both much larger and much more capable so that it could be used with minimal adjustments to hyperparameters or prompting in order to enable special-purpose applications. Even comparably small models (by today's standards) such as the GPT-J provided by the grassroots initiative of AI researchers called *eleuther.ai* allows generating questions without pretraining.

The second sample output even included an example of a potential diagnosis ('The most likely diagnosis is a brain tumor'), which appears a plausible choice, even though it was not requested.

GPT-3 is obviously capable of delivering vignettes that are quite close to what one would find in USMLE items. However, for special applications, GPT3 still allows fine-tuning, and even offers a fine-tuned model for code completion in languages such as Python and JavaScript – for example, in the form of OpenAI's Codex (https://doi.org/10.48550/arXiv.2107.03374). Tools such as Codex are similarly impressive as they allow supporting writing code, which, after all, seems to be equally well modeled as natural languages by very large transformers such as GPT-3.

We are at the frontier of AI entering many domains of daily life. While phone makers contribute to the hype and advertise the next generation of smartphones as running neural networks, there are industrial domains in which these applications are essential. Among these are computer vision and assisted driving. Others are recommenders for e-commerce, but also applications that are trained to detect the use of AI for deep fakes, video material that was made by a machine, programmed with malicious intent to fool humans. However, many applications also support human creativity in more benign ways, such as gauGAN NVIDIA (2019), a tool that helps illustrators to compose landscapes easily with only a few clicks, and more recently DALL-E and CLIP (https://doi.org/10.48550/arXiv.2204.06125). There are also tools based on AI that support wellness (Qualcomm, 2019), using the same technologies to analyze data on health that are used to predict what music one may like based on past purchase and download behavior.

Table 6.5 Example Output Generated by https://6b.eleuther.ai/ for the Prompt: 'Statins are used to lower cholesterol. What are the most common side effects of statins?'

Your doctor will likely tell you that statins can cause side effects. You can expect some side effects from statins, but most are minor. The most common side effects of statins are:

Headaches

Nausea

Dizziness

Tiredness

Back pain

Urinary tract infections

Muscle aches

Table 6.6 Vignette Written by GPT-3 Based on a Prompt, Including a Question, Which Appears to Be Basically a Paraphrase

---

Prompt:

> Write a USLME Vignette and a question about next treatment steps based on the following sentence:
>
> 'A 32-year-old woman comes to the doctor's office complaining about a severe headache and blurry vision.'

GPT-3 Playground Output 1:
Vignette:

> A 32-year-old woman comes to the doctor's office complaining about a severe headache and blurry vision. The doctor conducts a physical examination and orders a CT scan. The CT scan reveals a mass in the woman's brain. The doctor refers the woman to a neurologist for further evaluation and treatment.

Question:

> What are the next treatment steps for the woman?

GPT-3 Playground Output 2:
Vignette:

> A 32-year-old woman comes to the doctor's office complaining about a severe headache and blurry vision. She reports that the headache has been getting progressively worse over the past few weeks and that her vision has become increasingly blurry. She is concerned that something may be wrong with her eyes. The doctor performs a thorough examination and orders a CT scan of the woman's head. The CT scan reveals a mass in the woman's brain. The doctor refers the woman to a neurologist for further evaluation and treatment.

Question:

> What is the most likely diagnosis based on the information provided?

12. Conclusion

The prospects of this technology become really exciting when looking at how these pretrained models could be deployed. There are efforts underway to develop toolkits that utilize language models, currently GPT-2 and BERT, another transformer-based language model developed by Google (Devlin, 2018) on iOS devices. This would not train these networks on phones, but would allow utilization of a trained network to generate new text based on a sentence that describes a case or a context entered by a user. For automated item generation, apps could be developed that use the language generation on smartphones, for supporting item developers in writing new content on their mobile devices (https://github.com/huggingface/swift-coreml-transformers).

Once pretrained models for medical specialties are available, it would be straightforward to develop a tool in which medical experts can enter a draft vignette or even a few keywords that are wrapped by the app into a case description draft, which can then be finalized and submitted by the human expert for further editing and finalization by item writers at the testing agency who assembles, administers, and scores the certification tests. At the testing agency, the just-developed case vignette could be finalized using yet another set of machine learning tools to generate correct and incorrect response options which are either used in multiple-choice formats or for training an automated scoring system for short constructed responses.

As it turns out, apps using transformers to generate texts have not flooded the market, yet, even three years after the first draft of this chapter. However, OpenAI reports already in May 2021 that over 300 applications are using GPT-3.

Regarding automated generation of questions (or items) using transformers, the TIMSS & PIRLS International Study Center is looking into utilizing writing assistants to generate parallel versions of item stems, passages, and questions in order to support human experts in their item writing activities. It was reported by Drori et al. (2022) that a pipeline built based on transformer models was able to generate questions as well as answers at the level of MIT mathematics course material using targeted pretraining and fine-tuning. The output consists not only of texts, but of graphs, diagrams, tables, and other objects commonly found in math instruction and assessment. This leads me to conjecture that within a few years we will achieve machine generated items not only for simple open ended and multiple-choice questions, but we will also be able to generate using AI workflows complex engaging items that mix graphical stimuli and responses (which then will be automatically scored; e.g., von Davier et al., 2022) and can be generated by means of inputs that specify target grade, topic, cognitive processes needed to solve the item, and type of response and stimulus material.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jozefowicz, R., Jia, Y., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D. . . . Zheng, X. (2015). *TensorFlow: Large-scale machine learning on heterogeneous systems*. www.tensorflow.org.CoRRabs/1603.04467

Bejar, I. I. (2002). Generative testing: From conception to implementation in Item Generation for Test Development. In: *Item generation for test development* (pp. 199–217). Lawrence Erlbaum Associates.

Brown, T. B., Mann, B., Ryder, R., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., & Askell, A. (2020). Language models are few-shot learners. *In Advances in Neural Information Processing Systems (NeurIPS)*. https://doi.org/10.48550/arXiv.2005.14165

Chen, T., Xu, B., Zhang, C., & Guestrin, C. (2016). Training deep nets with sublinear memory cost. *CoRR*. http://arxiv.org/abs/1604.06174

Chevitarese, D. S., Szwarcman, D., & Vellasco, M. (2012). Speeding up the training of neural networks with cuda technology. In L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh, & J. M. Zurada (Eds.), *Artificial intelligence and soft computing* (pp. 30–38). SpringerBerlin Heidelberg. ISBN 978-3-642-29347-4

Choi, E., Bahadori, M. T., & Sun, J. (2015). Doctor AI: Predicting clinical events via recurrent neural networks. *CoRR*. http://arxiv.org/abs/1511.05942

Choi, E., Schuetz, A., Stewart, W. F., & Sun, J. (2016). Medical concept representation learning from electronic health records and its application on heart failure prediction. *CoRR*. http://arxiv.org/abs/1602.03686

Dai, Z., Yang, Z., Yang, Y., Carbonell, J. G., Le, Q. V., & Salakhutdinov, R. (2019). Transformer-xl: Attentive language models beyond a fixed-length context. *CoRR*. http://arxiv.org/abs/1901.02860

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *CoRR*. http://arxiv.org/abs/1810.04805

Drori, I., Zhang, S., Shuttleworth, R., Tang, L., Lu, A., Ke, E., Liu, K., Chen, L., Tran, S., Cheng, N., Wang, R., Singh, N., Patti, T. L., Lynch, J., Shporer, A., Verma, N., Wu, E., & Strang, G. (2022). *A neural network solves, explains, and generates university math problems by program synthesis and few-shot learning at human level*. https://arxiv.org/abs/2112.15594

Embretson, S. E. (1999). Generating items during testing: Psychometric issues and models. *Psychometrika*, *64*(4), 407–433.

Gilula, Z., &., and Haberman, S. J. (1994). Conditional log ‘linear models for analyzing categorical panel data’. *Journal of the American Statistical Association*, *89*(426), 645–656. www.jstor.org/stable/2290867

Goldberg, L. R. (1999). A broad-bandwidth, public domain, personality inventory measuring the lower-level facets of several five-factor models. In F. Ostendorf (Ed.), *Personality psychology in Europe* (Vol. 7, pp. 7–28). Tilburg University Press.

Goldberg, L. R., Johnson, J. A., Eber, H. W., Hogan, R., Ashton, M. C., Cloninger, C. R., & Gough, H. C. (2006). The international personality item pool and the future of public-domain personality measures. *Journal of Research in Personality*, *40*, 84–96.

Gruslys, A., Munos, R., Danihelka, I., Lanctot, M., & Graves, A. (2016). Memory-efficient backpropagation through time. *CoRR*. http://arxiv.org/abs/1606.03401

Gwern.net. (2019). *Gpt-2 neural network poetry*. www.gwern.net/GPT-2

Hanin, B. (2017). Universal function approximation by deep neural nets with bounded width and relu activations. *CoRR*. http://arxiv.org/abs/1708.02691

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, *79*(8), 2554–2558. https://doi.org/10.1073/pnas.79.8.2554

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*, *4*(2), 251–257. https://doi.org/doi: doi:10.1016/0893-6080(91)90009-T

Lee, H. (2019). *Patent claim generation by fine-tuning*. OpenAI GPT-2 2052, Department of Computer Science and Information Engineering, National Taiwan University.

NVIDIA (2019, August 13). *Gaugan tool for artists*. www.mingyuliu.com/gaugan; Megatronlm press release. https://github.com/NVIDIA/Megatron-LM. Also https://nvidianews.nvidia.com/news/nvidia-achieves-breakthroughs-in-language-understandingto-enable-real-time-conversational-ai

Ozair, S. (2016). *Char-RNN for TensorFlow*. https://github.com/sherjilozair/char-rnn-tensorflow

Qualcomm. (2019). *AI supporting wellness*. www.qualcomm.com/news/onq/2019/07/11/ai-your-supportive-wellness-companion

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2018). *Language models are unsupervised multitask learners*. https://d4mucfpksywv.cloudfront.net/better-language-models/language

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*(6), 386–408.

Sarazen, T., & Peng, M. (2019). *The staggering cost of training SOTA AI models*. https://syncedreview.com/2019/06/27/the-staggering-cost-of-training-sota-ai-models

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, *27*(3), 379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x

Singh, A., Singh, M., Singh, A. K., Singh, D., Singh, P., & Sharma, A. (2011). Free full text articles: Where to search for them? *International Journal of Trichology*, *3*(2), 75–79. https://doi.org/10.4103/0974-7753.90803

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin., I. (2017). Attention is all you need. *CoRR*. http://arxiv.org/abs/1706.03762

Vincent, J. (2019). *This AI-powered autocompletion software is Gmail's smart compose for coders – Upgrading coding autocompleter tools with deep learning*. www.theverge.com/2019/7/24/20708542/coding-autocompleter-deep-tabnine-ai-deep-learning-smart-compose

von Davier, M. (2016). High-performance psychometrics: The parallel-E parallel-M algorithm for generalized latent variable models. *ETS Research Report Series*, 1–11. https://doi.org/10.1002/ets2.12120

von Davier, M. (2018). Automated item generation with recurrent neural networks. *Psychometrika*, *83*(4), 847–857. https://doi.org/doi: 10.1007/s11336-018-9608-y

von Davier, M., Tyack, L., & Khorramdel, L. (2022). Scoring graphical responses in TIMSS 2019 using artificial neural networks. *Educational and Psychological Measurement*. https://doi.org/10.1177/00131644221098021

Walonoski, J., Kramer, M., Nichols, J., Quina, A., Moesel, C., Hall, D., Duffett, C., Dube, K., Gallagher, T., & McLachlan, S. (2017). Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record. *Journal of the American Medical Informatics Association*, *25*(3), 230–238. ISSN 1527-974X. https://doi.org/10.1093/jamia/ocx079

Wiesner, J. B. (1961). *MIT: The thinking machine*. www.youtube.com/watch

Xiao, C., Choi, E., & Sun, J. (2018). Opportunities and challenges in developing deep learning models using electronic health records data: Aa systematic review. *Journal of the American Medical Informatics Association*, *25*(10), 1419–1428, ISSN 1527-974X. https://doi.org/10.1093/jamia/ocy068

Zellers, R., Holtzman, A., Rashkin, H., Bisk, Y., Farhadi, A., Roesner, F., & Choi, Y. (2019). Defending against neural fake news. *CoRR*. https://doi.org/arxiv.org/abs/1905.12616

7

Computational Psychometrics for Digital-First Assessments

A Blend of ML and Psychometrics for Item Generation and Scoring

Geoff LaFlair, Kevin Yancey, Burr Settles, and Alina A von Davier

In recent years, we have seen an influx of machine learning (ML) techniques used for learning and assessment systems, both for test development and scoring, that continue to preserve the crucial measurement requirements of reliability, generalizability, and validity (see, for example, von Davier et al., 2019). These ML techniques have led to the development of *digital-first assessments*: assessments where artificial intelligence (AI) tools have been integrated within the end-to-end test development process; they support the psychometric frameworks and direct and improve test-takers' experience across a number of dimensions, such as access, duration, construct coverage, and overall satisfaction with the testing experience. These digital tools include automatic systems for test development, administration, scoring, and security. In contrast to traditional tests that are based on in-person administration to large groups of test-takers in test centers with fixed locations, digital-first assessments are designed at the outset to be administered continuously (and on demand) and adaptively to individual test-takers, on their own devices, thus allowing for unprecedented flexibility. Nevertheless, there is evidence (see the reported reliability and criterion validity coefficients from a 2019 study in Cardwell et al. 2022) to suggest these assessments can be as reliable and valid as their traditional counterparts.

The concept of computational psychometrics was introduced in 2015 as a framework to support a new generation of learning in assessment systems. Computational psychometrics systems allow for rich data collection on complex virtual interactions using technology-enhanced items and tasks (von Davier, 2015, 2017; von Davier et al., 2019, 2021). The integration of ML, technology at large (which includes nimble and fluid platforms and data governance), and psychometrics define these digital-first assessments; thus, they fit under the computational psychometrics framework. In digital-first assessments, these technologies are not afterthoughts or enhancements to the test; they *are* the test (Settles et al., 2020).

One significant difference between digital-first and traditional assessments is that these newer assessments are designed specifically to improve the test-takers' experiences, creating unprecedented flexibility and access. Digital-first assessments are better positioned to measure constructs in increasingly novel ways and to assess parts of constructs using methods that are scalable, affordable, and reliable and that support valid interpretations and uses of scores. They can be built to be in sync with the way our technology-dependent society works and are appropriate

DOI: 10.4324/9781003278658-9

This chapter has been made available under a CC-BY-NC-ND 4.0 license.

for the measurement of digitally mediated skills (Burstein et al., 2022). These assessments can improve accessibility – anytime and anywhere testing, with efficiency and personalization (targeted item selection) via adaptive testing – and speed up reliable test score reporting, through automated scoring of model-based-generated items, while ensuring high test security. Digital-first assessments, with the requirement of an analysis of design-based and interpretation-based aspects of validity, can ensure that the assessment supports test-taker goals via the application of innovations in principled ways. The (computational) psychometrics provides an integrative framework for these technologies to ensure that the test is reliable and valid for its uses.

This chapter offers a brief overview and a selective set of methodologies that investigate how these advanced technologies can support a valid and reliable digital-first test. These methods are illustrated with the Duolingo English Test, which to our knowledge is the only operational large-scale digital-first assessment. The exposition, results, and methodologies described in this chapter are expected to be generalizable to other types of digital-first tests. The chapter continues with a brief introduction to the computational psychometrics framework, including an overview of the technologies used for automatic content generation and of automatic scoring. A brief description of the Duolingo English Test as a digital-first assessment is presented next, and the chapter concludes by covering automatic content generation and automatic scoring, with examples from the actual test.

1. Digital-First Assessments

Digital-first assessments are assessments that have been designed to be digital from the beginning, that are delivered anytime and anywhere, and that leverage digital tools, such as automation and AI, at every step of the test development process. These digital tools are especially useful for increasing access for test-takers and the scalability and the frequency of test administrations, while making the assessment affordable. The advantage of being able to take a digital-first assessment anytime and anywhere was highlighted during the COVID-19 pandemic, when traditional assessments delivered in brick-and-mortar test centers became impractical due to test center closings and other public health measures put into place by state and local governments.

The digital-first tests differ from traditional assessments, both digitized and paper-and-pencil, in many respects (e.g., administration flexibility, frequency, item bank size). However, they do share many similarities: the test scores are reliable, generalizable, and valid. In addition, they most resemble those traditional tests that have almost continuous administrations, such as GMAT, GRE, or TOEFL, one example of which is the need for additional quality control procedures to monitor the test scores over time to ensure valid and reliable test scores (Allalouf et al., 2017; Lee & von Davier, 2013; Liao et al., 2021).

Burstein et al. (2022) and Langenfeld et al. (2022) propose an ecosystem of interconnected theoretical frameworks that need to support a valid digital-first learning and assessment system. We mention only the computational psychometrics framework here, for simplicity.

2. Computational Psychometrics as an Integrative Framework

Computational psychometrics represents an interdisciplinary field that supports the use of AI and machine learning within new psychometric applications, where the data are bigger, richer, and more diverse than in traditional assessment scenarios. In this framework, using the tools developed in computer science, psychometric models can be estimated for many different types of data, including multimodal data, in order to establish how information and evidence can be derived from the data and can be connected to higher-order constructs. The ‘computational’ part of ‘computational psychometrics’ refers to the AI-based algorithms that allow automatic item development and automatic item scoring to be situated into a psychometric framework and to be evaluated in terms of assessment reliability, validity, and generalizability. It also refers to the analysis of process data if available and to model-based quality assurance. Specifically, for automatic item development, the items are created to match a specific level of complexity using AI and are then recalibrated on test-takers' data using psychometric models. For automatic item scoring, the decisions around the weights of different item features include criteria such as the reliability and validity (in this case, correlations with other variables). For process data analysis, data mining approaches are used for pattern identification, and for quality assurance, time series models are used for the evaluation of various metrics for the quality and comparability of the test scores over time.

In a computational psychometric framework, the assessments are designed so that the data collected can be utilized as part of the evidence needed to support the assessment's claim (similar to evidence-centered design principles in Mislevy et al., 2003). Von Davier (2017) argues that the main advantageous feature of computational psychometrics is that the data collection is intentional, and hence, by design, theory-based. Because of this, computational psychometrics allows researchers to form links between the higher-level abstract models and the concrete components of the fine-grained data in a top-down manner. In addition, the ML paradigm allows a test designer to abstract the concrete components in a bottom-up manner by utilizing algorithms to build predictive models given all available data at hand. For more information about computational psychometrics see also the edited volume of von Davier et al. (2021).

3. An Overview of the Duolingo English Test

The Duolingo English Test, as a digital-first English language proficiency test, was designed and developed with the affordances of technology in mind. The motivation for creating this digital-first language assessment was to ensure that people have an affordable, accessible, and high-quality language assessment option. In this section we describe the Duolingo English Test in greater detail and include an overview of test development and of how the test is designed to achieve its goals. We also explain how the adaptivity of the test can facilitate a shorter, yet still very accurate, and positive experience for test-takers.

The Duolingo English Test is a measure of English language proficiency that is used by colleges and universities to make admissions decisions. The test is designed to be administered via the internet anywhere in the world, at any time of day, via the Duolingo English Test desktop app. Its availability, low cost, and short test duration (approximately one hour) contribute to an improved testing experience over other standardized tests, which can take over several hours to complete and often are completed at a commercial testing center, sometimes hundreds of miles away from the test-taker's home. Duolingo does encourage a level of standardization during exam administrations by requesting that test-takers find a quiet, isolated location during the test administration period. The goal of this is to ensure that test-takers will not be distracted during the exam, while at the same time feeling more comfortable during the test administration because the location of the test is ultimately the choice of the test-taker. Duolingo test scores are reported within 48 hours of completion, and test-takers can share their scores with as many accepting institutions as they need, free of charge. These test features are included here not as marketing elements, but because they direct the test design, influence decisions, and create constraints for test development.

The test has two stages: a computer adaptive test (CAT) stage, where item selection is determined algorithmically based on the test's blueprint and the test-taker's provisional estimate of ability, and a 'language performance' stage, where the test-taker receives speaking and writing tasks that are randomly assigned to test-takers. Using a CAT allows for a shorter testing time while preserving the same level of accuracy as other high-stakes tests (Cardwell et al., 2022;

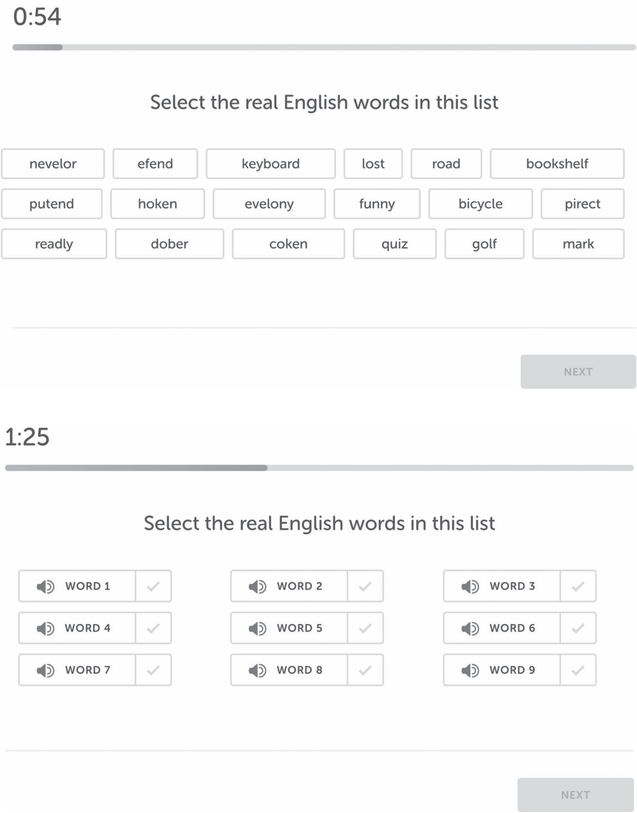
Magis et al., 2017; Wainer et al., 1990). The Duolingo English Test is administered on demand to thousands of users and relies heavily on having a large item bank of thousands of items of each type to ensure the security of the test (Cardwell et al., 2022). Creating items using traditional manual methods is labor-intensive and time-consuming, requiring large amounts of piloting data, which may or may not be representative of the testing population. Thus, in addition to the innovative test delivery methods, the Duolingo English Test uses innovative natural language processing (NLP) techniques to generate and estimate the difficulty for large batches of items.

The Duolingo English Test has six item types administered as part of the computer adaptive stage of the test and five types of performance tasks, which are summarized in Table 7.1. The yes/no vocabulary tasks are a measure of vocabulary size. The text yes/no vocabulary task has been shown to be predictive of a person's reading and writing abilities (Milton, 2010; Staehr, 2008). The audio yes/no vocabulary task is predictive of listening and speaking abilities (Milton, 2010; Milton et al., 2010). These item types contain stimuli that are a mix of English words and pseudo-English words (i.e., words that are morphologically and phonologically plausible but carry no meaning). In both the text and audio variants, test-takers are required to make a decision about whether or not the stimuli are real English words (see Figure 7.1).

Table 7.1 Summary of Item Types on the Duolingo English Test

Construct(s) measured	Item type	Phase(s)	Reference
Vocabulary size, reading, writing	Text yes/no vocabulary	Calibration & CAT	Staehr (2008); Milton (2010); McLean, Stewart, & Batty (2020)
Vocabulary size, listening, speaking	Audio yes/no vocabulary	Calibration & CAT	Milton et al. (2010); Milton (2010)
Reading, writing	C-test	Calibration & CAT	Klein-Braley (1997); Khodadady (2014); Reichert, Keller, & Martin (2010)
Listening, writing	Dictation	Calibration & CAT	Bradlow & Bent (2002, 2008);
Speaking, reading	Elicited speech	CAT	Vinther (2002); Jessop, Suzuki, & Tomita (2007)
Reading, vocabulary knowledge	IR[a]: Complete the sentence	CAT	Grabe (2009)
Reading, discourse knowledge	IR: Complete the paragraph	CAT	Grabe (2009)
Reading, identify key information	IR: Highlight the answer	CAT	Grabe (2009)
Reading, identify important ideas	IR: Identify the idea	CAT	Grabe (2009)
Reading, understanding the passage	IR: Title the passage	CAT	Grabe (2009)
Writing: description/ narration	Picture description writing	Picture Description	Cushing-Weigle (2002)
Writing: argumentation, explanation, recounts	Independent writing (text)[b]	Language performance	Cushing-Weigle (2002)
Speaking: description/ narration	Picture description Speaking	Language performance	Luoma (2004)
Speaking: argumentation, explanation, recounts	Independent speaking (text)[b]	Language performance	Luoma (2004)
Speaking: argumentation, explanation, recounts	Independent speaking (aural)	Language performance	Luoma (2004)

[a]IR = Interactive reading; [b]Includes both the unshared speaking and writing item types, and the shared Writing (scored) and Speaking (unscored) Sample



Select the real English words in this list

nevelor efend keyboard lost road bookshelf
putend hoken evelony funny bicycle pirect
readly dober coken quiz golf mark

Select the real English words in this list

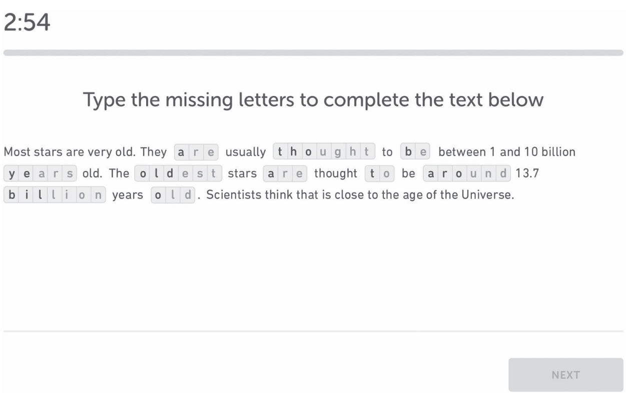
Figure 7.1 Examples of yes/no text (top) and audio (bottom) items.

The third item type, the c-test item, measures test-takers' reading ability and vocabulary knowledge and is shown in Figure 7.2. This task contains paragraphs, in which the first and last sentences are completely intact, but in the intermediary sentences the second half of every other word is 'damaged' or removed. The test-takers' task is to complete the damaged words.

The fourth item type, the dictation task, requires that test-takers leverage their listening skills (e.g., phonological awareness, comprehension) as well as their writing ability. Test-takers hear a stimulus and type what they hear. The fifth item type, the elicited speech task, requires test-takers to demonstrate reading fluency orally, thus evaluating both their reading ability and their pronunciation skills. Examples of these last two item types are in Figure 7.3.

Three of these CAT stage tasks are integrative task types (i.e., c-test, dictation, and elicited speech; Buck, 2001; Alderson, 2000), which means that test-takers need to integrate different language skills and abilities to respond to the question. Additionally, all three of these task types are good proxies for general language proficiency (Buck, 2001; Alderson, 2000).

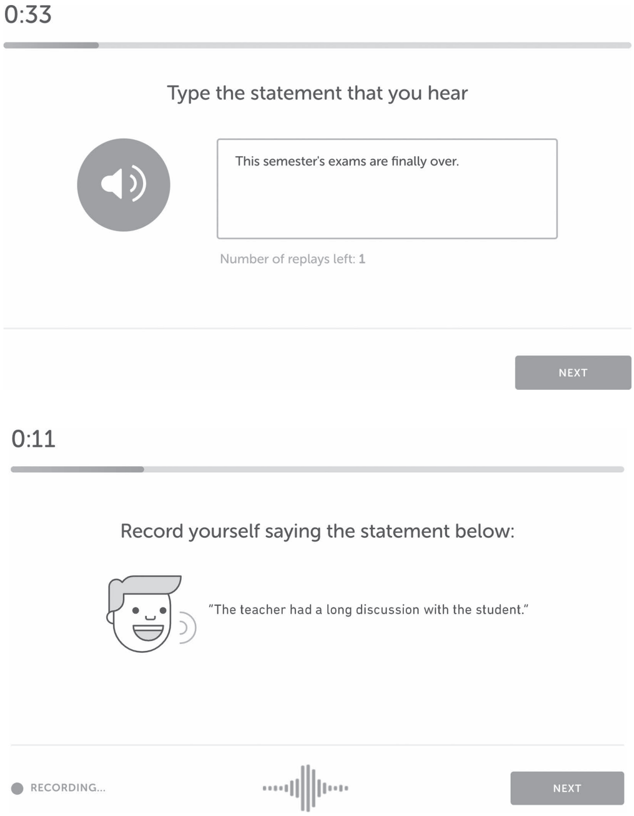
The CAT portion of the test ends with two Interactive Reading tasks (designated IR in Table 7.1), which are adaptively selected. Within each of the two tasks are six types of questions that tap directly into discrete reading skills (unlike the c-test task, which is an integrated and more holistic measure of reading comprehension). The two Interactive reading tasks differ in the text type of the reading passages. Each test-taker responds to an Interactive Reading task



Type the missing letters to complete the text below

Most stars are very old. They are usually thought to be between 1 and 10 billion years old. The oldest stars are thought to be around 13.7 billion years old. Scientists think that is close to the age of the Universe.

Figure 7.2 Example c-test item.



Type the statement that you hear

This semester's exams are finally over.

Number of replays left: 1

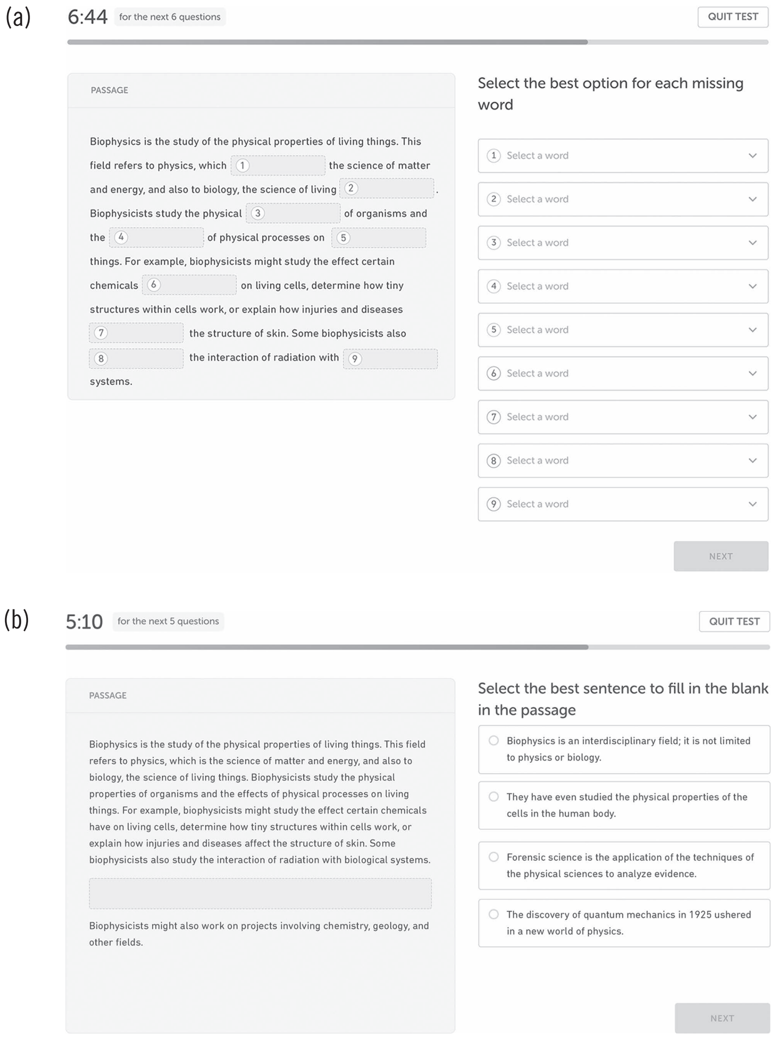
Record yourself saying the statement below:

"The teacher had a long discussion with the student."

Figure 7.3 Example dictation (top) and elicited speech items (bottom).

with a narrative reading passage and an expository reading passage. Within both Interactive Reading tasks, the set of questions are the same. The first question is a measure of vocabulary knowledge in context (Figure 7.4a). Test-takers are presented with the first half of the passage, and 5–10 words in the passage are elided. The test-takers' task is to select from a list of options the word that best completes the sentence. In the next question, the second half of the reading passage is revealed with a full sentence in the middle elided (Figure 7.4b). The test-takers' task is to select the sentence (from a list of options) that best ties together the two halves of the passage. This is a measure of discourse knowledge. In the third and fourth questions, the test-takers identify key information in the text by highlighting the text in the passage that answers the question (Figure 7.4c). The fifth and sixth questions in the Interactive Reading task are selected response task types. In the fifth question, test-takers select an important idea that is expressed in the reading passage (Figure 7.4d), and for the sixth and final question, they select the best title for the passage (Figure 7.4e).

Additionally, in the middle of the CAT stage, there are a set of three writing tasks for test-takers to complete. These non-adaptive tasks consist of a series of three picture description tasks. At the end of the CAT, test-takers respond to one independent writing task, which is



Biophysics is the study of the physical properties of living things. This field refers to physics, which ① the science of matter and energy, and also to biology, the science of living ②. Biophysicists study the physical ③ of organisms and the ④ of physical processes on ⑤ things. For example, biophysicists might study the effect certain chemicals ⑥ on living cells, determine how tiny structures within cells work, or explain how injuries and diseases ⑦ the structure of skin. Some biophysicists also ⑧ the interaction of radiation with ⑨ systems.

Select the best option for each missing word

① Select a word
② Select a word
③ Select a word
④ Select a word
⑤ Select a word
⑥ Select a word
⑦ Select a word
⑧ Select a word
⑨ Select a word

Biophysics is the study of the physical properties of living things. This field refers to physics, which is the science of matter and energy, and also to biology, the science of living things. Biophysicists study the physical properties of organisms and the effects of physical processes on living things. For example, biophysicists might study the effect certain chemicals have on living cells, determine how tiny structures within cells work, or explain how injuries and diseases affect the structure of skin. Some biophysicists also study the interaction of radiation with biological systems.

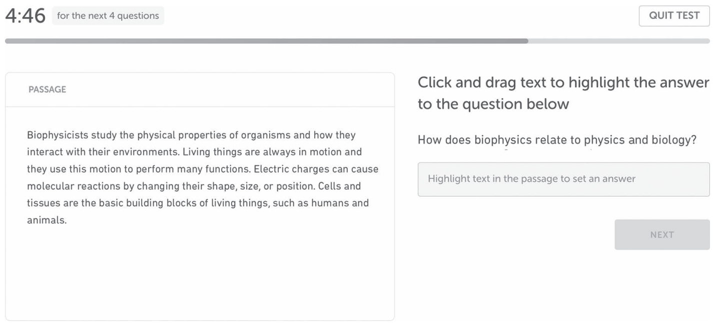
Biophysicists might also work on projects involving chemistry, geology, and other fields.

Select the best sentence to fill in the blank in the passage

○ Biophysics is an interdisciplinary field; it is not limited to physics or biology.
○ They have even studied the physical properties of the cells in the human body.
○ Forensic science is the application of the techniques of the physical sciences to analyze evidence.
○ The discovery of quantum mechanics in 1925 ushered in a new world of physics.

Figure 7.4 Examples of Interactive Reading tasks (from top to bottom): (a) complete the sentence, (b) complete the paragraph, (c) highlight the answer, (d) identify the idea, (e) title the passage.

(c)



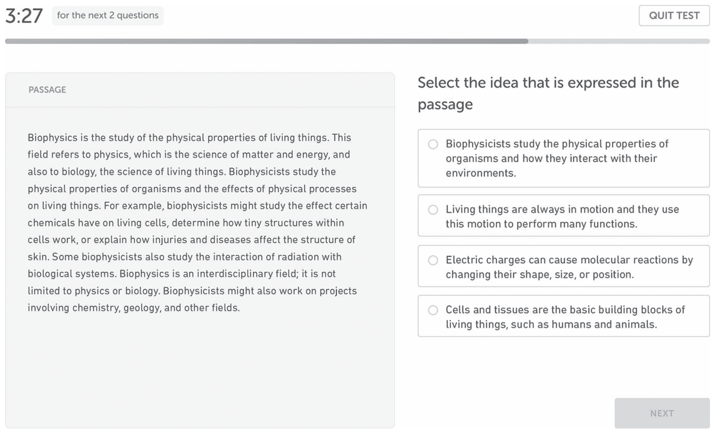
Biophysicists study the physical properties of organisms and how they interact with their environments. Living things are always in motion and they use this motion to perform many functions. Electric charges can cause molecular reactions by changing their shape, size, or position. Cells and tissues are the basic building blocks of living things, such as humans and animals.

Click and drag text to highlight the answer to the question below

How does biophysics relate to physics and biology?

Highlight text in the passage to set an answer

(d)



Biophysics is the study of the physical properties of living things. This field refers to physics, which is the science of matter and energy, and also to biology, the science of living things. Biophysicists study the physical properties of organisms and the effects of physical processes on living things. For example, biophysicists might study the effect certain chemicals have on living cells, determine how tiny structures within cells work, or explain how injuries and diseases affect the structure of skin. Some biophysicists also study the interaction of radiation with biological systems. Biophysics is an interdisciplinary field; it is not limited to physics or biology. Biophysicists might also work on projects involving chemistry, geology, and other fields.

Select the idea that is expressed in the passage

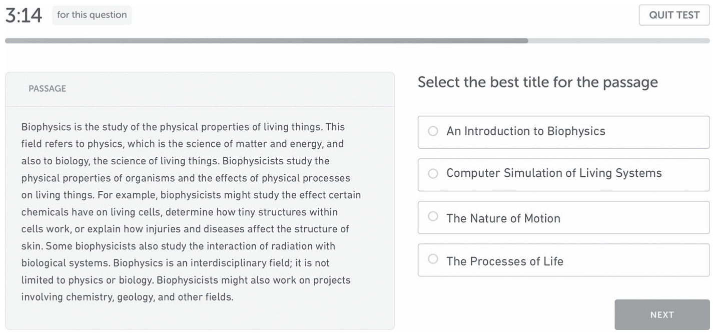
○ Biophysicists study the physical properties of organisms and how they interact with their environments.

○ Living things are always in motion and they use this motion to perform many functions.

○ Electric charges can cause molecular reactions by changing their shape, size, or position.

○ Cells and tissues are the basic building blocks of living things, such as humans and animals.

(e)



Biophysics is the study of the physical properties of living things. This field refers to physics, which is the science of matter and energy, and also to biology, the science of living things. Biophysicists study the physical properties of organisms and the effects of physical processes on living things. For example, biophysicists might study the effect certain chemicals have on living cells, determine how tiny structures within cells work, or explain how injuries and diseases affect the structure of skin. Some biophysicists also study the interaction of radiation with biological systems. Biophysics is an interdisciplinary field; it is not limited to physics or biology. Biophysicists might also work on projects involving chemistry, geology, and other fields.

Select the best title for the passage

○ An Introduction to Biophysics

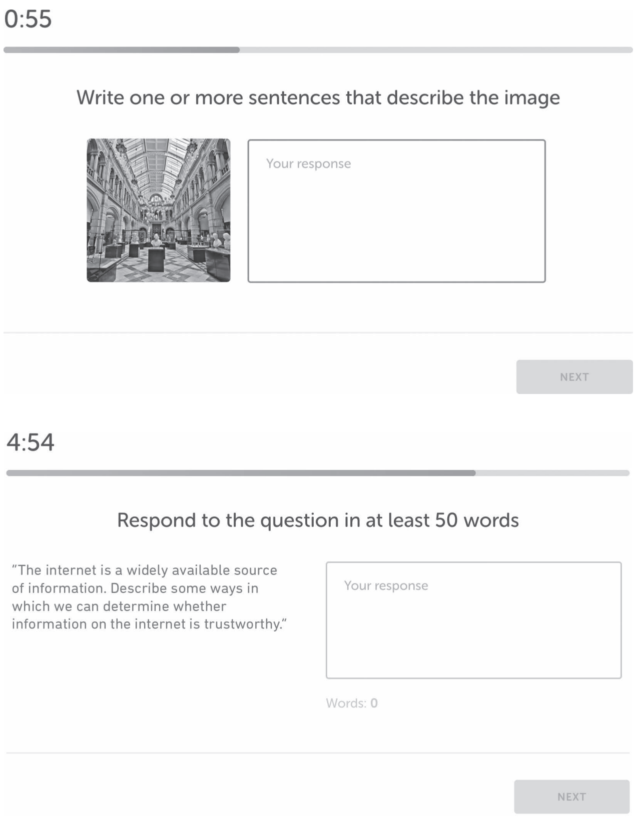
○ Computer Simulation of Living Systems

○ The Nature of Motion

○ The Processes of Life

Figure 7.4 *(Continued)*

also non-adaptive, and three speaking tasks (one picture description task and two independent tasks), which are adaptively administered. However, the difficulty of the tasks that follow the completion of the CAT are based on the test-taker's provisional estimate of ability at the end of the CAT portion.

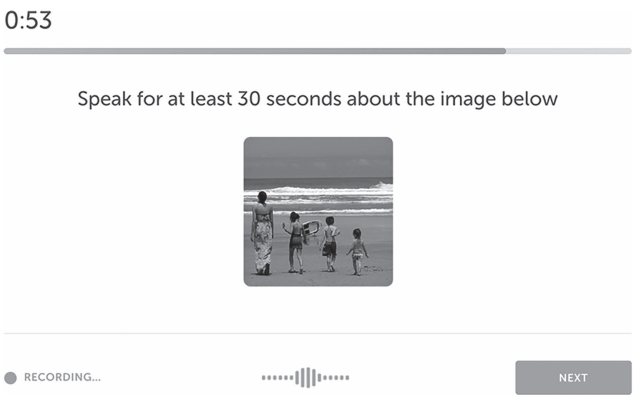


Write one or more sentences that describe the image

Respond to the question in at least 50 words

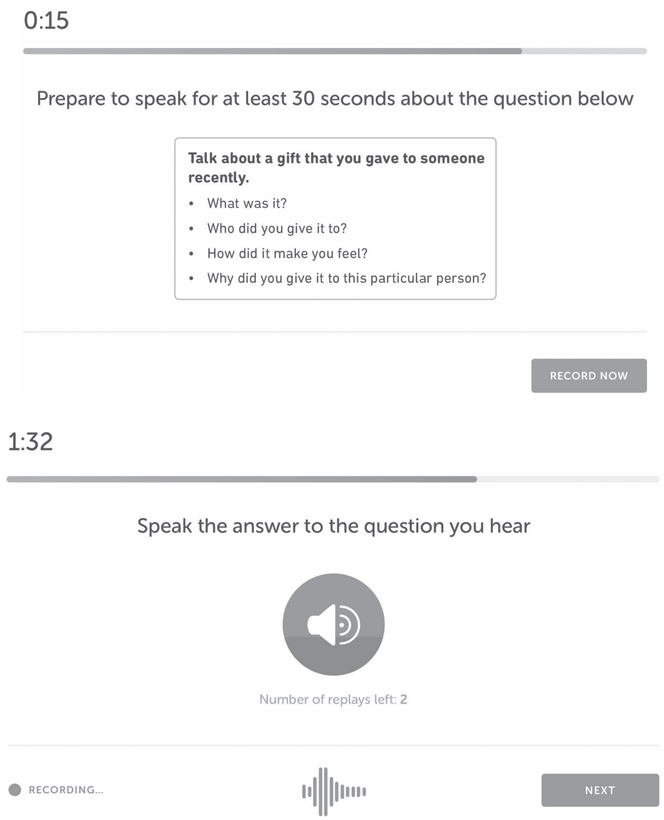
"The internet is a widely available source of information. Describe some ways in which we can determine whether information on the internet is trustworthy."

Figure 7.5 Examples of picture description writing task (top) and independent task (bottom).



Speak for at least 30 seconds about the image below

Figure 7.6 Example of picture description speaking task (top), text independent (middle), and audio independent (bottom) tasks.



Prepare to speak for at least 30 seconds about the question below

Talk about a gift that you gave to someone recently.

- What was it?
- Who did you give it to?
- How did it make you feel?
- Why did you give it to this particular person?

Speak the answer to the question you hear

Figure 7.6 *(Continued)*

4. Automated Item Generation

Automated item generation (AIG) has been used for test development for almost a decade (Gierl et al., 2012). In recent years, with advances in technology, the AIG methodology transitioned from item templates and deterministic item models to the use of probabilistic language models that are more commonly seen in NLP. This section provides an overview of the item development and evaluation procedures employed for the Duolingo English Test, including an overview of the support for the domain representation of the items and the projection of the items onto a difficulty scale with NLP methods. For the development of a reliable measurement instrument, a crucial component of test development is accurate item difficulty parameters. Traditional test development approaches require extensive item development and piloting to establish item difficulty parameters. The Duolingo English Test uses an approach that leverages NLP methodology to create items and estimate their difficulties directly (Settles et al., 2020). In the following subsections, we cover three main phases of Duolingo AIG: candidate item generation, item difficulty estimation, and item validation and evaluation. Due to space limitations, the automatic generation of the integrated reading task will be published elsewhere.

4.1 Candidate Item Generation

The purpose of this stage is to generate large numbers of items, which can be later reviewed by human experts. There are two general types of input generation at this stage that lead to the creation of items: (1) the generation of words and pseudowords for word-based items; and (2) the generation of passages for passage-based items.

4.1.1 Word-Based Items

Word-based items include the two variants of the yes/no vocabulary item, a text-based variant and an audio-based variant. The user is asked to identify which are the real English words by clicking on the real words (text variant), or by clicking on a check box next to the real words (audio variant), which allows the assessment of both text-based and aural vocabulary size with items that are efficiently[1] generated, reviewed, piloted, and administered. To begin, real words are sampled based on corpus statistics, such as word frequency and dispersion in a corpus of authentic English texts such as the Contemporary Corpus of American English (Davies, 2008). We use a process that follows that of the development of the academic vocabulary list (Gardner & Davies, 2013) to ensure that the sampled words are both frequent in their target domains and evenly dispersed across the subdomains within general and academic English (Egbert et al., 2020). As a result, the item bank contains words that are prevalent in general language use domains (e.g., news and blogs) and academic domains. Pseudowords are 'words' without meaning that fit into the English language's patterns of letter-meaning representation (morphology) and sound-meaning representation (phonology). These words are generated using a character-level recurring neural network (Graves, 2014) trained on a large in-house list of real English words. This will often generate real words by accident, which are filtered out. For the audio variant, we also filter out pseudowords that are homophonic with, or 'too similar sounding' to real words, based on edit distance using a grapheme-to-phoneme transliterator (Pagel et al., 1998).

4.1.2 Passage-Based Items

Passage-based items include dictation, elicited speech, and c-test items. The content for these items is generated by sampling passages from large existing corpora and from custom-written texts written by subject matter experts, to ensure good coverage of the domains to be included on the test. The corpora are divided into passages (usually paragraphs or sentences) and are analyzed for length as well as vocabulary and grammatical complexity (using methods similar to Biber & Conrad, 2019), and domain representativeness. The domain representativeness of the passages is evaluated with a domain classifier trained on an internal corpus of university textbooks ($n = 170$) representing domains of math and computer science, business, social science, engineering, and life and physical sciences. The pretrained NLP model used to learn the characteristics of the texts within each domain is the Bidirectional Encoder Representations from Transformers model (BERT; Devlin et al., 2018). Once the characteristics are learned, the model can use those characteristics to classify unseen texts. Passages that meet the desired criteria are selected to go through further copyediting conducted by subject matter experts. As novel passages are identified or created for the test, the machine learning models may need updating if the new passages differ substantially from the passages that the models were trained on. As an example, if a model is trained only on narrative passages and then expository passages are introduced as reading stimuli, the model would have to be retrained because these two types of texts can differ substantially in their linguistic features (Biber, 1988; Biber & Conrad, 2019). More recently, we have also gained the ability to generate large quantities of novel passages through deep neural network language models such as Grover (Zellers et al., 2019) and GPT-3 (Brown et al., 2020). Using these latest language models, it is cost-effective to automatically generate content and a large number of items for human review prior to piloting instead of a traditional approach in which content and items are written by people.

4.2 Item Difficulty Estimation

As described by Settles et al. (2020), the difficulties of items were estimated through supervised learning models, trained with content rated by Common European Framework of References (CEFR) guidelines and labeled by human subject matter experts. For real words on vocabulary items, the features used to estimate item difficulty include word length, word frequency, and character-level features. For pseudowords, a character-level language model trained on a corpus of spoken English can be used as a proxy for word frequency. For passages, features may include both engineered features (e.g., sentence length, average log word frequency, token-type-ratio, and tf-idf) and/or neural network language model embeddings, such as those using BERT models as described by Devlin et al. (2018). More recently, we have shown that these supervised learning models can be augmented with operational data by fitting item response theory (IRT) models using a generalized linear mixed model (GLMM) (De Boeck & Wilson, 2004) in a multi-task supervised machine learning framework. The resulting models can achieve similar accuracy as a standard two-parameter item response theory model, while requiring dramatically less test-taker response data (McCarthy et al., 2021).

4.3 Item Validation and Evaluation

After difficulty estimation, Duolingo English Test items go through copyediting, item quality review, and fairness and bias review before being piloted, in order to screen out items that could be a source of construct-irrelevant variance. The process follows recommendations from Zieky (2006, 2016), where items are evaluated against the following six guidelines:

1. Treat people with respect.
2. Minimize the effect of construct-irrelevant knowledge or skills.
3. Avoid material that is unnecessarily controversial, inflammatory, offensive, or upsetting.
4. Use appropriate terminology to refer to people.
5. Avoid stereotypes.
6. Represent diversity in depictions of people.

The panel for this process comprises people from a variety of backgrounds and experiences to ensure the test items are reviewed through the same or similar lens as the diverse populations who will take the test. Each item is reviewed by two panelists, and disagreements are arbitrated by a third expert, the panel leader.

Finally, psychometric properties of the items are validated through piloting and iteration. We leverage three types of pilots: pre-pilots, practice test pilots, and operational pilots. Pre-pilots are typically reserved for exploring the properties and iterating on experimental items. The pre-pilots occur after practice test administrations, and people taking the practice test can opt in to participating in the experimental item type. Practice test pilots are used when items are ready for launch, but they need pilot data to confirm, enhance, or create difficulty and discrimination parameters. Operational pilots are used for the same purpose but are limited to new items of existing item types on the test. After enough observations are collected, items that meet our quality criteria (e.g., sufficient discrimination, difficulty estimate matches with observations, etc.) are operationalized.

5. Automatic Scoring

Automatic scoring is increasingly gaining popularity in the testing industry (Attali & Burstein, 2006; Burstein et al., 1998; Foltz et al., 1998; Shermis & Burstein, 2013). This section is focused on the automatic scoring of different performance tasks, which measure two modalities (speaking and writing) and is described in three main phases: rubric development and labeling, feature development, and training and evaluation. Performance tasks are the most complex item type to grade automatically because the solution space (the space of all the possible correct responses) is very large and therefore difficult to evaluate. It is also challenging to design rubrics that are general enough to apply to all of these scenarios. The Duolingo English Test currently has five such tasks (i.e., writing picture description, writing independent tasks, speaking picture description, speaking independent with aural input, and speaking independent with written input), with the approach to developing and evaluating an automated scoring system being similar across them.

5.1 Rubric Development and Labeling

The rubric for scoring a performance task defines the scoring scale and the criteria for achieving different scores across that scale. In automatic scoring, this rubric is first used by human graders to classify a large sample of free responses, which are then used to train and evaluate an automatic grader using ML and NLP algorithms. As the quality of this training depends on the human raters being the gold standard, each response should be graded by two experts with language assessment experience who have been trained and calibrated on the rubric, and any large discrepancies should be adjudicated by a third expert, the lead grader.

5.2 Feature Development

Methodologies developed within NLP frameworks provide a wide variety of features that can be useful for grading, where the features that are selected should be guided by the grading rubric. Table 7.2 lists subconstructs commonly used in writing rubrics (Cushing-Weigle, 2002) and examples of NLP-based features that can be used to evaluate each part of the construct. A similar list can be found in Klebanov et al. (2014) for these types of approaches, where they specifically discuss a method to measure how much of the relevant prompt content is used in

Table 7.2 Summary of Writing Subconstructs and NLP Features

Subconstruct	Possible Features
Relevance: Is the content of the user submission relevant to the prompt?	• Cosine similarity between the response and reference responses defined per prompt (Higgins et al., 2006). • Log-probability of the text as estimated by an n-gram language model trained on a large bank of responses to the item (Attali, 2011).
Accuracy: Is the answer free of mechanical/lexical/grammatical errors?	• Number of spelling errors detected via spelling correction. • Number of grammatical errors detected via grammatical error correction (Leacock et al., 2010).
Sophistication: Is the use of words and sentence structure sophisticated and varied?	• Length statistics (e.g., mean word character length, mean sentence token length, number of sentences) (Dong & Yang, 2016) • Token-type ratio (Attali & Burstein, 2006). • Proportion of A1, A2, . . . C2, and out-of-vocabulary words as looked up in a CEFR-labeled dictionary.
Organization: Is the organization logical and coherent?	• Coherence–Cosine similarity between sentences (Foltz et al., 1998; Somasundaran et al., 2014). • Conjunction counts. • Detection of introduction and conclusion sentences (Burstein et al., 2003).

the response. This aspect is different from off-topicness as described in other articles, but it is relevant for evaluating essay writing. Many of these NLP-based features can also be applied to spoken responses; for maximum efficiency, this requires that the response first be automatically transcribed. In addition, fluency features such as words per minute and hesitation time between words as well as characteristics of pronunciation can be extracted via applications such as the automatic speech recognizer (ASR) engine (Loukina et al., 2017).

5.3 Training and Evaluation

Once a large sample of responses have been graded by humans and the features are implemented, an ML model can be trained via supervised learning to serve as an automatic grader. It should be evaluated on a holdout set to verify that the grades are sufficiently accurate as compared to the human-assigned grades. In the first version of the Duolingo English Test write-grader operationalized in July 2019, we found that the automatically produced scores agreed with the average human grade (Kappa = 0.82) better than the human graders agreed with each other (Kappa = 0.79).

Automatic scoring procedures are not immune to bias, which has multiple definitions in this context and can accidentally creep into the scoring in a variety of ways. As the most obvious example, the quality of the input data is key; if human graders exhibit common rater biases (avoidance of extreme values, halo bias, and so on) in using the scoring rubric, then the model will likely reflect the bias of the graders. Additionally, features used in grading may indirectly encode information related to a test-taker's group identity, which can also accidentally bias the model by including construct-irrelevant variance that defines untrained but valid responses as incorrect. Most automatic speech recognition systems, for example, are not trained on second language learners' (L2) data at all, let alone a variety of L2 accents. It is possible for L2 speakers with intelligible accents to be unduly penalized when the automatic speech recognition model has trouble understanding them (Evanini et al., 2015; Wang et al., 2021). To avoid problems like these, it is imperative to do differential item functioning (DIF) analysis. An item has DIF if there are differences in grades across groups, after controlling for the proficiency of the groups of test-takers. The role of this analysis is to ensure the automatically produced grades are not systematically biased against any particular group based on variance that is irrelevant to the construct of interest. Additionally, DIF analysis can play a confirmatory role as a process that provides a check on fairness and bias, or sensitivity review to ensure that the human review prior to launching items is indeed screening out items that exhibit DIF.

6. Conclusions

In a digital-first assessment, the steps in the test development are designed and built differently from the traditional tests, but the measurement requirements for valid, reliable, and generalizable test results remain. This chapter provided an overview of the theoretical and methodological underpinnings of digital-first assessments, illustrated with a language assessment, the Duolingo English Test.

By leveraging the methods presented in this chapter in a way that is underpinned by theories of language learning and assessment, it is possible to develop an English language assessment that supports valid and reliable interpretations and uses of test scores while also creating an assessment that enhances test-takers' experience and improves access to the assessment. Although characteristics of digital-first assessment were contextualized with examples from the Duolingo English Test, the tools and processes presented could be extended to other types of tests (e.g., tests of other languages, other purposes for measuring English language proficiency, and tests of other skills and abilities, such as math). Our daily lives are becoming increasingly digitally mediated, where appropriate measures of skills and abilities need to start incorporating the effects of this phenomena. Additionally, assessments that are created for digital administration can do more to start leveraging how digital administrations can facilitate enhanced measures of existing constructs. The Highlight the Answer question in the Interactive Reading task is a step in this direction. This item format would be very difficult to score in a paper-based administration. However, in digital administrations, it is trivial to keep track of a highlighted answer and compare it with the expected response to get a score. Furthermore, it improves how the construct of identifying information in a text is measured because test-takers have to find and identify the information rather than select the answer from a list of options. Additionally, the process of highlighting a text is an authentic and meaningful reading activity. It is a form of annotation that occurs in the target domain of university study and is indicative of reading ability (Winchell et al., 2020).

While digital-first assessment can improve the test-taking experience for test-takers, the process of development and administration does introduce new and interesting considerations. As discussed in this chapter, additional and varied approaches to security are necessary, such as an extremely large item bank to prevent the likelihood of item preknowledge (LaFlair et al., 2022). Additionally, the content review process is slightly different. The latest advance in large language models, such as GPT-3, do allow for the creation of quality texts. However, not all texts that are produced by the models are of equal quality or of high enough quality to be included on a high-stakes assessment. As a result, it is necessary to implement both automated filters of the content as well as thorough human review of the content that the models create. The implementation of digital-first assessments requires leveraging many processes and tools used in traditional test development (i.e., content review, DIF analysis, creation of item banks), but occasionally it also requires test developers to either modify those processes or create new tools and processes.

Digital-first assessments require a sophisticated platform that allows for embedded automation and AI, combined with rigorous (computational) psychometrics in order to support the intended use of the test and the interpretation of the test results.

Note

1 Efficient item creation means being able to create a large number of items at scale and field them on the operational test at a lower cost than traditional high-stakes test development by minimizing the number of person-hours spent reviewing and number of pilot observations required.

References

Alderson, J. C. (2000). *Assessing reading*. Cambridge University Press.

Allalouf, A., Gutentag, T., & Baumer, M. (2017). Quality Control for Scoring Tests Administered in Continuous Mode: An NCME Instructional Module. *Educational Measurement: Issues and Practice, 36*(1), 58–68.

Attali, Y. (2011). *A differential word use measure for content analysis in automated essay scoring (ETS RR-11-36)*. Educational Testing Service.

Attali, Y., & Burstein, J. (2006). Automated essay scoring with e-rater v.2.0. *Journal of Technology, Learning, and Assessment*, *4*(3).

Biber, D. (1988). *Variation across speech and writing*. Cambridge University Press.

Biber, D., & Conrad, S. (2019). *Register, genre, and style*. Cambridge University Press.

Bradlow, A. R., & Bent, T. (2002). The clear speech effect for non-native listeners. *Journal of the Acoustical Society of America*, *112*, 272–284.

Bradlow, A. R., & Bent, T. (2008). Perceptual adaptation to non-native speech. *Cognition*, *106*, 707–729.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., & Amodei, D. (2020). Language models are few-shot learners. *arXiv*. https://arxiv.org/abs/2005.14165

Buck, G. (2001). *Assessing listening*. Cambridge University Press.

Burstein, J., Kukich, K., Wolff, S., Lu, C., Chodorow, M., Braden-Harder, L., & Harris, M. D. (1998, August). *Automated scoring using a hybrid feature identification technique*. Proceedings of the Annual Meeting of the Association of Computational Linguistics.

Burstein, J., LaFlair, G. T., Kunnan, A. J., & von Davier, A. A. (2022). *A theoretical assessment ecosystem for a digital-first assessment – The Duolingo English test*. https://duolingo-papers.s3.amazonaws.com/other/det-assessment-ecosystem.pdf

Burstein, J., Marcu, D., & Knight, K. (2003). Finding the WRITE stuff: Automatic identification of discourse structure in student essays. In S. Harabagiu & F. Ciravegna (Eds.), *Special issue on advances in natural language processing, IEEE intelligent systems* (Vol. 18, No. 1, pp. 32–39). IEEE.

Cardwell, R., LaFlair, G. T., Naismith, B. & Settles, B. (2022). *Duolingo English test: Technical manual*. https://duolingo-papers.s3.amazonaws.com/other/det-technical-manual-current.pdf

Cushing-Weigle, S. (2002). *Assessing writing*. Cambridge University Press.

Davies, M. (2008). *The corpus of contemporary American English (coca)*. www.english-corpora.org/coca/

De Boeck, P., & Wilson, M. (Eds.). (2004). *Explanatory item response models: A generalized linear and nonlinear approach*. Springer.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv*. https://arxiv.org/abs/1810.04805

Dong, F., & Zhang, Y. (2016, November). *Automatic features for essay scoring – an empirical study* (pp. 1072–1077). Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing.

Egbert, J., Burch, B., & Biber, D. (2020). Lexical dispersion and corpus design. *International Journal of Corpus Linguistics*, *25*(1), 89–115. https://doi.org/10.1075/ijcl.18010.egb

Evanini, K., Singh, S., Loukina, A., Wang, X., & Lee, C. M. (2015). *Content-based automated assessment of non-native spoken language proficiency in a simulated conversation*. Proceedings of SLU NIPS 2015 workshop on Machine Learning for Spoken Language Understanding and Interaction.

Foltz, P. W., Kintsch, W., & Landauer, T. K. (1998). The measurement of textual coherence with latent semantic analysis. *Discourse Processes*, *25*(2), 285–307.

Gardner, D., & Davies, M. (2013). A new academic vocabulary list. *Applied Linguistics*, *35*(3), 305–327. https://doi.org/10.1093/applin/amt015

Gierl, M. J., Lai, H., & Turner, S. R. (2012). Using automatic item generation to create multiple-choice test items. *Medical Education*, *46*(8), 757–765.

Grabe, W. (2009). *Reading in a second language: Moving from theory to practice*. Cambridge University Press.

Graves, A. (2014). Generating sequences with recurrent neural networks. *arXiv*. https://arxiv.org/abs/1308.0850

Higgins, D., Burstein, J., & Attali, Y. (2006). identifying off-topic student essays without topic-specific training data. In J. Burstein & C. Leacock (Eds.), *Special issue of natural language engineering on educational applications using NLP*. IEEE.

Jessop, L., Suzuki, W., & Tomita, Y. (2007). Elicited imitation in second language acquisition research. *Canadian Modern Language Review*, *64*(1), 215–238.

Khodadady, E. (2014). Construct validity of C-tests: A factorial approach. *Journal of Language Teaching and Research*, *5*(6), 1353.

Klebanov, B. B., Madnani, N., Burstein, J., & Somasundaran, S. (2014, June). *Content importance models for scoring writing from sources* (Vol. 2, pp. 247–252). Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers).

Klein-Braley, C. (1997). C-tests in the context of reduced redundancy testing: An appraisal. *Language Testing*, *14*(1), 47–84.

LaFlair, G. T., Langenfeld, T., Baig, B., Horie, A. K., Attali, Y., & von Davier, A. A. (2022). Digital-first assessments: A security framework. *Journal of Computer Assisted Learning*, 1–10. https://doi.org/10.1111/jcal.12665

Langenfeld T, Burstein J and von Davier AA (2022) Digital-First Learning and Assessment Systems for the 21st Century. *Frontiers in Education*, *7,* 1-16. https://doi.org/10.3389/feduc.2022.857604

Leacock, C., Chodorow, M., Gamon, M., & Tetreault, J. (2010). Automated grammatical error detection for language learners. *Synthesis Lectures on Human Language Technologies*, *3*(1), 1–134.

Lee, Y.-H., & von Davier, A. A. (2013). Monitoring scale scores over time via quality control charts, model-based approaches, and time series techniques. *Psychometrika*, *78*, 557–575.

Liao, M., Attali, Y., & von Davier, A. A. (2021, July). *Quality assurance in digital-first assessments*. Paper presentation, International Meeting of Psychometric Society, Virtual.

Loukina, A., Madnani, N., & Cahill, A. (2017, September). *Speech-and text-driven features for automated scoring of English speaking tasks* (pp. 67–77). Proceedings of the Workshop on Speech-Centric Natural Language Processing.

Luoma, S. (2004). *Assessing speaking*. Cambridge University Press.

Magis, D., Yan, D., & von Davier, A. A. (2017). *Computerized adaptive and multistage testing with R: Using packages catR and mstR*. Springer.

McCarthy, A. D., Yancey, K. P., LaFlair, G. T., Egbert, J., Liao, M., & Settles, B. (2021). *Jump-starting item parameters for adaptive language tests* (pp. 883–899). Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing.

McLean, S., Stewart, J., & Batty, A. O. (2020). Predicting L2 reading proficiency with modalities of vocabulary knowledge: A bootstrapping approach. *Language Testing*, *37*(3), 389–411.

Milton, J. (2010). The development of vocabulary breadth across the CEFR levels. In I. Bartning, M. Martin, & I. Vedder (Eds.), *Communicative proficiency and linguistic development: Intersections between SLA and language testing research* (pp. 211–232). Eurosla.

Milton, J., Wade, J., & Hopkins, N. (2010). Aural word recognition and oral competence in English as a foreign language. In R. Chacón-Beltrán, C. Abello-Contesse, & M. Torreblanca-López (Eds.), *Insights into non-native vocabulary teaching and learning* (Vol. 52, pp. 83–98). Multilingual Matters.

Mislevy, R. J., Almond, R. G., & Lukas, J. F. (2003). A brief introduction to evidence-centered design. *ETS Research Report Series*, *2003*(1), i–29.

Pagel, V., Lenzo, K., & Black, A. (1998). *Letter to sound rules for accented Lexicon compression* (pp. 2015–2020). Proceedings of the International Conference on Speech and Language Processing.

Reichert, M., Keller, U., & Martin, R. (2010). The C-test, the TCF and the CEFR: A validation study. Der C-Test: Beiträge Aus der Aktuellen Forschung. In R. Grotjahn (Ed.) *The C-test: Contributions from current research* (pp. 205–231). Peter Lang.

Settles, B., LaFlair, G. T., & Hagiwara, M. (2020). Machine-learning driven language assessment. *Transactions of the Association of Computational Linguistics*, *8*, 247–263.

Shermis, M. D., & Burstein, J. (2013). *Handbook of automated essay evaluation: Current applications and future directions*. Routledge.

Somasundaran, S., Burstein, J., & Chodorow, M. (2014, August 23–29). *Lexical chaining for measuring discourse coherence quality in test-taker essays*. The 25th International Conference on Computational Linguistics (COLING).

Staehr, L. S. (2008). Vocabulary size and the skills of listening, reading and writing. *Language Learning Journal*, *36*, 139–152.

Wang, X., Evanini, K., Qian, Y., & Mulholland, M. (2021, January). Automated scoring of spontaneous speech from young learners of English using transformers. In *2021 IEEE spoken language technology workshop (SLT)* (pp. 705–712). IEEE.

Winchell, A., Lan, A., & Mozer, M. (2020). Highlights as an early predictor of student comprehension and interests. *Cognitive Science*, *44*(11). https://doi.org/10.1111/cogs.12901

Vinther, T. (2002). Elicited imitation: A brief overview. *International Journal of Applied Linguistics*, *12*(1), 54–73.

von Davier, A. A. (2015). *Virtual and collaborative assessments: Examples, implications, and challenges for educational measurement* (F. Bach & D. Blei, Eds.). Workshop on Machine Learning for Education, International Conference on Machine Learning.

von Davier, A. A. (2017). Computational psychometrics in support of collaborative educational assessments. *Journal of Educational Measurement*, *54*(1), 3–11.

von Davier, A. A., Deonovic, B., Yudelson, M., Polyak, S., & Woo, A. (2019). Computational psychometrics approach for holistic learning and assessment systems. *Frontiers in Education*, *4*(69). https://doi.org/10.3389/feduc.2019.00069

von Davier, A. A., Mislevy, R. J., & Hao, J. (2021). *Computational psychometrics: New methodologies for a new generation of digital learning and assessment. With examples in R and Python*. Springer-Verlag.

Wainer, H., Dorans, N. J., Green, B. F., Steinberg, L., Flaugher, R., Mislevy, R. J., & Thissen, D. (1990). *Computerized adaptive testing: A primer*. Lawrence Erlbaum Associates.

Zellers, R., Holtzman, A., Rashkin, H., Bisk, Y., Farhadi, A., Roesner, F., & Choi, Y. (2019). Defending against neural fake news. *arXiv*. preprint arXiv:1905.12616. https://arxiv.org/abs/1905.12616

Zieky, M. J. (2006). Fairness reviews in assessment. In S. M. Downing & T. M. Haladyna (Eds.), *Handbook of test development* (pp. 359–376). Routledge.

Zieky, M. J. (2016). Developing fair tests. In S. Lane, M. R. Raymond, & T. M. Haladyna (Eds.), *Handbook of test development* (2nd ed., pp. 359–376). Routledge.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Part III

Validity and Fairness

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

8

Validity, Fairness, and Technology-Based Assessment

Suzanne Lane

Technology provides opportunities but also poses challenges in the design and validation of assessments. The use of technology in assessment programs ranges from automated scoring to the design of simulation tasks. The design of technology-based assessments leverages artificial intelligence (AI) and natural language processing (NLP) along with psychometric methods, allowing for the measurement of complex cognitive skills that have been difficult to measure on a large scale, including written and spoken language. An overarching goal of the use of technology in test design is to improve construct representation by more accurately capturing and representing examinees' processes and actions while minimizing sources of construct-irrelevant variance. The design of technology-based assessments requires experts from multiple disciplines, including relevant content areas, psychometrics, test design, computational linguistics, and NLP, to collaborate from the design phase through to the evaluation of the interpretation and use argument.

Adopting an argument-based approach to validity (Cronbach, 1988; Kane, 1992, 2006, 2013), this chapter addresses validity and fairness issues when using automated scoring, assessments composed of computer-based simulations, and automated item generation. To provide a context for discussing validity and fairness issues pertaining to technology-based assessments first is a discussion on validity and fairness and their relationship, followed by a discussion on validity and fairness issues in testing individuals with diverse cultural and linguistic backgrounds. Next is a discussion on validity and fairness issues in using AI and NLP in the design and use of automated scoring engines, technology-based simulation, and automated item generation. These sections do not represent an exhaustive treatise on validity and fairness issues in the design and use of technology-based assessment; instead, the intent is to highlight some relevant issues.

1. Validity

Validity underscores all aspects of the testing process, from defining the construct, to designing the test, to evaluating the consequences of test use (Lane & Marion, forthcoming). Validity

DOI: 10.4324/9781003278658-11

This chapter has been made available under a CC-BY-NC-ND 4.0 license.

refers to the extent to which theoretical, logical, and empirical evidence supports or refutes test score interpretations, decisions, uses, and their resulting consequences (Cronbach, 1988; Kane, 2006). The specific purposes and uses of tests frame validity investigations and the evidence needed in support of test use. The first step in testing is to clearly articulate the intended interpretations and uses of test scores and the intended consequences of test use. This also includes identifying any potential unintended, negative consequences so as to take steps to help minimize them.

Construct underrepresentation and construct-irrelevant variance are two threats to the validity of score interpretations and uses. Construct underrepresentation occurs when the assessment does not measure important aspects of the intended construct. As an example, construct underrepresentation occurs if an assessment based on automated item generation includes only more easily generated items, such as items that require only recall, and the targeted construct (i.e., knowledge, skills, and other attributes, or KSAs) calls for a range of cognitive complexity. Construct-irrelevant variance occurs when the assessment measures not only the targeted construct, but also factors that are irrelevant to the targeted construct. For a writing assessment, if features in an automated scoring algorithm reflect aspects that are irrelevant to the writing construct, construct-irrelevant variance arises.

Interpretation and use arguments (IUAs) and validity arguments provide a framework for evaluating the validity and fairness of score interpretations and uses for technology-based assessments. An IUA 'specifies the proposed interpretations and uses of test results by laying out the network of inferences and assumptions leading from the observed test performances to the conclusions and decisions based on the performances' (Kane, 2006, p. 23). A clear delineation of the proposed interpretations, uses, and consequences allows for consideration of the needed validity evidence early in test design. The validity argument involves obtaining theoretical, logical, and empirical evidence to evaluate the soundness of the claims and underlying assumptions in the IUA. The validity argument is an evaluation of the plausibility of the proposed IUA by providing an analysis of the evidence for and against proposed score interpretations and uses (Cronbach, 1988; Kane, 1992, 2006). As an example, a claim in the IUA for an automated scoring system may refer to the appropriateness of the system for all relevant groups, with the assumption that score inferences and uses are similarly valid across groups. Validity evidence to support the claim includes how the design of the system accounted for different groups and the extent to which the prediction to human scores is similarly accurate across groups.

Inferences that are part of the IUA that provide a validation framework include the scoring, generalization, extrapolation, and decision/uses inferences (Kane, 1992, 2006). These inferences involve the evaluation of test performance to test score (i.e., the extent to which the test score reflects the intended assessment of examinee performance); generalization of test score to the expected score over the intended universe or population of items, testing occasions, raters, contexts, and other relevant facets; extrapolation from the hypothetical universe to the broader domain of practice in the real world; and appropriateness of decisions and uses as well as their resulting consequences.

To help illustrate each of these inferences, an example of the type of evidence to support each inference follows. Evidence for the scoring inference entails an evaluation of the accuracy and consistency of applying the scoring rules to all examinees. Evidence for the generalization inference includes the adequacy of the sampling of items on an assessment and the extent to which it allows for valid generalizations to the intended item population. The evaluation of the extrapolation inference includes expert judgment of the similarity between the KSAs used in practice (i.e., real world) and the tested KSAs. As an example, the KSAs assessed on a teacher certification test should reflect actual teaching practices in the classroom if the test claims to do so. Evidence that decisions and consequences of test use do not lead to unintended, negative consequences for relevant groups provides support for the decision/use inference.

In addition to these inferences in the interpretive chain, the ‘performance inference’, linking the items to the examinee performance, requires evidence in support of the claim that the items elicit the intended KSAs and that the test is representative of the construct (Lane & Marion, forthcoming; Lane & Perez, 2023). The performance inference explicitly positions the conceptualization of the construct and test design at the beginning of the IUA. As an example, if a simulation task requires examinees to analyze and evaluate the relationship among relevant sources of information, asking examines to think aloud as they solve the task can provide an indication of whether the task elicits the intended KSAs. The evaluation of the construct representation of tasks in terms of the KSAs’ underlying item performance as well as principled approaches to test design also provide backing in support of the performance inference.

The relative importance of each of the inferences in the IUA and the evidence to support or refute each inference is dependent on the purpose and use of a particular test. To support each inference typically requires multiple sources of evidence. For each testing application discussed in this chapter, additional examples of evidence to support these inferences are provided.

1.1 Fairness and Validity

At the core of validity investigations is the appraisal of the fairness of score meaning and test use for different groups of examinees, defined by ethnicity, race, culture, language, disability, and other relevant characteristics. Marion and I (Lane & Marion, forthcoming) argue that validity is an overarching concept that subsumes fairness, and therefore fairness investigations are validity investigations. As indicated in the *Standards for Educational and Psychological Testing*, fairness is a ‘fundamental validity issue and requires attention throughout all stages of test development and use’ (AERA et al., 2014, p. 49). Fairness issues arise throughout the testing process, from identifying the purpose and use of the test to specifying and evaluating the intended and potential unintended consequences of test use. Kane (2010) claimed that fairness is central to the evaluation of consequences of test use, especially when there are differential outcomes across relevant groups. Fairness investigations require evidence for the delineated inferences, uses, and consequences for relevant groups in the targeted population, with attention paid to the heterogeneity of individuals within each defined group.

The values held by those who make decisions throughout the testing process can threaten the validity and fairness of test score interpretations and uses for different groups of examinees (Cronbach, 1976; Kane, 2006; Messick, 1989). Value judgments arise in the articulation of the intended uses of a test, the conceptualization of the target construct, the design and development of items and scoring rules, and the development of performance standards (Kane, 2006), as well as the articulation of the intended positive consequences and potentially unintended, negative consequences. Throughout the testing processes, including the design of the IUA and validity argument, attention to the differing values and beliefs of different stakeholder groups must be considered.

2. Fairness in Testing Individuals with Varying Cultural and Linguistic Backgrounds

Testing is positioned within a cultural context and typically reflects the values and beliefs of those who mandate and design tests. To address validity and fairness issues for different cultural groups requires the inclusion of critical individuals from these groups in the design, development, implementation, use, and validation of the testing process (Lane & Marion, forthcoming). The intended construct may not be assessed by a test, but instead the test may assess the individual's KSAs as they have been shaped through cultural, linguistic, and social experiences (Hood, 1998; Mislevy, 2018). Test design and validation need to address the critical role of how cultural and linguistic histories shape how individuals learn and demonstrate their KSAs.

For more valid and fair test score meaning and use, tests should reflect the way individuals learn and express their KSAs in their design (Trumbull & Nelson-Barber, 2019). The necessary evidence to support the validity of score meaning and test use for individuals with diverse backgrounds needs consideration throughout the testing process, from defining the construct to examining the consequences of test use for relevant groups. As an example, standardization in test materials could be a source of invalidity across different cultures because individuals' cultural and linguistic experiences affect how they demonstrate their KSAs (Messick, 1989). Considering testing as a cultural practice and acquisition of KSAs as culturally bounded has the potential to lead to more valid and fair testing practices.

Mislevy (2018) proposed that examinee experiences with certain linguistic, cultural, and substantive (LCS) patterns can help explain variance in test performance. Inferences made about examinee competency from task performance rest on the LCS resources the examinee uses to respond to tasks. Leveraging knowledge about the examinees' backgrounds can enhance validity (Mislevy, 2018). Test design that specifies task features that represent the targeted KSAs, and task features associated with non-targeted LCS patterns that provide alternative explanations of competencies, may lead to more fair assessments (Mislevy, 2018). Of course, evidence is required to support such assumptions.

Principled approaches to test design, such as evidence-centered design (ECD) (Mislevy et al., 2003), affords a mechanism to make assessments responsive to different cultural, linguistic, and educational groups (see Mislevy, 2018). Such design approaches clearly articulate the KSAs needed for successful task performance. Delineation of the claims and evidence to support those claims can help minimize construct-irrelevant variance and help ensure representation of the intended KSAs. Including critical individuals from relevant groups in all aspects of test design and validation, including the specification of the IUA and validity argument, helps ensure more inclusive assessments (Lane & Marion, forthcoming).

3. Automated Scoring

Most automated scoring systems for written and spoken language involve the extraction of features from responses, and then a machine learning algorithm maps the features in an examinee response to a human score. Bejar (2017) labeled these two steps as feature extraction and evidence synthesis. Feature extraction entails analyzing a response into a set of features related to the target construct, and evidence synthesis includes mapping the weighted features onto a score level which typically involves regressing human scores on the extracted features. To model the scores for each response, a machine learning algorithm, previously trained on a corpus of responses, identifies the features and their weighting. A threat to validity is a lack of understanding of 'how sets of features interact for particular responses to reflect cognitive response processes of learners or raters' (Rupp, 2018, p. 200). This rests on the assumption that both learners and raters are using relevant cognitive processes. Another added layer in automated scoring systems for spoken language that may impact validity is an automated speech recognition (ASR) system that first provides text transcriptions of recordings of the spoken language.

Automated scoring algorithms embody features that can enhance validity, such as consistency in applying the scoring rubric, the control of features used for scoring responses, and the capacity to capture multiple aspects of performances (Powers et al., 1998; Williamson et al., 2006). The uniform application of the algorithm has the potential to promote score comparability, but if the algorithm is more appropriate for some groups over others, its use poses threats to validity and fairness. Validity is also threatened if the assessment measures only those KSAs that are amenable to machine scoring and disregards other relevant KSAs. Principled approaches to design will facilitate the identification of relevant features, minimizing such validity threats.

3.1 IUAs and Validity Arguments

IUAs and validity arguments for automated scoring engines (ASEs) depend on the specific interpretation and use argument, but there are some general aspects that are relevant to most ASEs. Williamson et al. (2012) and Bennett and Zhang (2016) proposed argument-based validity frameworks for ASEs that focus on construct relevance and representation. When discussing a validity argument for ASEs, Bejar (2011) distinguished between defects in the design of ASEs and quality defects that can be corrected when monitoring produced scores. Rupp (2018) extended Bejar's framework by focusing on a validity argument for methodological design decisions – those affecting the system design and those affecting the quality control of the system – in the development, evaluation, and implementation of ASEs.

The focus of this section is on the aspects of IUA and validity arguments that address validity and fairness issues in the design of ASEs for a diverse population (also see Lane & Marion, forthcoming). Although not discussed here, the ASR system for spoken language assessment requires validity evidence.

3.1.1 IUA and Validity Arguments for Human Scores

Automated scores depend on the quality of the human scores used in training the algorithm and evaluating the automated scores. A common source of validity evidence to support the scoring inference involves the comparison of the relationship between human and automated scores to the relationship between two human scores. Bernstein et al. (2020) found that automated comprehension and expression scores for an oral fluency test correlated higher with human scores as compared to human-to-human correlations. The appropriateness of this type of comparison as well as other uses of human scores in the design and validation of ASEs rests on the accuracy and consistency of human scores. Bejar (2012) proposed a first-order validity argument – human scores require an appraisal of their own IUA by examining the rater response processes relative to the scoring rubric and the construct domain. The extent to which rater scores reflect the construct as intended and reflect irrelevant constructs depends on raters' interpretation and implementation of the rubric (Lane & DePascale, 2016; Lane & Stone, 2006). Evidence for the performance inference for raters should include rater think-alouds to evaluate whether raters have a shared understanding of the rubric as they apply it to examinee responses, and whether they accurately apply it to responses from relevant groups of examinees (Lane & Marion, forthcoming). Evaluations of the rubrics, exemplars, and training materials and procedures provide additional validity evidence.

The generalizability of human scores over raters, rater occasions, and sampling of responses provides evidence for the generalization inference. The extrapolation inference for raters requires an examination of the expected relationships between human scores and scores on other measures intended to assess similar and different constructs. To ensure the validity of human score meaning and uses requires an evaluation of the invariance of these generalizations and relationships across relevant groups. There should be sufficient validity and fairness evidence for human scoring prior to the design of an ASE.

3.2 IUA and Validity Arguments for Automated Scoring Engines

The conceptualization of the ASE should be at the forefront of the assessment design process so that identification and weighting of features allow for a scoring model that sufficiently captures the intended construct and does not assess irrelevant constructs (Bejar, 2017; Lane, 2017).

3.2.1 Scoring Inference

The scoring inference rests on the assumption that the set of features and their weighting underlying the machine learning model accurately predicts human performance. Features used in machine learning algorithms serve as a proxy for the targeted construct and may be a poor reflection of the construct, an oversimplification of the construct, and/or function differently across groups (Suresh & Guttag, 2021). Construct-irrelevant variance and construct underrepresentation affect the validity of score inferences and can result in a higher or lower score than the examinee deserved (Powers et al., 2002).

Features identified for automated scoring should be based on a model of proficiency, and they vary in terms of how well they represent the construct as well as predict human scores. Typically, for writing and speaking assessments, the extracted features are measures of latent sematic analysis, and their representation of the construct varies (Rupp, 2018). The synthesis of features can rely on relatively transparent modeling approaches such as regression or, more recently, black box modeling approaches such as neural networks that may identify construct-irrelevant features (Rupp, 2018). Regardless of modeling approach, evidence for the scoring inference should include an evaluation of the features and their weighting in terms of how well they represent the construct and do not contribute to construct-irrelevant variance.

Mislevy (2018) discusses test design in terms of task features associated with expected targeted linguistic, cultural, and substantive patterns and non-targeted patterns that provide alternative explanations of competencies. Applying such an approach to the design of ASEs may lead to more equitable assessments. While this is a complex challenge, the attention to different knowledge acquisition and learning styles has the potential to enhance score meaning for examinees from diverse cultural and linguistic backgrounds.

Validity and fairness issues arise in training and calibrating the algorithm. The sample of scored examinee responses for training and calibrating should be representative of the different types of responses. The sample of responses for training should be drawn from the relevant groups within the population of potential examinees to ensure the scoring algorithm does not favor one group's way of responding over another (Kolen, 2011). Measures developed to evaluate the fairness of algorithms used outside of testing (e.g., predicting promotion of firefighters, predicting whether individuals are a good or poor credit risk) may be of value. These measures are based on group-conditional accuracy of the predicted outcome, which allows for evaluating the similarity of the error rates across groups (e.g., Friedler et al., 2018). Later chapters in this text propose statistical measures to evaluate the fairness of ASEs.

An evaluation of whether a nonlinear predictive model leads to a better fit than a linear model may reveal that some construct-relevant features are not linearly related to response quality (Foltz, 2020; Foltz et al., 2013). Foltz and colleagues (2013) demonstrated that the feature related to the coherence of the response departed from linearity at the continuum ends; the upper end of the continuum reflected too much coherency, which may suggest repetitive essays. ASEs may also perform similarly by general measures but show different performance patterns across score performance categories (Chen et al., 2016). Designing scoring algorithms requires an understanding of the behavior of the variables in the algorithm, the relationship between the features and the construct, and the modeling that best accounts for performance (Foltz, 2020).

Also needed is an evaluation of the extent to which the evidence identification and synthesis are vulnerable to construct-irrelevant response strategies such as gaming (Bejar, 2017). Because gaming strategies may have effects on the behavior of automated scoring algorithms, ASE designers need to evaluate such construct-irrelevant strategies prior to operational administrations to mitigate their impact (Bejar, 2013; Williamson et al., 2012).

3.2.2 Generalization Inference

Underlying the generalization inference is the assumption that ASEs generalize to prompts, occasions, groups, and other relevant facets. Backing to support the generalization inference includes evaluations of the generalizability of automated scores to other tasks based on the same task model, occasions, contexts, and groups as well as evaluations of the expected relationships to external criteria (Bennett & Zhang, 2016; Clauser et al., 2002; Williamson et al., 2012). As an example, there is a tradeoff in developing a general scoring algorithm or prompt specific algorithms for written and spoken language assessments (Rupp, 2018). A general algorithm may lower predictive accuracy but increase generalizability, whereas a prompt-specific algorithm may increase predictive accuracy but decrease generalizability. Messick's (1994) suggestion regarding the design of rubrics should be considered in the design of scoring algorithms in that they should not be 'specific to the task nor generic to the construct but are in some middle ground reflective of the classes of tasks that the construct empirically generalizes or transfers to' (p. 17).

Generalizability of the ASE across relevant cultural and linguistic groups provides evidence on the extent to which subgroups are differentially impacted by the ASE. As an example, on a state writing assessment, both Asian American and Hispanic students received higher scores from the ASE than from human raters, whereas White and African American students scored similarly across the two scoring methods (Bridgeman et al., 2009). Under the assumption that Asian American and Hispanic groups have a higher proportion of students with English as a second language, the authors suggested that this finding may be due to linguistic differences. An evaluation of an ASE for spoken language showed that ASE scores were lower for Germans than corresponding human scores as compared to other language groups (Wang et al., 2018). The authors indicated that a reason for this difference could be that features relevant for German speakers were not included in the design of the ASE. When examining model-data fit for relevant groups, an evaluation of the cognitive theory underlying performance and think-alouds might shed light on differences in score patterns across groups.

Kolen called for research studies that evaluate score comparability across identifiable groups, such as cultural groups, by having the algorithm trained only on responses from one examinee group, only on responses from a second examinee group, or on a combination of the groups' responses (2011; personal communication, June 17, 2019). Such a study addresses the generalizability of the algorithm across responses from different groups. Training data may capture historical discrimination, or there may be subtle patterns in the data such as under-representation of a marginalized group (Friedler et al., 2018).

3.2.3 Extrapolation and Decision/Use Inference

Evidence to support the extrapolation inference includes evaluating the relationship between automated scores and performance on a real-world criterion and evaluating whether the expected relationships between measures of similar and different constructs hold for relevant groups. The decision/use inference for formative assessment scores requires an evaluation of the assumption that formative feedback provided by automated scores improves performance.

Mao et al. (2018) provided evidence for the decision/use inference in that feedback based on students' automated scores on a formative assessment of science argumentation prompted students to modify their responses and improved student scientific argumentation skills as they revised their responses. To address consequential issues related to fairness requires an examination of the potential differential impact on improved learning.

The appropriateness of the algorithm and the process used to develop the algorithm impact the soundness of the decisions, uses, and consequences for relevant groups. Performance-level decisions in education and pass/fail decisions in certification and licensing may result in negative consequences if the algorithms measure factors irrelevant to the construct, do not sufficiently represent the construct, and do not accurately capture different ways of knowing and performing.

4. Simulation-Based Assessments

Technological advances in the design of computer-based simulation tasks allow for the measurement of complex skills that are difficult to measure in other assessment forms. Simulations can capture evidence of the processes underlying performance in real time and assess a wider and deeper range of examinee behaviors. Advantages and challenges of simulation-based assessments are like those of other performance assessments (Lane & Stone, 2006), but additional challenges arise that may threaten validity. Features to consider in the design and validation of simulations include the nature of examinee interactions with the tools in the virtual environment and the recording of how examinees use the tools (Vendlinski et al., 2008). Validity evidence to support their use includes the extent to which the simulation-based assessment represents the construct and does not measure irrelevant features such as familiarity with the interface. The designer also needs to guard against restricting the intended range of content and cognitive skills to those skills that are more easily assessed using computer technology.

4.1 IUAs and Validity Arguments for Simulation-Based Assessments

IUAs and the relevancy of each inference depend on the purpose and use of a particular simulation-based assessment, but there are some general aspects across IUAs (also see Lane & Marion, forthcoming). A common belief about simulation tasks is improved construct representation and improved extrapolation from the performance to the real-life context. This depends, however, on meeting the assumption that the KSAs required by the simulated performance reflect those used in the real-life context. Simulation tasks may model some features of the real task and not others, limiting construct representation. Capturing the full richness of complex performances may be challenging due to constraints in the virtual environment.

Evidence to support the assumption that the simulation reflects the actual performance includes evidence gathered from the task and scoring design processes. This includes cognitive models used in the design and their appraisal by content experts. Empirically and theoretically based cognitive models are not available or fully developed in many domains; thus, specification of the cognitive process features may pose challenges and affect the validity of score meaning. As an example, Andrew et al. (2017) detected a source of construct-irrelevant variance in a simulation-based assessment requiring collaborative problem-solving by modeling different patterns of interaction. They identified a 'fake collaboration' pattern where both group members appeared to collaborate; however, regardless of their prior agreement with the other's response, they kept their own responses.

4.1.1 Performance Inference

The use of principled approaches to test design, such as ECD, for simulation-based assessments provides validity evidence in support of the inference made from the task to the target performance.

Examinee models comprised of the targeted KSAs are linked to task models and evidence models that describe examinee competency. The examinee, task, and evidence models provide validity evidence. Construct representation rests on the alignment of the elicited KSAs and interaction patterns in the virtual assessment with those used in real-life contexts (Mislevy, 2018). Explicit delineation of claims and evidence can help minimize construct-irrelevant variance.

Along with a principled approach for designing simulation-based assessments, the use of machine learning and AI holds promise in their design. As examinees engage in simulation-based, game-based, and collaborative problem-solving assessments, streams of fine-grained examinee activity, processes, and interaction data can be mined to assess competency. These data can be used to generate feedback based on the actions and interactions of the examinee. In game-based assessment, features of paths and actions collected in log files and work products provide evidence of players' processes, strategy use, and metacognitive skills (Mislevy et al., 2016). In collaborative problem-solving assessments, log files consisting of process data provide information on the interactions of team members (von Davier, 2017). Virtual tasks designed to generate logged actions can serve as evidence for a competency model in ECD, and evidence models can be designed based on statistical rules that are informed by content experts and encoded using neural networks (Mislevy, 2018). Such modeling allows for estimating the probability that an examinee has mastered a specific KSA component conditional on the response sequences given to previous task features. Data mining techniques have the potential to produce scores based on different clusters or patterns of responding (Mislevy et al., 2012). Validity and fairness issues occur if the model does not account for different ways of responding.

Decisions made in the design phase of the simulation affects examinee performance and scores (Mislevy et al., 2016). Evidence for the performance inference includes an evaluation of how score meaning and use can vary under different design choices. Well-designed virtual agents may allow for individuals to better demonstrate their KSAs (Rosen, 2017; Scoular et al., 2017), but the validity of the inferences depends on the nature of the virtual agent. Although virtual agents allow for uniformity in paths for those examinees who perform the same action, they may be a source of construct-irrelevant variance due to not adjusting to differences in ways that examinees respond (Rosen, 2017).

The goal of capturing and representing diverse cultural norms and linguistic patterns in the design phase along with attending to relevant features of task design and construct representation is to achieve more equitable assessments (Mislevy, 2018). Inclusion of cultural and linguistic experts in the design phase will help uncover construct-relevant cultural differences in interaction patterns and avoid introducing construct-irrelevant variance (Oliveri et al., 2019). Response process evidence obtained using think-aloud sessions can provide evidence for the performance inference and can uncover potential sources of both construct-relevant and construct-irrelevant variance. Think-aloud sessions allow for investigating whether features of the interaction space, such as virtual agents and linguistic features, impede performance for individuals from relevant groups.

4.1.2 Scoring, Generalization, Extrapolation, and Decision/Use Inferences

Clauser et al. (2016) examined the warrants and backing for the scoring, generalization, extrapolation, and decision/use inferences for simulation-based assessments. Simulations produce a large amount of data that are scorable, which requires intentional approaches to identifying relevant data to score and combining the data to produce meaningful information. Evidence to support the scoring inference includes the uniformity of task administration based on the action taken by the examinee and the standardization of how scores are derived for examinees who have the same path when responding to the task (Clauser et al., 2016). Such standardization and uniformity, however, may introduce construct-irrelevant variance due to restricting different pathways that reflect different cultural, linguistic, and experiential backgrounds, and therefore affect the validity of score meaning for individuals with differing backgrounds.

Although construct representation may be supported for a particular task in terms of assessing the targeted KSAs, the generalization inference for the assessment score may be compromised. The assessment may be composed of a small number of simulation tasks, limiting construct representation and the generalizability of the score inferences to the broader construct domain. Evaluations of the generalizability of scores across groups bear on validity and fairness issues. Collaborative assessments may require the examination of the generalizability of performance across different groups of collaborators as well as generalizing from interacting with a virtual agent to a human collaborator (Rosen, 2017).

Content experts' evaluation of both the alignment of the KSAs of the simulation task to the KSAs needed to perform the task in the real-life setting and the relationship between assessment scores and performances in the real-life setting provides validity evidence to support the extrapolation inference (Clauser et al., 2016). Differences in processes and strategies used by examinees when responding to the task in a testing context as compared to the criterion setting can threaten validity (Clauser et al., 2016). Additional potential sources of construct-irrelevant variance include the use of an interface and representations that are not familiar to the examinee, and the level of examinee motivation. These sources may interact with the examinees' cultural, social, and experiential backgrounds, leading to potential validity and fairness issues for examinees from diverse cultures.

Quellmalz et al. (2011) reported some initial consequential evidence suggesting that simulations can narrow achievement gaps for English learners and students with disabilities. For a simulation-based science classroom assessment for middle school students, the performance gap for the simulation averaged 12.1% as compared to 25.7% on the traditional assessment. For students with disability, the performance gap averaged 7.7% for the simulation and 18% for the traditional test. Differences in the achievement gaps may suggest that English learners and students with disability had an easier time accessing the simulations as compared to more traditional tests. As indicated by the authors, this may have been due in part to the scaffolding and interactive features used in the design of the simulations.

Undesirable consequences can arise when an assessment consists of a small number of tasks and thereby affect the accuracy of test score decisions and uses. The IUA for a given simulation-based assessment would benefit from considering the issues outlined in this section, but it should be tailored to the specific interpretation and use argument.

5. Automated Item Generation

Automated item generation (AIG) approaches use task and item models as well as AI and NLP. Task model approaches generate items based on rules that embody theoretical and logical information regarding the features that represent the KSAs and how features are combined. This approach rests on trained content experts to develop the cognitive models. AI and NLP approaches typically generate items without the involvement of content experts except during the evaluation and use phase. Some AIG approaches allow for the estimation of the difficulty of the items based on model features representing the targeted KSAs. Advantages in using AIG include control of the KSAs assessed by items, larger-item banks, and increased test security due to the substantial number of items needed to develop test forms and CAT systems.

5.1 Task Model Approach

Drasgow et al. (2006) discuss the development of item models by using either a weak or strong theory approach. With a weak theory approach, typically experience and, to a lesser extent, research and theory provide the guidelines necessary for identifying and manipulating the elements in an item model that generates items (Drasgow et al., 2006). The manipulated elements are fewer under a weak theory model, which may result in item clones if constraints are not implemented. A strong theory approach provides a theoretical underpinning for identifying the content for an item model by first specifying a cognitive model that provides an organized representation of the KSAs (Bejar, 2002a, 2002b). As a model of difficulty, the cognitive model drives the design of item models with the expectation that items will have certain psychometric characteristics. The cognitive theory underlying the item models provides a rationale that inferentially links performance on items to the underlying construct (Bejar, 2002a, 2002b; Luecht & Burke, 2020). Because underlying cognitive models provide a theory of item difficulty and cognitive complexity, test design procedures and test content provide validity evidence to support score meaning. Item models that specify the manipulable item features to generate items, including features for the stem, distractors, and auxiliary information are then created (for an interpretation/use argument and validity argument for tests developed using the task-based model, see Lane, 2022).

5.2 AI and NLP Approach

AIG leverages NLP and AI when generating the stem and distractors of multiple-choice items, cloze items, and constructed-response items. Much of the AIG research using NLP has been for assessments of language learning, including reading comprehension, and medical assessments (Kurdi et al., 2020). The former is due to the public availability of a large corpus of text and the use of NLP tools for shallow understanding of texts (e.g., tagging parts of speech). The latter is due to ease of NLP tools for processing medical text such as named entity recognition (i.e., identifying named entities in text and classifying them into defined categories such as medical codes) and co-reference resolution (i.e., finding linguistic expressions in text that refer to the same real-world entity).

Approaches for generating items include syntax-based, semantic-based, and neural network approaches. Syntax-based approaches uncover the syntactic structure of the input to generate items using, for example, syntax parsing and part of speech tagging. These methods do not require an understanding of the input entities and their meaning, which may threaten validity in that the items may not closely represent the intended KSAs. In addition, syntactic clues may allow examinees to respond correctly without understanding the content which threatens validity (Kurdi et al., 2020). Using NLP techniques, such as topic modeling and keyword extraction, semantic-based approaches function at a deeper level by identifying text features that indicate the meaning of the information. These approaches typically rely on content sources such as taxonomies and ontologies. A validity threat occurs if the ontologies are not representative of the targeted test construct. Neural network approaches attempt to learn directly from an already existing database of items (e.g., von Davier, 2018). A validity threat for this approach arises if the items in an existing database do not represent the construct, measure irrelevant features, or are of inadequate quality.

5.3 IUA and Validity Argument for AIG

Although all previously mentioned inferences are relevant for tests using AIG, the following discussion focuses on the inferences that are more related to the use of AI and NLP for generating items and test forms: performance inference and generalization inference.

5.3.1 Performance Inference

The validity of the performance inference is dependent on evidence regarding representation of the construct and the extent to which construct-irrelevant features are assessed. As an example, construct underrepresentation arises if an assessment based on automated item generation only includes more easily generated items, such as factual items, whereas the targeted construct calls for a range of item cognitive complexity. To minimize threats against construct representation, Kurdi et al. (2020) suggests translating generated items to a machine-processable representation and computing item features to examine their effect on item difficulty. Validity threats also arise when generating stems and options separately because difficulty of both the stem and options affect overall item difficulty. Generating items that measure complex skills and controlling for difficulty for such items is challenging.

As an example, Gao et al. (2018) generated items with a broader range of difficulty. They demonstrated a sequence-to-sequence prediction model to generate questions from reading passages while controlling for item difficulty level. They used two difficulty categories of questions: easy questions, where the answers are facts described in the text; and difficult questions, where the answer is not explicitly stated in the text. For generating difficult questions, position embeddings were trained to 'capture the proximity hint of the answer in the input sentence' (p. 2). Gao and colleagues provide validity evidence for their approach by using two reading comprehension systems to automatically assess the difficulty of the questions and comparing the generated difficulty level of a sample of questions with human judgments of their difficulty. The average difficulty human rating for the automated labeling of easy items was lower than that for difficult items, providing some validity evidence for their approach. Additional validity evidence for such an approach would include using examinee performance to evaluate difficulty prediction models and the appraisal of the quality of items by content experts (Kurdi et al., 2020).

A major challenge in developing multiple-choice items is developing plausible distractors. Using concept mapping (i.e., cosine similarity), Ha and Yaneva (2018) used the item stem and correct answer as input to produce a list of suggested distractors, and then information-retrieval methods ranked the distractors based on their similarity to the stem and correct answer. To evaluate the approach, they took the existing human-developed items and options to predict one or more of the existing distractors. As they indicate, their method relies on the quality of the distractors developed by humans. In a similar study, Baldwin (2021) used NLP to mine existing item banks for potential distractors based on the similarity between a new item's stem and answer and the stems and options for items in the bank. Using a prediction algorithm, Baldwin estimated the probability of an option being an appropriate distractor for the new item. For both studies, the intent is to provide a list of generated distractors to aid item writers in developing items. Validity threats arise if the human-generated items are of inferior quality and not representative of the construct. These approaches, however, address validity concerns regarding methods that only use the correct answer to generate distractors. As indicated by Kurdi et al. (2020), varying only the similarity between the key and distractors disregards construct-relevant facets of difficulty. Typically, a well-designed distractor reflects partial understanding or a misconception. These approaches, which are not based on cognitive models or theory, rely on content experts' review and revision.

5.3.2 Generalization Inference

The generalization inference involves inferring from performance to the test score over the broader domain of items, contexts, and so on. Studies to support the generalization inference include examining the internal structure of the test, evaluating the exchangeability of generated test forms, and generalizability studies examining the extent to which the sampled items generalize to the targeted domain.

As an example, von Davier (2018) used neural networks trained on a database of over 3,000 publicly available personality items to generate items and examined the internal structure of the set of items. The AI prediction model predicted the next character in the string until the item was completed. A factor analysis of a selected number of administered generated items and items publicly available indicated comparable internal structures for the generated items and human-developed items, providing validity evidence for the internal structure of the generated items.

Cole et al. (2020) used NLP and unsupervised ML techniques for syntactic partitioning of automatically generated items to construct parallel forms and for automated test assembly. They demonstrate an unsupervised clustering approach (K-means clustering) of the automatically generated items into syntactically distinct categories to facilitate selection of similar or dissimilar items. As they discuss, a validity threat was the sole reliance on syntactic features to represent the clusters in which items vary. Because semantic features and information about the targeted KSAs were not used to generate test forms, threats to construct representation arise. When assembling test forms using AIG items, other validity threats include content overlap within forms and differences in both construct representation and difficulty across forms (Kurdi et al., 2020). To minimize such validity threats, designers should evaluate the exchangeability of generated items and forms in terms of content complexity, cognitive demand, and psychometric properties.

6. Concluding Thoughts

Validity and fairness are fundamental to all aspects of testing. The design of technology-based assessments begins with an IUA that explicitly links the inferences from the tasks to the decisions and uses. The validity argument provides a framework, including the assumptions and evidence, to support (or refute) the inferences and uses specified in the IUA. Much research has focused on minimizing the validity threats related to the use of AI and NLP in automated scoring, whereas the use of AI and NLP for generating items and test forms requires continued research that focuses on minimizing threats to underrepresenting the targeted KSA and assessing construct-irrelevant features.

Acknowledgment

I thank Victoria Yaneva and Matthias von Davier, editors of this text, as well as two external reviewers for their helpful comments on an earlier version of this chapter.

References

American Educational Research Association (AERA), American Psychological Association (APA), & National Council on Measurement in Education (NCME) (2014). *Standards for educational and psychological testing*. American Educational Research Association.

Andrew, J. J., Kerr, D., Mislevy, R. J. von Davier, A., Hao, J., & Liu, L. (2017). Modeling collaborative interaction patterns in a simulation-based task. *Journal of Educational Measurement*, *54*, 54–69.

Baldwin, P., Mee, J., Yaneva, V., Paniagua, M., D'Angelo, J., Swygert, K., & Clauser, B. E. (2021). A natural language-processing-based procedure for generating distractors for multiple-choice questions. *Evaluation & the Health Professions*. https://doi.org/10.1177/01632787211046981

Bejar, I. I. (2002a). Generative testing: From conception to implementation. In S. H. Irvine & P. C. Kyllonen (Eds.), *Item generation for test development* (pp. 199–217). Lawrence Erlbaum Associates.

Bejar, I. I. (2002b). Adaptive generative testing: From conception to implementation. In S. Irvine & P. Kyllonen (Eds.), *Item generation for test development* (pp. 199–218). Lawrence Erlbaum Associates.

Bejar, I. I. (2011). A validity-based approach to quality-control and assurance of automated scoring. *Assessment in Education: Principles, Policy & Practice*, *18*(3), 319–341.

Bejar, I. I. (2012). Rater cognition: Implications for validity. *Educational Measurement: Issues and Practice*, *31*(3), 2–9.

Bejar, I. I. (2013, April). *Gaming a scoring engine: Lexical and discourse level construct-irrelevant response strategies in the assessment of writing*. Paper presented at the meeting of the National Council on Measurement in Education.

Bejar, I. I. (2017). Threats to score meaning in automated scoring. In K. Ercikan & J. W. Pellegrino (Eds.), *Validation of score meaning for the next generation of assessments* (pp. 75–84). Routledge.

Bennett, R., & Zhang, M. (2016). Validity and automated scoring. In F. Drasgow (Ed.), *Technology and testing* (pp. 142–173). Routledge.

Bernstein, J., Cheng, J., Balogh, J., & Downey, R. (2020). Artificial intelligence for scoring oral reding fluency. In H. Jiao & R. Lissitz (Eds.), *Applications of artificial intelligence to assessment* (pp. 51–76). Information Age Publishing.

Bridgeman, B., Trapani, C., & Attali, Y. (2009, April). *Considering fairness and validity in evaluating automated scoring*. Paper Presented at the Meeting of the National Council on Measurement in Education. Sand Diego, CA.

Chen, J., Fife, J. H., Bejar, I. I., & Rupp, A. A. (2016). *Building e-rater scoring models using machine learning methods (RR-16-04)*. Educational Testing Service.

Clauser, B. E., Kane, M. T., & Swanson, D. B. (2002). Validity issues for performance-based tests scored with computer-automated scoring systems. *Applied Measurement in Education*, *15*(4), 413–432.

Clauser, B. E., Margolis, M. J., & Clauser, J. C. (2016). Issues in simulation-based assessment. In F. Drasgow (Ed.), *Technology and testing* (pp. 49–78). Routledge.

Cole, B. S., Lima-Walton, E., Brunnert, K., Vesey, W. B., & Raha, K. (2020). Taming the firehose: Unsupervised machine learning for syntactic partitioning of large volumes of automatically generated items to assist automated test assembly. *Journal of Applied Testing Technology, 2*(1), 1–11.

Cronbach, L. L. (1976). Equity in selection – where psychometrics and political philosophy meet. *Journal of Educational Measurement*, *13*(1), 31–41.

Cronbach, L. J. (1988). Five perspectives on validity argument. In H. Wainer & H. Braun (Eds.), *Test validity* (pp. 3–17). Lawrence Erlbaum Associates.

Drasgow, F., Luecht, R. M., & Bennett, R. (2006). Technology and testing. In R. L. Brennan (Ed.), *Educational measurement* (4th ed., pp. 471–516). American Council on Education.

Foltz, P. W. (2020). Practical considerations for using AI models in automated scoring of writing. In H. Jiao & R. W. Lissitz (Eds.), *Applications of artificial intelligence to assessment*. Information Age Publishing.

Foltz, P. W., Rosenstein, M., & Lochbaum, K. E. (2013). *Improving performance of automated scoring through detection of outliers and understanding model instabilities*. Paper presented at the annual conference of the National Council on Measurement in Education.

Friedler, S. A., Scheiegger, C., Venkatasubramanian, S., Choudhary, S., Hamilton, E. P., & Roth, D. (2018). A comparative study of fairness-enhancing interventions in machine learning. In *ACM conference on fairness, accountability and transparency (FAT*)*. ACM. http://arxiv.org/abs/1802.04422

Gao, Y., Bing, L., Chen, W., Lyu, M. R., & King, I. (2018). Difficulty controllable generation of reading comprehension questions. *arXiv*. https://arxiv.org/abs/1807.03586

Ha, L. A., & Yaneva, V. (2018). *Automatic distractor suggestion for multiple-choice tests using concept embeddings and information retrieval* (pp. 389–398). Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications.

Hood, S. (1998). Introduction and overview: Assessment in the context of culture and pedagogy: A collaborative effort, a meaningful goal. *The Journal of Negro Education*, *67*(3), 184–186.

Kane, M. T. (1992). An argument-based approach to validity. *Psychological Bulletin*, *112*(3), 527–535.

Kane, M. T. (2006). Validation. In R. L. Brennan (Ed). *Educational measurement* (4th ed., pp. 17–64). American Council on Education, Praeger.

Kane, M. T. (2010). Validity and fairness. *Language Testing*, *27*(2), 177–182.

Kane, M. T. (2013). Validating the interpretations and uses of test scores. *Journal of Educational Measurement*, *50*(1), 1–73.

Kolen, M. J. (2011). *Comparability issues associated with assessment for the common core state standards*. Paper presented at the NCME Invited Symposium: Will Common Core State Standards and Assessment Consortia Ensure Comparability of Assessment Results Across Students, Schools, Districts, and States.

Kurdi, G., Leo, J., Parsia, B., Sattler, U., & Al-Emari, S. (2020). A systematic review of automatic question generation for educational purposes. *International Journal of Artificial Intelligence in Education*, *30*, 121–204.

Lane, S. (2017). The value of response process studies to evaluate score meaning. In K. Ercikan & J. W. Pellegrino (Eds.), *Validating score meaning for the next generation of assessments* (pp. 138–144). Routledge.

Lane, S. (2022). *Interpretation/use and validity argument for AIG using a task-based approach*. Springer.

Lane, S., & DePascale, C. (2016). Psychometric considerations for performance-based assessments and student learning objectives. In H. Braun (Ed.), *Meeting the challenges to measurement in an era of accountability* (pp. 77–106). Routledge.

Lane, S., & Marion, S. (forthcoming). Validity and validation. In L. L. Cook & M. J. Pitoniak (Eds.), *Educational measurement* (5th ed.). Oxford University Press.

Lane, S., & Perez, A. (2023). Validity. In R. Tierney, F. Rizvi, K. Ercikan, & G. Smith (Eds.), *International encyclopedia of education* (4th ed., pp. 35–44). Elsevier.

Lane, S., & Stone, C. A. (2006). Performance assessments. In B. Brennan (Ed.), *Educational measurement* (pp. 387–432). American Council on Education, Praeger.

Luecht, R., & Burke, M. (2020). Reconceptualizing items: From clones and automatic item generation to task model families. In R. Lissitz & H. Jiao (Eds.), *Applications of artificial intelligence to assessment*. Information Age Publishing.

Mao, L., Liu, O., Roohr, K., Belur, V., Mulholland, M., Lee, H., & Pallant, A. (2018). Validation of automated scoring for a formative assessment that employs scientific argumentation. *Educational Assessment*, *23*(2), 121–138. https://doi.org/10.1080/10627197.2018.1427570

Messick, S. (1989). Validity. In R. L. Linn (Ed.), *Educational measurement* (3rd ed., pp. 13–104). American Council on Education and Macmillan.

Messick, S. (1994). The interplay of evidence and consequences in the validation of performance assessments. *Educational Researcher*, *23*(2), 13–23.

Mislevy, R. J. (2018). *Sociocognitive foundations of educational measurement*. Routledge.

Mislevy, R. J., Behrens, J. T., Dicerbo, K. E., & Levy, R. (2012). Design and discovery in educational assessment: Evidence-centered design, psychometrics, and educational data mining. *Journal of Educational Data Mining*, *4*(1), 1–38.

Mislevy, R. J., Corrigan, S., Oranje, A., DiCerbo, K., Bauer, M. I., von Davier, A., & John, M. (2016). Psychometrics and game-based assessment. In F. Drasgow (Ed.), *Technology and testing* (pp. 23–48). Routledge.

Mislevy, R. J., Steinberg, L. S., & Almond, R. G. (2003). On the structure of educational assessments. *Measurement: Interdisciplinary Research and Perspectives*, *1*(1), 3–62.

Oliveri, M. E., Lawless, R., & Mislevy, R. J. (2019). Using evidence-centered design to support the development of culturally and linguistically sensitive collaborative problem-solving assessments. *International Journal of Testing*, *19*(3), 270–300.

Powers, D. E., Burstein, J. C., Chodorow, M., Fowles, M. E., & Kukich, K. (2002). Stumping e-rater: Challenging the validity of automated essay scoring. *Computers in Human Behavior, 18*(2), 103–134.

Powers, E. E., Burstein, J. C., Chodorow, M. S., Fowles, M. E., & Kukich, K. (1998). *Qualifying essay readers for an online scoring network (OSN) (RR 98-20)*. ETS.

Quellmalz, E. S., Silberglitt, M. D., & Timms, M. J. (2011). *How can simulations be components of balanced state science assessments?* WestEd.

Rosen, Y. (2017). Assessing students in human-to-agent settings to inform collaborative problem-solving. *Journal of Educational Measurement*, *54*(1), 36–53.

Rupp, A. A. (2018). Designing, evaluation, and deploying scoring systems with validity in mind: Methodological design decisions. *Applied Measurement in Education*, *31*(3), 191–214.

Scoular, C., Care, E., & Hesse, F. (2017). Designs for operationalizing collaborative problem solving for automated assessment. *Journal of Educational Measurement*, *54*(1), 12–35.

Suresh, H., & Guttag, J. (2021, Summer). Understanding potential sources of harm throughout the machine learning life cycle. *MIT Case Studies in Social and Ethical Responsibilities of Computing*. https://doi.org/10.21428/2c646de5.c16a07bb

Trumbull, E., & Nelson-Barber, S. (2019). The ongoing quest for culturally responsive assessment for Indigenous students in the U.S. *Frontiers in Education*. https://doi.org/10.3389/feduc.2019.00040

Vendlinski, T. P., Baker, E. L., & Niemi, D. (2008). *Templates and objects in authoring problem-solving assessments*. CRESST Tech. Report. No. 735. University of California, National Center for Research on Evaluation, Standards, and Student Testing (CRESST).

von Davier, A. A. (2017). Computational psychometrics in support of collaborative educational assessments. *Journal of Educational Measurement*, *54*(1), 3–11. https://doi.org/10.1111/jedm.12129

von Davier, M. (2018). Automated item generations with recurrent neural networks. *Psychometrika*, *83*(4), 834–857.

Wang, Z., Zechner, K., & Sun, Y. (2018). Monitoring the performance of human and automated scores for spoken responses. *Language Testing*, *35*(1), 101–120.

Williamson, D. M., Bejar, I., & Mislevy, R. J. (2006). Automated scoring of complex tasks in computer-based testing: An introduction. In D. M. Williamson, R. J. Mislevy, & I. Behar (Eds.), *Automated scoring of complex tasks in computer-based testing* (pp. 146–173). Routledge.

Williamson, D. M., Xi, X., & Breyer, F. B. (2012). A framework for evaluation and use of automated scoring. *Educational Measurement: Issues and Practice*, *31*(1), 2–13.

9

Evaluating Fairness of Automated Scoring in Educational Measurement

Matthew S. Johnson and Daniel F. McCaffrey

1. Background

For over 50 years, computer-based evaluations of written responses have been used to provide scores or feedback to students and test-takers. These methods have evolved and become more sophisticated as natural language processing (NLP) has evolved. Currently, these methods are used in many testing situations to produce scores. For example, tens of millions of responses from elementary and secondary students in the United States are scored using NLP-based automated scoring, and some states moved to having all student responses from their elementary and secondary school testing programs scored by such methods (Ohio Department of Education, 2018). They are also used in large-scale international assessment surveys such as the Programme for International Student Assessment (PISA) and the Programme for the International Assessment of Adult Competencies (PIAAC), and they are being explored for the National Assessment of Educational Progress (NAEP). Moreover, each year, testing companies such as Cambium Learning Group, ETS, GMAC, Measurement Incorporated, and Pearson use NLP-based automated methods to score millions of responses from high-stakes tests such as the GMAT, GRE, TOEFL, the Duolingo English Test, the Pearson Test of English, and the Pearson Test of English Academic. According to the standards for educational testing, scores must be *fair*. Consequently, given the increased use of *automated scores* (AS), it is essential that the fields of educational and psychological measurement have methods to evaluate the fairness of the scores produced with NLP and artificial intelligence and to help ensure that any reported scores are fair to all test-takers. This chapter presents methods to detect one type of unfairness, which we call differential prediction bias. This occurs when the predicted scores generated from a specific AI scoring algorithm are lower (or higher) than what would be expected for a specified subgroup (e.g., racial group), given their performance on the item as defined by the human rating true score (i.e., the expected value of the human ratings across all potential ratings).

DOI: 10.4324/9781003278658-12
This chapter has been made available under a CC-BY-NC-ND 4.0 license.

1.1 Review of the Use of AI in Assessment

The use of computers for scoring student responses traces back to (Page, 1968, 1966). Page (1968) selected 30 variables that a computer could extract from typed responses, such as average sentence length, the number of words in the essay, and the number of periods, commas, and other punctuation marks. The features were then used as predictors in a multiple linear regression where the outcome variable was the essay's human assigned score; the 30 predictors were able to explain about 50% of the variation in human scores for the dataset he examined.

In the past 50 years, the use of AI and NLP for automated scoring has evolved in two directions: the sophistication of the features used for prediction, and the use of more powerful machine learning algorithms for prediction.

As an example of the types of input features used in the prediction of human scores today, consider ETS's e-rater system described in Attali and Burstein (2006). As defined, the system uses 10 NLP features, including rates of error (errors per word) in grammar, usage, mechanics, and style; features related to organization and development; two features related to lexical complexity (vocabulary sophistication and average word length); and two measures related to how similar a student's essay is compared to the population of essays at different score levels. The e-rater scores are the predicted values of a multiple linear regression of the human ratings on the NLP features.

Increased computer processing speed and the development of new AI algorithms has allowed for the inclusion of much larger sets of features in the prediction algorithms. Machine learning methods like support vector classifiers, machines, and regressions (SVC, SVM, SVR; Vapnik, 2000); the LASSO; and kernel-based methods (Hastie et al., 2009) have allowed for the inclusion of features such as character and word *n-grams* (indicators of sequences of *n* characters or words; see Madnani et al., 2017, and the references therein). For a given sample of examinees, the number of observed n-grams is often much larger than the sample size itself.

No matter the sophistication of the machine learning or NLP approaches used in the automated scoring algorithm, it is important to ensure that resulting automated scores produced are fair to all test-takers. Fairness in assessment is typically defined as the assurance that all assessment participants have an equal opportunity to demonstrate their knowledge and skills and requires that the assessment be culturally sensitive, free from bias against any group, and accessible to special populations. After reviewing selected concepts from psychometrics and fairness as defined in testing in the next sections, we introduce our conceptualization of fairness as it relates to freedom of bias against groups in the section 'Definitions of Fair AI-scores in Assessment'.

1.2 Fairness in Testing

Fairness has long been a central concern of assessment. Commonly, fairness is considered in terms of validity as noted in the *Standards for Educational and Psychological Testing*: 'Fairness is fundamentally a validity issue and requires attention throughout all stages of test development and use' (p. 49). And, 'In summary, this chapter interprets fairness as responsiveness to individual characteristics and testing context so that test scores will yield valid interpretation for intended uses' (p. 50). The meaning of validity (and fairness as well) is open to philosophical and theoretical debate and has evolved over time (Markus & Borsboom, 2013). However, a widely accepted conceptualization of validity is that a test measures what it intends to measure and supports the proposed interpretations and uses (Messick, 1989). In addition, validity is judged by the degree of the evidence that demonstrates these requirements. The evidence is presented through the *validity argument* (Kane, 2006). All aspects of the testing process, from initial conceptualization and specification through development, administration, scoring, score reporting, uses, and consequences, are part of creating the evidence for the validity argument.

If fairness is essentially applying the concept of validity to all individuals, then fairness depends on evidence that the test supports its intended interpretations and uses for all individuals. Fairness needs to be part of all aspects of the testing process. Beyond testing, social and legal fairness is often considered in terms of the equitable treatment of individuals across groups such as people of different racial or ethnic backgrounds, biological sex or gender, age, disability status, or others (Camilli, 2006). Historically, this has translated in testing to ensuring an equitable treatment of test-takers through the assessment process and demonstrating a lack of measurement bias. Of particular concern has been that test-takers of equal ability or proficiency from different groups have similar score distributions on an item or test, or that the test score has not been differentially predictive of an intended outcome for test-takers of different groups (AERA et al., 2014; Camilli, 2006). In particular, the statistical methodological literature and the empirical studies on fairness have centered on testing for differential functioning of items or test scores for test-takers of equal ability from different groups. In addition to such empirical tests of functioning of test items and scores, ensuring fairness has also included such things as experts reviewing the content of test items for potential bias or differential impact on test-takers of different groups, standardization of procedures, and the development of adaptations for test-takers with disabilities.

In more recent years, increasingly focus has been on connections between fairness and justice and whether students in different groups have equal opportunity to learn. As Gipps and Stobart (2004) state, 'The notion of the standardised test as a way of offering impartial assessment is a powerful one, though if equality of educational opportunity does not precede the test, then the "fairness" of this approach is called into question' (p. 31). It is simply not enough to demonstrate that the test or its items function similarly for test-takers of equal ability from different groups. Equal opportunity must also be considered. Increasingly, fairness of measurement incorporates impact on social justice and equity in the uses and consequences of assessment in addition to traditional concerns with equitable treatment and measurement bias. However, as will be discussed, in the AI literature, fairness has generally been conceived in terms of bias (or a lack of it), which can be tested through statistical analysis.

1.3 Brief Review of Key Psychometrics Concepts

To introduce our proposed methods for evaluating fairness and the procedures to remedy unfair AI-scoring algorithms, we briefly review classical true score models and item response theory models in the context of AI scoring.

Consider the problem of scoring a constructed response such as a written essay. We assume there is a random sub-sample of assessment participants whose responses have each been scored by two randomly selected human raters according to a predefined rubric, another sample scored by one randomly selected human rater, and a final sub-sample not scored by any human raters. In some assessment programs, there may be a sample with more than two raters, but a minimal requirement is that we have a sub-sample with at least two raters to be able to estimate some of the quantities needed for our proposed methods.

Let H_{ij} denote the score assigned to the response from participant i by the first $(j=1)$ or second $(j=2)$ of two randomly selected raters. The true score for participant i is defined as the expected value of the assigned score H_{ij} averaged over the population of potential raters, and it is denoted as $T_i = E\left[H_{ij}\right]$. The true score measurement model assumes that measurement errors are additive so that between $H_{ij} = T_i + \epsilon_{ij}$. To assess the quality of the human score H_{ij} as a predictor of the true score target T_i, the reliability of the human score is typically reported. The reliability of a single human score H_{ij} is the squared correlation between the human score and the true score,

$$\text{Reliability} = \left(\text{cor}\left(H_{ij}, T_i\right)\right)^2 = \frac{\sigma_T^2}{\sigma_T^2 + \sigma_\epsilon^2},$$

where $\sigma_T^2 = \text{Var}(T)$ and $\sigma_\epsilon^2 = \text{Var}(\epsilon)$. Given a sub-sample of data with multiple human ratings, these quantities can be estimated from the data using any of number of methods for estimating variance component methods (e.g., ANOVA, REML; Searle et al., 1992).

When multiple items make up a test, we often assume a higher-level latent variable that is measured by the item-level true scores. When we can assume that the items making up a test are parallel, we can define the assessment-level true score θ_i as the expected value of the item-level true scores T_i over the population of items; in other words, $\theta_i = E\left[T_{ik}\right]$, where T_{ik} is an item-level true score of item k and the expectation is taken over the population of items. In this case, we can write:

$$T_{ik} = \theta_i + \delta_{ik},$$

where δ_{ik} is an error uncorrelated with θ_i. Combining the item-level and assessment-level true score models, we have the following combined model:

$$H_{ikj} = \theta_i + \delta_{ik} + \epsilon_{ikj},$$

where H_{ikj} and ϵ_{ikj} are the observed ratings and errors associated with item k and rater j. This simple generalizability theory model (Cronbach et al., 1963) is sometimes expanded by including parameters/facets for item difficulty and/or rater biases.

1.4 Methods for Evaluating Unfairness of Observed Scores in Testing

In their paper '50 Years of Test (Un)fairness: Lessons for Machine Learning', Hutchinson and Mitchell (2019) note that the field of educational assessment has a long history of developing methods for evaluating fairness. Much of that research has centered around methods for detecting biased test items, items that are more difficult for some subgroup than would be expected given their overall performance on the test. This type of unfairness is described by conditional independence assumptions.

Suppose, for example, that we have scores from multiple test items for each participant, denoted by Y_{ij} for participant i and item j. For the items to be fair, we would assume that the item scores are conditionally independent of subgroup membership, which we will denote as G_i, given the latent variable θ_i. Using standard notation, we denote this conditional independence by $Y_{ij} \perp G_i \mid \theta_i$. That is, a test item is considered fair only if all the association between the item scores and subgroup membership is described through their shared association with the target variable we are trying to measure, θ. Items that violate this conditional independence assumptions are said to exhibit *differential item functioning* (DIF; Holland & Wainer, 1993). This definition of fairness is related to the separation fairness criteria put forth in the machine learning literature (Barocas et al., 2019), which we will describe in greater detail.

A common method for evaluating DIF is the Mantel-Haenszel DIF procedure (MHDIF; Holland & Thayer, 1988). The MHDIF procedure stratifies the assessment participants by their total score on the test, $S_i = \sum_{j=1}^{J} Y_{ij}$, and produces 2×2 tables within each stratum that classify individuals according to their subgroup attribute G (e.g., Male vs. Female) and whether or not the item under study is correct. The Mantel-Haenszel test then tests whether the odds ratio is equal to 1.0 in all strata.

2. Evaluating Fairness in Automated Scores

As with testing, the focus on fairness in the AI literature is on biases of the type that similar individuals from different groups are treated differently by the AI algorithms or systems. Also, as with testing, evidence of a lack of such bias is considered essential for an AI application to be fair. Consequently, we focus this chapter on methods for testing for bias in AI scores. However, multiple sources (Educational Testing Service, 2021; International Test Commission and Association of Test Publishers, 2022; Williamson et al., 2012) recommend fairness in AI scoring not be limited to such statistical tests. It should also include evaluation of the processes for developing the AI algorithms and tuning models, the samples used for model building, explorations of the NLP features used in scoring, the entire test and item development process, the human raters, how the backgrounds of the test-takers might affect their response in ways that may interact with the algorithms, and consequences for test-takers.

In order to develop methods for evaluating fairness (or bias) in automated scores, we must first start with a formal definition of fair AI scores. The section 'Definitions of Fair AI Scores in Assessment' proposes definitions of fair AI scores. The sections 'Existing Methods for Evaluating Fairness in Automated Scores' and 'Limitations of Existing Methods for Evaluating Fairness of Automated Scores' review and evaluate existing methods in relation to these definitions.

2.1 Definitions of Fair AI Scores in Assessment

Barocas et al. (2019) note that most fairness criteria proposed in the AI literature are based on properties of the joint distribution of three (sets of) variables: the target variable we are trying to predict, the sensitive attribute we are attempting to ensure fairness for, and the predicted score generated by AI. In what follows, we denote the predicted score by M and the sensitive attribute or subgroup indicator variable by G. In assessment applications, the target variable for an item is the true score $T_i = E\left[H_{ij}\right]$, the unobservable mean of the score assigned by a randomly selected human rater. Friedler et al. (2016) recognized the importance of considering latent constructs, like our true score T, in evaluating the fairness of algorithmically generated scores, and noted that definitions of fairness should explicitly state assumptions about the relationships between constructs and observations.

Building off standard psychometric assumptions, we assume conditional independence of the multiple observed human scores H_{ij} given the latent true score T, i.e., $H_{ij_1} \perp H_{ij_2} \mid T_i$ for all $j_1 \neq j_2$, and $H_{ij} \perp G_i \mid T_i$ for all j. If raters are randomly assigned, then $H_{ij} \perp G_i \mid T_i$ for all j is necessarily true since G is a characteristic and cannot be associated with a randomly assigned rating other than through the true score. We further assume that the AI score M is a function of observable input features $\mathbf{X}$, $M \equiv \hat{t}(\mathbf{X})$, meant to minimize some optimization criteria such as the mean squared error $E\left[\left(T - \hat{t}(\mathbf{X})\right)^2\right]$, in which case $M \equiv \hat{t}(\mathbf{X}) = E\left[T \mid \mathbf{X}\right] = E\left[H \mid \mathbf{X}\right]$, provided $\breve{t}$ is sufficiently flexible and properly specified to capture the functional form of $E\left[T \mid \mathbf{X}\right]$. We define fairness as restrictions on the conditional relationships among the variables in the joint distribution of (T, G, M). Barocas et al. (2019) summarizes three different fairness definitions that are typically considered. They are:

- *Independence* (Disparate impact; $M \perp G$) : Predicted score M is independent of subgroups G.

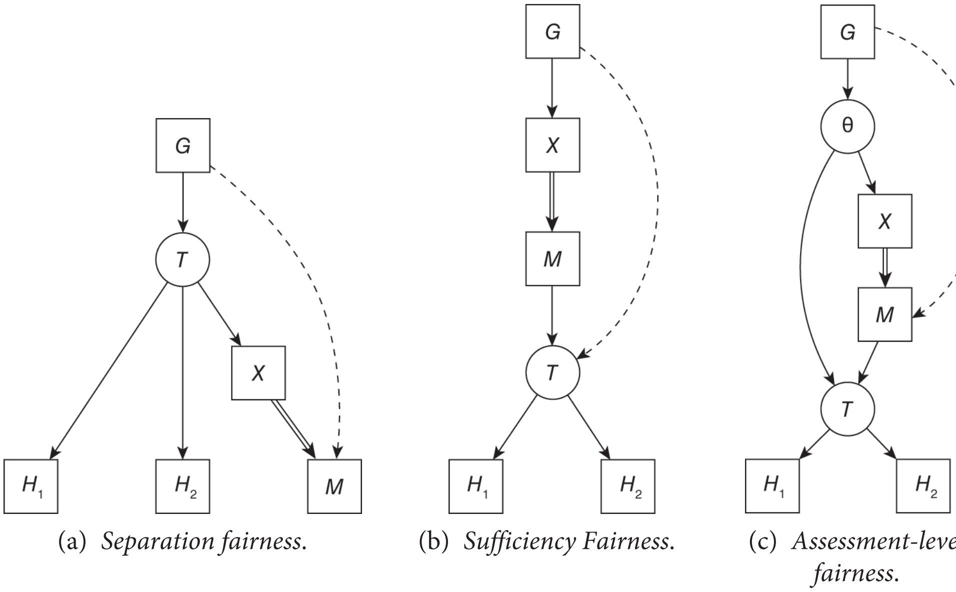
- *Separation* $(M \perp G | T)$: Predicted score M is conditionally independent of subgroups G given the true score T.
- *Sufficiency* $(T \perp G | M)$: True score T is conditionally independent of subgroups G given the predicted score M.

We believe that the independence condition is too strict for assessment purposes. If there are true differences in the construct being assessed (e.g., writing ability), then forcing independence on the predicted scores M would be unfair to subgroups with higher levels of the construct. Separation and sufficiency, however, are reasonable definitions of fairness in the context of assessments.

The concept of fairness by separation is displayed in the graph in Figure 9.1a.

The conditional independence assumptions used to define differential item functioning in the section 'Methods for Evaluating Unfairness of Observed Scores in Testing' are analogous to fairness by separation with θ, the target of measurement for the test, as the conditioning variable rather than an item-specific construct. As with differential item functioning, the automated score M would be considered unfair if it was not conditionally independent of the group variable G given the true score, i.e., if the dashed line was present in the generating model. If the dashed path was present in the graph, then some of the *rubric-irrelevant* variance in the machine scores is associated with the subgroup variable G. Also, note that by assumption, the human ratings meet the criteria for fairness by separation.

Sufficiency fairness is depicted by the directed graph in Figure 9.1b. The features $\mathbf{X}$ used to produce the predictions encapsulate all the information about differences in subgroup performance on the assessment item. None of the variation in the residual $T - E[T | M]$ is associated



(a) *Separation fairness.* (b) *Sufficiency Fairness.* (c) *Assessment-level fairness.*

Figure 9.1 Graphical models representing different types of (un)fairness in the AI rating process. G is a group indicator, θ is the assessment-level true score, T is the true human item score, H_1 and H_2 are observed human scores, $\mathbf{X}$ is a vector of features used to produce the predicted score M. The presence of the dashed paths indicates situations where the machine scores are unfair.

with subgroups, and therefore the machine scores are fair. If $M = E\left[T \mid \mathbf{X}\right]$, as is the case if M minimizes mean squared error and T meets the necessary assumption, then $M = E\left[T \mid M\right]$ and none of the variation $T - M$ is associated with subgroups.

The graphs in Figures 9.1a and 9.1b are useful for defining fairness when assessments are based on a single response – for example, a single written essay. When a final assessment score is based on multiple items, ensuring separation or sufficiency fairness at the item level would be sufficient for ensuring fairness at the test level. However, they are not necessary for fairness at the test level.

Figure 9.1c demonstrates the difference between the item and test levels. As discussed in our brief introduction to generalizability in the section 'Brief Review of Key Psychometrics Concepts', θ represents the overall writing ability we are attempting to measure with the item under consideration. Even though the model depicted by Figure 9.1c would violate the separation and sufficiency definitions presented earlier at the item level, the machine scores would still be fair at the assessment level as long at the dashed path is not present. The sort of process depicted by Figure 9.1c might occur when the features used for prediction only partially account for what the human raters are judging. The remaining variation in T conditional on M might still be related to overall writing ability, and thus subgroup membership G, but is an appropriate measure of θ since there is no direct path from G to T or M.

2.2 Existing Methods for Evaluating Fairness in Automated Scores

The evaluation of automated scores in assessment has typically compared how the scores perform relative to the observed human scores. The typical metrics examined are all summaries of the confusion matrix, a contingency table recording the joint distribution of either two human ratings, or one human rating and the automated score, i.e., a two-way table where the element n_{ij} in row i and column j denotes the number of participants assigned a score of i by the human rater and a score of j by the automated scoring algorithm.

In addition to standard summaries of the distributions of the scores, like means and standard deviations, AI-scores are often evaluated with the following statistics:

- Percent exact agreement: $\text{EXACT} = \frac{1}{n_{..}} \sum_j n_{jj} \times 100\%$.
- Percent adjacent agreement: $\text{ADJACENT} = \frac{1}{n_{..}} \sum_j \sum_{\{k:|k-j| \le 1\}} n_{jk}$.
- Cohen's quadratic weighted kappa: $\text{QWK} = 1 - \frac{\sum_j \sum_k (j-k)^2 n_{jk} / n_{..}}{\sum_j \sum_k (j-k)^2 n_{j.} n_{k.} / n_{..}^2}$.
- Pearson product-moment correlation.

For example, the PARCC assessment technical report (Pearson, 2017) describes the following criteria when evaluating AI scores. Exact agreement between human and AI scores should be at least 65%, and the difference between human-human and human-AI agreement should be less than 5.25%. Quadratic weighted kappa between human and AI scores should be at least .40, and the difference between human-human and human-AI QWK should be less than .10. Pearson correlation between human and AI scores should be at least .70, and the difference between human-human and human-AI correlations should be less than .10.

While the Pearson correlation can be used to evaluate continuous automated scores, the other statistics described earlier assume that the machine scores are discrete variables. This often leads analysts to round or otherwise discretize continuous automated score. Discretizing the continuous automated scores can reduce some measures of agreement, and statistics that use the continuous score are preferable. Although there are no direct analogs for percent agreement or adjacent agreement, Haberman (2019) provides a generalization of QWK for continuous measures. When the mean and variance of M equal the mean and variance of H, then the QWK equals their correlation.

These criteria describe how well the AI score serves as a predictor of a random human score H in comparison to a second human score. They do not evaluate fairness. To evaluate fairness of AI-scores, Williamson et al. (2012) suggested four checks: (1) checking that the *standardized mean score differences* between human raters and AI scores by subgroup are small (they recommend that the absolute value be less than .10); (2) examining differences in the association between human raters and AI scores across subgroups; (3) estimating the reliability of test scores by group; and (4) checking for differences across subgroups in the prediction ability of an external criterion by AI scores. Although any of these checks can identify difference in the AI scores across subgroups and possibly support improving the models, only the standardized mean differences check is a direct test for bias. It is the only check for which the authors give an explicit numerical target. In our experience, it is the most commonly used check for fairness. For example, it was the only check of fairness required by the recent competition for AI-scoring constructed response items for NAEP.[1]

For a given subgroup g, the standardized mean score difference equals

$$\mathrm{SMD}_g = \frac{\bar{M}_g - \bar{H}_g}{s_g},$$

where s_g is an appropriate standard deviation. Williamson et al. (2012) suggested the pooled standard deviation $s_g = \sqrt{\frac{s_{hg}^2}{2} + \frac{s_{mg}^2}{2}}$, where s_{hg} and s_{mg} denote the standard deviation of the human and machine scores, respectively. Alternatively, when the scores are evaluated on a human-score scale, as is the case when AI scores are used as a substitute for human scores, using the human standard deviation in the denominator (i.e., $s_g = s_{hg}$) may be more appropriate.

In one study, Bridgeman et al. (2012) used the standardized mean differences and found that Chinese examinees had relatively higher AI scores than human scores, while Arabic- and Hindi-speaking examinees received relatively lower AI scores.

When the human scores and the AI scores are on different scales (i.e., different standard deviations and/or means), then a more appropriate way to evaluate subgroup differences would be to examine the differences in mean standardized scores. That is, within each set of scores, the scores should first be standardized. For example, for individual i within sub-group g, the standardized human score is

$$Z_{hgi} = \frac{H_{gi} - \bar{H}_{\cdot\cdot}}{s_{h\cdot}},$$

where $\bar{H}_{\cdot\cdot}$ is the overall grand mean of human scores, and $s_{h\cdot}$ is the standard deviation of all human scores. Similarly, we denote the standardized AI score for individual i in group g by Z_{mgi}. Then the difference in standardized mean scores for subgroup g is simply $\mathrm{DSM}_g = \bar{Z}_{mg\cdot} - \bar{Z}_{hg\cdot}$.

Another way in which fairness has been evaluated in the assessment literature is through the conditional distribution of the features $\mathbf{X}$ used in the predictive model. Penfield (2016) and Zhang et al. (2017) have suggested generalizing the idea of differential item functioning (see the section 'Methods for Evaluating Unfairness of Observed Scores in Testing') to the features $\mathbf{X}$. The procedure, which they call differential feature functioning (DFF), evaluates conditional independence of each feature X_k and group membership G conditional on either the machine score M or the observed human score H. That is, DFF evaluates the assumptions $X_k \perp G \mid M$ and/or $X_k \perp G \mid H$ for all features k. As demonstrated in an example later in this chapter, DFF can provide useful information for diagnosing possible sources of bias in AI scores. However, DFF does not necessarily indicate bias. There could be true differences in the skills or abilities across subgroups measured by a feature and not fully explained by the overall score or human rating.

2.3 Limitations of Existing Methods for Evaluating Fairness of Automated Scores

Although each of the existing methods for testing the fairness of automated scores can detect some violations of fairness, each also has notable limitations, which are described in this section.

The standardized mean difference (SMD), proposed by Williamson et al. (2012), is an appropriate method to evaluate sufficiency fairness described by the graph in Figure 9.1b, provided $M = E\left[T \mid \mathbf{X}\right]$. However, it is possible that an AI-scoring procedure produces scores that satisfy separation fairness (Figure 9.1a) but have nonzero standardized mean differences. For example, suppose we have two groups designated by $G = 0$ and $G = 1$, and that we have differences at the item level such that $E\left[T \mid G = g\right] = \mu_g$ with $\mu_0 \neq \mu_1$ and $E\left[T\right] = 0$. Further, assume that $E\left[H_{ij} \mid T_i\right] = T_i$, T_i and M_i are linearly related, M_i is the best linear predictor of T_i, and $E\left[M_i \mid T_i, G\right] = E\left[M_i \mid T_i\right]$. The last assumption implies M_i meets the criteria for fairness by separation. Because M_i is the best linear predictor, $E\left[M_i\right] = E\left[T_i\right]$, $v_M^2 = \mathrm{Var}\left(M_i\right) = \rho_{MT}^2 v_T^2$, where $\rho_{MT}^2 = \mathrm{cor}\left(M_i, T_i\right)$ and $v_T^2 = \mathrm{Var}\left(T_i\right)$, and $E\left[M_i \mid T_i\right] = \rho_{MT} \frac{v_M}{v_T} T_i = \rho_{MT}^2 T_i$. The numerator of the standardized mean difference would be an estimate of

$$s_g \mathrm{SMD}_g = E\left[M - H \mid G = g\right] = E\left[\rho_{MT}^2 T - H \mid G = g\right] = \left(\rho_{MT}^2 - 1\right)\mu_g .$$

Sometimes the automated scores are scaled to have the same mean and variance as the human ratings. For example, if M is the maximum QWK linear regression predictor of H given a set of features, then the mean and variance of M equal the mean and variance of H. In this case,

$$s_g \mathrm{SMD}_g = \left(\rho_{MT}^2 \sqrt{\frac{v_H}{\rho_{MT}^2} v_T} - 1\right)\mu_g = \left(\frac{\rho_{MT}}{\rho_{HT}} - 1\right)\mu_g .$$

In this special case, the SMD will be zero if the correlation between M and T equals the correlation between H and T. Since the correlation between M and H, $\rho_{MH} = \rho_{MT}\rho_{HT}$ and correlation between two human ratings equal the square of the correlation between H and T, if the correlation between M and H equals the correlation between two raters, then $\rho_{MT} = \rho + HT$ and when the automated scores and T are linearly related and the automated scores are best linear predictor T rescaled to have the same mean and variance of H, then under separation fairness the SMD is zero. At the time that Williamson and colleagues (2012) recommended using SMD to test for fairness, ETS used linear regression for its automated scores and often rescaled the scores to have the same mean and variance as H. Implicit in their recommendation was the assumption of linearity between T and M.

The difference in standardized means (DSM) also is problematic. For the same example described in the previous paragraph, the mean standardized difference is

$$\mathrm{DSM}_g = E\left[\frac{M}{\sqrt{v_m}} - \frac{H}{\sqrt{v_h}} \mid G = g\right] = \left(\frac{\rho_{MT}^2}{\sqrt{v_m}} - \frac{1}{\sqrt{v_h}}\right)\mu_g \left(\frac{\rho_{MT}}{\sqrt{v_t}} - \frac{\rho_{HT}}{\sqrt{v_t}}\right)\mu_g ,$$

where v_h is the marginal variance of the human scores. Unless $v_x = v_h$ the difference in standardized means is nonzero. That is, unless the correlation of true score T with H equals the correlation of the true score with M, the difference in standardized means will be nonzero, even though, the scores meet the criteria for separation fairness.

The differential feature functioning procedure is potentially less problematic than the standardized mean difference (SMD) and differences in standardized means (DSM) approaches for evaluating fairness. However, it is not without its limitations. Consider a slight expansion of the equation described in the previous two examples, where we have two features for our predictive model with $X_{ij} = T_i + \delta_{gj} + \epsilon_j$ for features $j = 1,2$. The DFF procedure for examining the fairness of the second feature would examine $E\left[X_2 \mid M, G = g\right]$ for groups defined by g and compare them. If there is no DFF, then the differences should be zero. If M is based on least squares regression, $\boldsymbol{\epsilon}$ is bivariate normal, $\operatorname{Cov}(\epsilon_1, \epsilon_2) = 0$, $\operatorname{Var}(\epsilon_1) = \operatorname{Var}(\epsilon_2)$, then we are examining

$$Q_{g2s} = E\left[X_2 \mid X_1 + X_2 = s, G = g\right] = \mu_g + \delta_{g2} + \frac{1}{2}\left(s - 2\mu_g - \delta_{g1} - \delta_{g2}\right) = \frac{1}{2}\left(s + \delta_{g2} - \delta_{g1}\right).$$

There are a couple of undesirable consequences of this result. First, if $\delta_{g2} = 0$ for all g, so that it does not have differential functioning across groups, but $\delta_{11} - \delta_{21} \neq 0$, then $Q_{12s} - Q_{22s} = \frac{1}{2}\left(\delta_{21} - \delta_{11}\right) \neq 0$, so the second feature would be considered as unfair. The second consequence of this result is the fact that if $\delta_{g2} = \delta_{g1} \neq 0$, i.e., the level of differential feature functioning is constant across features, then $Q_{12s} - Q_{22s} = 0$. In this case, the DFF method would fail to recognize the unfairness in the features. The alternative approach which conditions on observed human scores H instead of AI scores M suffers from the same limitations and introduces others.

3. Proposed Methods for Evaluating Fairness in AI Scores

Given the limitations of the existing methods for evaluating separation fairness of scores generated by artificial intelligence, we develop and evaluate new methods for this purpose. Specifically, we introduce two methods to evaluate the assumption of separation fairness: one based on structural equation modeling, and one based on errors-in-variables regression. In addition to methods for detection of unfairness, we present statistical methods based on constrained optimization and penalization to mitigate unfairness that may become evident in the AI scores. We will demonstrate all the methods using data from a large-scale educational assessment.

In the methods that we present in the coming sections, we start with the assumption that the human ratings are fair and themselves do not exhibit any differential item functioning. That is, we assume that the human ratings assigned to a participant are conditionally independent of subgroup membership given the true score T, $H_{ij} \perp G_i \mid T_i$. Therefore, it is imperative that fairness of human scorers is evaluated prior to carrying out the methods we propose for evaluating fairness of the AI scores.

3.1 Structural Equation Model Methods

The directed graph describing sufficiency fairness in Figure 9.1a can be viewed as a specific type of multi-group structural equation model. As such, the parameters associated with the model can be readily estimated using standard software for SEM. The full model assumes that the human ratings H_{gij} and automated scores M_{gi} for individual i in group $G = g$ follow the structural equation model

$$\begin{aligned} H_{gij} &= T_{gi} + \epsilon_{gij} \\ M_{gi} &= \lambda_g T_{gi} + \gamma_g + \zeta_{gi} \\ T_{gi} &= \mu_g + \delta_{gi}. \end{aligned}$$

Because we assume that rater errors are independent of group membership, so the $\mathrm{Var}\left(\epsilon_{gij}\right) = \sigma_\epsilon^2$ is constant across groups. In contrast, the residual variance for the true score T_{gi} and the automated score M_g, which we denote by σ_{Tg}^2 and σ_{Mg}^2 respectively, are allowed to vary across groups under the full model.

In terms of this structural equation model, the assumption of separation fairness is defined as constraints on three sets of parameters that are allowed to vary in the full model. Namely, the loading λ_g, the automated score intercept γ_g, and the automated score residual variances σ_{Mg}^2 must all be constant across the levels g of the grouping variable G in order for separation fairness to hold. Therefore, separation fairness can be evaluated by testing the following null and alternative hypotheses:

$$H_0 : \gamma_g = \gamma; \sigma_{Mg}^2 = \sigma_M^2; \lambda_g = \lambda, \text{for all } g \tag{1}$$

$$H_1 : \gamma_{g_1} \neq \gamma_{g_2} \text{or } \sigma_{Mg_1}^2 \neq \sigma_{Mg_2}^2 \text{or } \lambda_{g_1} \neq \lambda_{g_2}, \text{for some groups } g_1 \neq g_2 \tag{2}$$

When it is safe to assume that scores H and M are approximately jointly normally distributed, standard likelihood ratio tests for nested models can be calculated to test this null H_0 against the alternative H_1 and compared to a chi-squared distribution with $3(K-1)$ degrees of freedom to determine statistical significance.

If there is significant evidence to reject the null hypothesis of separation fairness, we recommend examining the magnitude of the unfairness for each group. The unfairness effect for group g is defined as the difference between the mean automated score for the group implied by the full model and the mean implied by the reduced null model. Under the full model, the mean automated score is

$$\lambda_g \mu_g + \gamma_g.$$

The mean under the reduced model is

$$\lambda \mu_g + \gamma.$$

Therefore, the separation unfairness effect for group g is the difference:

$$\Delta_g = \left(\lambda_g - \lambda\right)\mu_g + \left(\gamma_g - \gamma\right). \tag{3}$$

In the application in the section 'Application to Real Data', we estimate all of the parameters using the full model, which provides estimates of the group-specific parameters λ_g and γ_g. For the estimate of the common parameter values λ and γ, we take simple linear combinations of the group-specific estimates as follows:

$$\begin{aligned} \hat{\gamma} &= \sum_g p_g \hat{\gamma}_g \\ \hat{\lambda} &= \sum_g p_g \hat{\lambda}_g, \end{aligned}$$

where p_g is the proportion of the sample that belongs to group g.

3.2 Errors-in-Variables Regression

The separation fairness model in Figure 9.1a is a simple analysis of covariance (ANCOVA) model where the control variable T is unobservable. However, we do have a noisy proxy, the human rating, that we can use in an errors-in-variables regression (Culpepper & Aguinis, 2011; EIV; Fuller, 1980). The idea of EIV regression is that if we are able to directly observe the variable T, we could estimate the strength of the dashed path in Figure 9.1a relatively easily; we would simply regress the AI score M on T and $K-1$ dummy indicator variables for group membership, e.g.,

$$M_i = \beta_0 + \beta_1 \left(T_i - \bar{T}\right) + \sum_{g=2}^{K} \beta_g I\{G_i = g\} + \varepsilon_{gi}.$$

Separation unfairness could then be evaluated by examining the practical and statistical significance of the null hypothesis

$$H_0 : \beta_g = 0 \text{ for all } g \geq 2.$$

If the mean centered item true scores were observed directly, the least squares estimates of the regression coefficients $\boldsymbol{\beta}$ would equal

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{D}^\top \mathbf{D}\right)^{-1} \mathbf{D}^\top \mathbf{M}, \tag{4}$$

where $\mathbf{D}$ is the $n \times (K+1)$ design matrix $\mathbf{D} = \left[\mathbf{1} \,|\, \mathbf{T}^\star \,|\, \mathbf{U}\right]$, $\mathbf{1}$ is an n -vector of all 1's, and $\mathbf{M}$ is the vector of automated scores, $\mathbf{U}$ is an $n \times (K-1)$ matrix of dummy variables, and $\mathbf{T}^\star$ is the vector of mean centered true scores.

The problem is that T_i is not directly observable, and therefore $\mathbf{D}^\top \mathbf{D}$ and $\mathbf{D}^\top \mathbf{M}$ cannot be calculated and standard regression analysis is not possible. However, under the true score model that assumes that the human ratings satisfy

$$H_{ij} = T_i + \epsilon_{hij},$$

it is possible to consistently estimate both $\mathbf{D}^\top \mathbf{D}$ and $\mathbf{D}^\top \mathbf{M}$ by replacing the mean-centered vector of item true scores $\mathbf{T}^\star$ in $\mathbf{D}$ with $\mathbf{H}_1^\star$, the mean-centered vector of the first human ratings, and subtracting a term that depends on the reliability of the human ratings from the diagonal term corresponding to H_{i1} (Fuller, 1980). Specifically, let $\mathbf{D}_H = \left[\mathbf{1} \,|\, \mathbf{H}_1^\star \,|\, \mathbf{U}\right]$ and note:

$$\begin{aligned}
\mathbf{D}_H^\top \mathbf{D}_H &= \begin{pmatrix} n & 0 & \mathbf{1}^\top \mathbf{U} \\ 0 & \mathbf{H}_1^{\star\top} \mathbf{H}_1^\star & \mathbf{H}_1^{\star\top} \mathbf{U} \\ \mathbf{U}^\top \mathbf{1} & \mathbf{U}^\top \mathbf{H}_1^\star & \mathbf{U}^\top \mathbf{U} \end{pmatrix} \\
&= \begin{pmatrix} n & 0 & \mathbf{1}^\top \mathbf{U} \\ 0 & \mathbf{T}^{\star\top} \mathbf{T}^\star & \mathbf{T}^{\star\top} \mathbf{U} \\ \mathbf{U}^\top \mathbf{1} & \mathbf{U}^\top \mathbf{T}^\star & \mathbf{U}^\top \mathbf{U} \end{pmatrix} + \begin{pmatrix} 0 & 0 & \mathbf{0}^\top \\ 0 & \boldsymbol{\epsilon}_1^{\star\top} \boldsymbol{\epsilon}_1^\star & \mathbf{0}^\top \\ \mathbf{0} & \mathbf{0} & \mathbf{O} \end{pmatrix} + \begin{pmatrix} 0 & \mathbf{1}^\top \boldsymbol{\epsilon}_1^\star & \mathbf{0}^\top \\ \mathbf{1}^\top \boldsymbol{\epsilon}_1^\star & 2\mathbf{T}^{\star\top} \boldsymbol{\epsilon}^\star & \boldsymbol{\epsilon}^{\star\top} \mathbf{U} \\ \mathbf{0} & \mathbf{U}^\top \boldsymbol{\epsilon}^\star & \mathbf{O} \end{pmatrix}.
\end{aligned}$$

The first matrix in the last equation is the cross-product matrix $\mathbf{D}^\top \mathbf{D}$ that we need for least squares regression of the automated scores on the item true scores. The single nonzero term

$\boldsymbol{\epsilon}^{\star\top}\boldsymbol{\epsilon}$ in the second matrix has expected value equal to $(n-1)\sigma_{\epsilon}^{2}$, which can be estimated from our sample of responses that have multiple ratings. The expected value of the third matrix is the zero matrix. Therefore, if

$$\mathbf{V} = \begin{pmatrix} 0 & 0 & \mathbf{0}^{\top} \\ 0 & \frac{n-1}{n}\hat{\sigma}_{\epsilon}^{2} & \mathbf{0}^{\top} \\ \mathbf{0} & \mathbf{0} & \mathbf{O} \end{pmatrix},$$

we have

$$\frac{1}{n}\mathbf{D}_{H}^{\top}\mathbf{D}_{H} - \mathbf{V} \overset{n\to\infty}{\to} \frac{1}{n}\mathbf{D}^{\top}\mathbf{D} \text{ in probability.}$$

as long as $\widehat{\sigma^{2}}_{\epsilon}$ is a consistent estimator of σ_{ϵ}^{2}. Furthermore, it can be shown that

$$\frac{1}{n}\mathbf{D}_{H}^{\top}\mathbf{M} \overset{n\to\infty}{\to} \frac{1}{n}\mathbf{D}^{\top}\mathbf{M} \text{ in probability.}$$

It follows directly that the EIV regression coefficient vector found by replacing $\mathbf{D}$ with $\mathbf{D}_{H}$ in Equation (4),

$$\tilde{\boldsymbol{\beta}} = \left(\mathbf{D}_{H}^{\top}\mathbf{D}_{H} - n\mathbf{V}\right)^{-1}\mathbf{D}_{H}^{\top}\mathbf{M} \tag{5}$$

is a consistent estimator of the true regression coefficient vector $\boldsymbol{\beta}$ (Fuller, 1980).

One way to assess the fairness could be to estimate the variance among the coefficients for the groups, including zero for the holdout group, or to create a measure analogous to a partial eta squared. Such measures would be similar to the study of the variance of the residual group-specific expectations considered in Yao et al. (2019). To formally test the null hypothesis of separation fairness, we calculate a Wald chi-squared statistic using the estimated coefficients and annbiasedor of the (asymptotic) variance covariance matrix of the EIV coefficients.

$$\chi^{2} = \tilde{\boldsymbol{\beta}}_{G}\boldsymbol{\Sigma}_{\tilde{\beta}_{G}}^{-1}\tilde{\boldsymbol{\beta}}_{G}, \tag{6}$$

where $\tilde{\boldsymbol{\beta}}_{G}$ is the set of EIV regression coefficients only related to the group dummy variables, and $\boldsymbol{\Sigma}_{\tilde{\beta}_{G}}$ is the estimator of their covariance. This chi-squared statistic will approximately follow a chi-squared distribution with $K-1$ degrees of freedom in large samples.

3.3 Thresholds for Flagging Unfair Scores

Commonly, test statistics are compared to thresholds to decide if there is evidence of unfairness. These may be chosen to test for formal statistical significance or to identify meaningful differences, as was done by Williamson et al. (2012) for the standardized mean differences, or may be done with DIF. We do not provide general thresholds for our proposed statistical checks. Rather, following ETS's *Best Practices for Constructed-Response Scoring* (Educational Testing Service, 2021), we suggest that a threshold for any particular application consider multiple factors such as how the scores will be used (e.g., as sole score or in combination with a human rating), the risk or consequences of the test, the contribution of the item score to the total score, other evidence in support of the fairness of the AI scores, or other relevant information.

4. Remedies for Unfair AI Scores

In traditional assessment settings that do not use AI for scoring, fairness of test items is evaluated with differential item functioning procedures. When DIF is found to be present, a common practice is to have the item content reviewed by experts to determine if there is obvious content that could be producing fairness issues. When the experts and test developers think there is sufficient evidence, the items will be removed from the test and from the calculation of participant test scores.

Test developers and psychometricians could follow the same procedures and drop items for which the AI scores show evidence of being unfair to some groups, or score those items with human raters only. However, this is not always feasible, as the test might have few constructed response items or a small pool of items; so deleting all items with unfair AI scores threaten the validity of the test for other reasons, and using only human raters might make the test too costly. Another approach would be to identify problematic features by performing EIV regressions of the feature values on the item true score and group indicators and removing features that are found to be problematic. This process would continue until unfairness is no longer detected. Again, this could distort the content evaluated by the AI scores or lead to substantial losses in the predictive power of the AI-scoring model. Alternatively, the AI-scoring model could be altered to remove difference across groups that the model introduces. In this section, we discuss two methods of altering the model to remove group differences introduced by AI scoring: constrained optimization and penalization.

4.1 Constrained Optimization

In constrained optimization, the model parameters are constrained so that a measure of group difference introduced by the AI scoring model equals zero. For example, suppose we are using a linear function of the features as our AI scores, so that $M = \mathbf{a}^\top \mathbf{x}$. Then, typical predictive methods would generally find the vector of coefficients $\mathbf{a}$ that minimizes either the mean squared error $\sum_i \left(H_i - \mathbf{a}^\top \mathbf{x}_i\right)^2$ or the mean squared error plus some penalty term to account for model complexity, such as L_1 (Lasso) or L_2 (ridge) penalties. If we use the errors-in-variables estimate of unfairness, then our measure of unfairness in Equation (5) can be written as a linear function of the coefficients $\mathbf{a}$. For example, if we want to ensure that separation fairness holds exactly in the training sample, we would require that $\tilde{\boldsymbol{\beta}}_G = \left(\tilde{\beta}_2, \ldots, \tilde{\beta}_K\right)^\top$ would all equal zero, i.e.,

$$\begin{aligned}\tilde{\boldsymbol{\beta}}_G &= \left[\left(\mathbf{D}_H^\top \mathbf{D}_H - V\right)^{-1}\right]_{[3:(K+1),]} \mathbf{D}_H^\top \mathbf{M} \\ &= \left[\left(\mathbf{D}_H^\top \mathbf{D}_H - V\right)^{-1}\right]_{[3:(K+1),]} \mathbf{D}_H^\top \mathbf{X}\mathbf{a} = \mathbf{C}\mathbf{a} = 0\end{aligned} \tag{7}$$

where the notation $[A]_{[3:(K+1),]}$ means to take rows 3 to $K+1$ (the rows corresponding to group effect coefficients) from the matrix A. Note that the matrix $\mathbf{C}$ is the matrix of estimated regression coefficients of the feature scores on the true item scores and the group dummy indicators. We can correct issues of unfairness in the AI score by minimizing whatever fitting function (e.g., least squares) or penalized fitting function (e.g., Lasso) with respect to the coefficients $\mathbf{a}$ subject to the constraint $\mathbf{C}\mathbf{a} = 0$ using standard constrained optimization techniques as long as the number of constraints is not larger than the number of scoring features (e.g., Lange, 2004).

The proposed constrained optimization ensures only that the *estimate* of the dashed path in Figure 9.1a is zero, it does not guarantee that fairness in the general population of assessment participants will be achieved. Fairness should still be evaluated in an evaluation sample that is distinct from the training sample, or through cross-validation to determine how effective this approach at alleviating unfairness issues is.

This method of constrained optimization can be extended to any model, including nonlinear models, where the model parameters are solutions to estimating equations such as the score equations for maximum likelihood estimation. In these cases, the constraints can be added as additional equations to the optimization equations. This creates more equations than parameters, so the solution can be obtained using generalized method of moments (Hall, 2005). Extensions to other nonlinear AI algorithms, like neural networks and support vector machines, remain an area for future research.

4.2 *Penalization Methods*

The constrained optimization approach described in the previous section requires $K-1$ linear constraints when there are K groups to ensure the estimate of unfairness is zero for all groups. However, if the number of groups we consider becomes large, this might start to reduce the predictive power of our AI scores. Furthermore, as described earlier, the method only guarantees that the estimate of unfairness is zero in the sample used to train the model; with more constraints, the results might be more sensitive to the sample for fixed sample sizes. Hence, it may not be worth the cost to the predictive power if we cannot guarantee fairness in the larger population of participants.

Rather than forcing estimates of unfairness to be zero for all groups, it may be preferable to allow them to be nonzero, but to penalize the scoring algorithm by the magnitude of the unfairness. The linear constraint $\mathbf{Ca}=\mathbf{0}$ could be replaced with a penalty on a single measure of the total unfairness present in the scoring algorithm, such as $\mathbf{a}^\top\mathbf{C}^\top\mathbf{WCa}$, where $\mathbf{W}$ is a $(K-1)\times(K-1)$ positive definite weighting matrix that determines how the effects should be combined. For example, if we wanted to penalize the squared unfairness effect relative to the average effect, we might use the weighting matrix $\mathbf{W}=\mathbf{I}-\frac{1}{K}\mathbf{J}$, where $\mathbf{I}$ is the identity matrix and $\mathbf{J}$ is a matrix of ones. If we were performing least squares prediction, we would optimize the objective function

$$\sum_{i=1}^{n}\left(H_i-\mathbf{a}^\top\mathbf{X}_i\right)^2+\lambda\mathbf{a}^\top\mathbf{C}^\top\mathbf{WCa},$$

where $\lambda>0$ is parameter fixed to set the level of penalization on unfairness. As λ increases, the penalized solution will approach the constrained method described in the previous section.

Yao et al. (2019) proposed a similar method for combining information across different subscales to produce subscores that minimized subgroup biases in the resulting scores. The method, which they call the penalized best linear predictor (PBLP) method, was shown to substantially reduce unfairness while maintaining good levels of predictive accuracy.

Any method to reduce unfairness via constraints or penalizing group differences will typically result in a loss of prediction accuracy. When creating automated scores, analysts will need to balance the two competing goals of accurate predictions and reducing differential prediction bias across groups – for example, by choosing penalization over constraining the measures of unfairness to zero, or by setting λ to a smaller value when using penalization.

5. Application to Real Data

To demonstrate our approach for evaluating the fairness of automated scores, we examine the responses to three items on a large-scale reading assessment administered to middle school students. The three items all related to the same reading passage, which describes the natural habitat and behavior of a particular species of animal. The first item asked the student to provide evidence from the reading passage about a specific claim, and was scored on a 3-point scale. The second item was similar but asked the student to provide two pieces of evidence from the passage about a claim, and also was scored on a 3-point scale. The third item asked students to compare two position statements and make an argument about whether one of the two positions was more persuasive; this item was measured on a 4-point scale. The first two items had approximately 18,500 responses each, whereas the third item had approximately 16,200 responses. Each response was scored by a randomly assigned human rater, and approximately 5% of responses were scored by a second randomly assigned human rater for evaluation purposes.

To produce automated scores for the responses we used character bigram indicators as the predictor variables in a Lasso regression, where the penalty term was determined by 10-fold cross-validation and the outcome was the first (single) human rating. For the purposes of this example, we used the unrounded and unbounded predicted value for the response based on the fold where the response was not part of the training of the Lasso.

Table 9.1 provides information about the association between the automated scores and the human ratings. We provide the human-machine score correlation, the proportion reduction in mean squared error (PRMSE), and the quadratic weighted kappa (QWK). The correlation between human ratings is provided as a reference as well.

Our simple scoring method based on character bigrams does not perform great for any of the items. The best performing item, Item 1, has a PRMSE of 0.777, which leaves more than 22% of the variation in true human scores left unexplained by the machine scores. Items 2 and 3 have even lower PRMSEs, leaving an opportunity for a non-negligible amount of rubric-irrelevant variation to be associated with the demographic characteristics of the test-takers.

We evaluate the separation fairness of our automated scores by testing for evidence against the null hypothesis in (1) in favor of the alternative in (2) with a normal likelihood ratio test with the lavaan package (Rosseel, 2012) in R. Table 9.2 contains the deviances (−2 × log-likelihood) for the full model and reduced models, the SEM likelihood ratio test, and the EIV Wald test of separation fairness defined in Equation (6). With the exception of the EIV Wald test for Item 2, all significance tests are highly significant with p-values less than 0.01. The p-value for the EIV Wald test for Item 2 is 0.13.

To better understand the magnitude of the separation unfairness for each item, we examine the effects produced by the error-in-variables (EIV) regression. Table 9.3 examines the mean centered unfairness effects for each group, which we define as

Table 9.1 Measures of Association Between Automated Scores and Human Scores

	H-H	**H-M**		
Item	**Corr.**	**Corr.**	**PRMSE**	**QWK**
Item 1	0.911	0.841	0.777	0.829
Item 2	0.907	0.787	0.683	0.765
Item 3	0.876	0.738	0.622	0.707

Table 9.2 Tests of Separation Bias for the Three Reading Comprehension Items

	df	**Item 1**	**Item 2**	**Item 3**
SEM Full Model Deviance	27	79.56	54.67	437.39
SEM Reduced Model Deviance	45	164.08	194.50	833.78
SEM Likelihood Ratio Test	18	84.52	139.83	396.40
EIV Regression Wald Test	6	17.46	9.88	212.49

Table 9.3 EIV Regression Estimated Effects of Separation Unfairness for the Three Reading Comprehension Items

	Item 1		Item 2		Item 3	
Group	**d**	**z**	**d**	**z**	**d**	**z**
American Indian/Alaska Native	−0.006	−0.004	−0.022	−0.037	−0.116	−0.167
Asian	0.041	0.059	0.035	0.060	0.108	0.155
Black	−0.007	−0.010	−0.001	−0.001	−0.084	−0.121
Hawaiian & Pacific Islander	−0.018	−0.025	−0.009	−0.015	−0.095	−0.137
Hispanic	−0.002	−0.003	0.005	0.009	−0.060	−0.086
White	−0.001	−0.001	−0.003	−0.006	0.038	0.054
Multi-racial (non-Hispanic)	0.012	0.016	−0.007	−0.012	0.066	0.095

Note: The column marked *d* is the unstandardized effect, and the column marked *Z* is the standardized effect.

$$d_g = \hat{\gamma}_g - \sum_k p_k \hat{\gamma}_k,$$

where p_k is the proportion of the sample in group k, $\hat{\gamma}_1 = 0$ and $\hat{\gamma}_k$ for $k = 2,\ldots,K$ are the coefficients on the group dummy variables from the EIV regression. We also provide a standardized effect found by dividing d_g by the observed marginal standard deviation of the automated scores in the sample. Although not reported here, our suggested metrics based on SEM defined in Equation (3) are similar.

Although there was statistical evidence to suggest all three items had issues with separation unfairness, the magnitude of the unfairness for the first two items are quite small. The largest difference between standardized effects for Item 1 is between Asian and American Indian/Native Alaskan test-takers, with the standardized difference equaling 0.063, and the unstandardized difference being 0.047 (on a three-point item). Similarly, for Item 2, the standardized and unstandardized effects are 0.097 and 0.057.

In contrast, Item 3 has non-negligible unfairness effects. For example, the standardized and unstandardized differences between Asian and American Indian/Native Alaskan test-takers are 0.322 and 0.224 (on 4-point item), suggesting that on average, Asian test-takers get approximately a one-quarter point advantage when scored with the automated scoring algorithm compared to human scoring when compared to American Indian/Native Alaskan test-takers. In fact, Asian students get approximately a one-tenth of a point advantage with automated scoring compared to the average student, whereas American Indian/Native Alaskan, and Hawaiian/Pacific Islander students see approximately one-tenth of a point disadvantage.

5.1 Exploratory Analysis to Investigate Potentially Problematic Features

To understand what might be causing the issues of separation fairness for the third item, we carry out an exploratory analysis to identify features that may be leading to unfair results. To do so, we first carry out EIV regressions of the character bigram indicators $\mathbf{X}$ on the item true score T and group dummy variables. Let $\hat{b}_{gj}$ denote the group g effect for feature j from this regression. This effect is similar to the idea of differential feature functioning introduced by Zhang et al. (2017) and Penfield (2016); however, the authors in those papers conditioned on either the observed human ratings H or the automated score M instead of the item true human score T. We then multiply $\hat{b}_{gj}$ by the a_j, the Lasso estimated regression weight associated with feature j. Taking the difference of this index across groups gives us a measure of how much a given feature advantages or disadvantages one group compared to another.

The exploratory analysis described in the previous paragraph identified a number of features that appeared to advantage Asian test-takers over American Indian/Native Alaskan test-takers. The top five character bigrams that advantaged Asian test-takers were ‘pr’, ‘ua’, ‘id’, ‘tw’, and ‘mo’. Upon closer examination of the item and bigrams, we found that four of these five are contained in words taken directly from the prompt, which asked students to identify ‘two distinct sides’ on the issue and argue about which one is ‘more persuasive’. Asian test-takers were more likely to repeat the words from the prompt than American Indian/Alaska Native test-takers, and hence were advantaged by the bigram Lasso scoring engine.

In addition to this finding, it turns out the weights associated with the bigram indicators are somewhat more likely to be positive than negative. Asian test-takers tended to have more unique bigrams on average (127.4) compared to American Indian/Alaska Native test-takers (82.2). This difference alone contributes approximately 0.07 points of advantage to Asian test-takers.

These explorations of DFF offer insights into the subgroup differences. They do not suggest a specific path to removing those differences. Subject matter experts will need to interpret the implications of Asian students than other groups seemingly being more likely to use components of the item in their responses and whether this is adding construct-irrelevant variance and bias. Also, classifications of students by other factors, e.g., gender or economic status, or by interactions of such factors might reveal other differences for additional exploration. This is not unlike DIF analysis. Typically, items flagged by DIF are reviewed by subject matter experts, and the path forward comes from the combination of empirical analysis and expert judgments.

5.2 Reducing Separation Unfairness

In an effort to reduce the separation unfairness for the third item, we apply the penalty described in the section ‘Penalization Methods’ as part of a Lasso regression. Specifically, we find the set of coefficients $\mathbf{a}$ that minimizes the following penalized least squares fitting function,

$$\frac{1}{2}\sum_{i=1}^{N}\left(H_i-\mathbf{a}^{\top}\mathbf{X}_i\right)^2+\frac{\lambda}{2}\mathbf{a}^{\top}\mathbf{C}^{\top}\mathbf{W}\mathbf{C}\mathbf{a}+\eta\sum_{j=1}^{J}\left|a_j\right|.$$

For this application, we define $\mathbf{W}=\mathbf{U}\mathbf{U}^{\top}$ with $\mathbf{U}$ defined as the $(K-1)\times K$ matrix

$$\mathbf{U}^{\top}=\begin{pmatrix}\mathbf{I}-\mathbf{1}\otimes\mathbf{p}^{\top}\\ -\mathbf{p}^{\top}\end{pmatrix},$$

where $\otimes$ denotes the Kronecker product, so that $\mathbf{1}\otimes\mathbf{p}^{\top}$ equals a matrix with K rows and every row equals $\mathbf{p}^{\top}$. This results in placing a penalty on $\sum_k\left(\gamma_k-\tilde{\gamma}\right)^2$, where $\tilde{\gamma}=\sum_k p_k\gamma_k$, and p_k is the proportion of the sample in group k.

In order to fit this penalized Lasso regression model, we use a data augmentation approach similar to one described in Tibshirani and Taylor (2011) and Gaines et al. (2018) by including $(K-1)$ artificial observations with $H_i=0$ for $i=N+1,\ldots,N+K-1$ and feature vectors equal to the rows equal of $\sqrt{\lambda}\mathbf{W}^{\frac{1}{2}}\mathbf{C}$, where $\mathbf{W}^{\frac{1}{2}}$ is the Cholesky decomposition of $\mathbf{W}$; the constant/intercept is also removed for these artificial observations.

We find the best Lasso regression for nine values of $\lambda=10^k$ for $k=0,\ldots,8$ by selecting the L_1 penalty η that produces the lowest 10-fold cross-validation estimated mean squared error in the sample data. For each value of λ, we plot the PRMSE, an overall measure of fairness defined as $\frac{1}{s_m\sqrt{K}}\sum_k\left(\hat{\gamma}_k-\tilde{\gamma}\right)^2$, and plot of the group-level effects $\hat{\gamma}_k$. These plots appear in Figure 9.2.

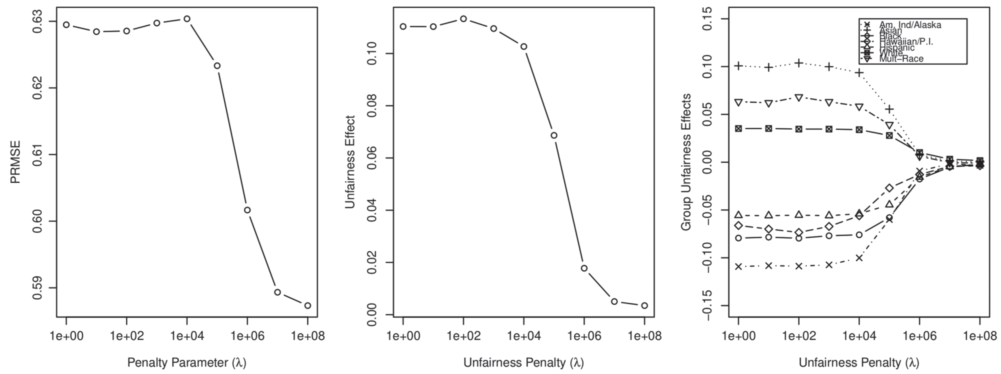


Figure 9.2 Line plots of PRMSE, our measure of overall unfairness, and the group-level unfairness effects for different penalties on unfairness in the Lasso regression.

The unfairness effect in the middle panel is directly related to the penalty term we use in our penalized Lasso. The unfairness index remains relatively constant at about 0.11 for values of λ from 1 to 10^4. [2] However, the overall measure of unfairness falls off sharply for values of between $\lambda = 10^4$ and 10^6 before it starts to bottom out near zero, as all the group means become equal to the overall mean. This also can be seen in the plot of the effects at the group level. By $\lambda = 10^7$, it looks like most of the unfairness has been removed from the automated scoring procedure. However, as the left panel indicates, the removal of the unfairness from the automated scoring engine comes at the cost of accuracy; PRMSE decreases as fairness improves.

Where to ultimately set the unfairness penalty parameter λ will depend on the specific application. For this particular example, setting $\lambda = 10^6$ does fairly well at reducing the amount of unfairness in the automated scores but does not reduce the PRMSE quite as much as the highest two penalty levels do. However, the resulting PRMSE is still quite low at 0.60. Therefore, if this automated scoring algorithm was being considered for this item, the best decision might be simply not to use automated scoring for this particular item, or to try to find an alternative scoring algorithm using different features. More generally, how to choose among the competing goals of high overall accuracy and removing difference across groups will need to consider the overall context of the test and the AI system. For example, if there is evidence that differences across groups in AI features are capturing construct-relevant variance, then subgroup difference might be less of a concern.

6. Discussion

One of the most common applications of AI in education is its use in automated scoring of responses to test items. Fairness is a central tenant of educational and psychological testing and the application of AI. However, methods for ensuring the fairness of automated scores have not been given thorough evaluation. In this chapter, we noted that multiple definitions of fairness in the application of AI exist in the literature, including independence, sufficiency, and separation. We note that separation fairness, which holds that, conditional on the latent ability scores, distribution should be the same for test-takers regardless of test-takers' group identity, is the underlying principle commonly used for defining fair scores in educational measurement. It is the principle implicitly tested by DIF analysis. In most educational testing applications, the availability of multiple items allows for testing for violations of fairness. In automated scoring, the human ratings allow for testing for the fairness of the automated score. Using the scores, we can test that the automated scores are fair for making inferences about the human true score. The human true scores (and implicitly the human raters) must support fair inferences about latent ability.

Although the human ratings allow for tests of separation fairness, the widely used test of checking that the SMD or DSM is small in absolute value is not a valid test of separation fairness, as SMD and DSM can be small when separation fairness does not hold and can be large when it does. We propose two methods to directly test for violations of separation fairness. Important to these methods is their accounting for the measurement error in human ratings. Previously, authors have suggested regressing the automated score on group indicators and the human rating (Loukina et al., 2019); however, because of measurement error, this could increase the chances of spuriously rejecting the hypothesis that scores are fair and lead to incorrect conclusions about the fairness of the scores. EIV regression also regresses the automated scores and group indicators but corrects for the measurement error in the human ratings and can reduce the biases of the simple regression approach. As we saw in our example, the EIV approach might be less powerful than the model-based structural equation modeling method. In the example, the structural equation modeling method found that scores for all three items violated separation fairness, but the EIV regression failed to reject the hypothesis that the scores for Item 2 were fair. On the other hand, the structural equation modeling method is dependent on distributional assumptions, and violation of those assumptions could yield incorrect inferences. The robustness of the method to violations of the method to violations of assumptions needs further study, as does, the relative power and other tradeoffs between these two methods. Johnson et al. (2022) suggest another approach for testing fairness that also corrects for measurement error by using the conditional score method (Carroll et al., 2006) and creating a sufficient statistic for the true score from a linear combination of the human and machine score. How this method compares with methods presented here should also be considered in the future.

Testing for fairness is only a small part of the work to ensure the fairness of automated scores. As recommended by the Standards, fairness should be considered throughout the entire testing process from development through to test administration, scoring, score reporting, and uses of the test. Fairness reviews of the item content and human rater rubrics should be conducted as they would be in the absence of AI scoring. The use of AI scoring and details about the AI methods and models, Iing any NLP features derived from responses, should be considered in those reviews. Similarly, the potential for AI scoring to impact the equity of the uses or consequences (e.g., by increasing accessibility through lower cost) should also be evaluated as part of the assessment development and ongoing monitoring of assessment programs that use AI scoring. More specifically, exploring the features through different feature functioning analysis or other exploratory methods could reduce construct-irrelevant differences in the scores across groups and improve the quality of the scores. These investigations could also improve the understanding of the scores, providing more evidence of the validity of the scores for the proposed inferences and uses of the tests. They could also suggest ways to revise the features, models, or item to improve validity and fairness.

Constraining the model to remove or reduce group differences can also serve as an important component of the work to ensure fairness. As demonstrated in the example, adding a penalty for differences among groups can lead to model fits that yield minimal group differences. On the surface such solutions might appear to be preferable, but that might not be the case. First of all, including the penalty reduced the accuracy of the automated scoring model for predicting the true scores. In terms of the validity of the scores for their intended purposes, the relative value of small group differences or more accurate predictions is difficult to judge. It will clearly depend on the starting point of both. For example, if the accuracy for the model without the penalty is very high, then some loss of accuracy might be less costly than if the accuracy was initially only moderate. Second, there are often multiple ways to classify test-takers into groups. Removing differences among one set of groups could exacerbate differences among groups from a different classification. For instance, removing differences on language groups could create differences among test-takers of different genders. Third, the penalization methods reweight input features without any consideration of the substance of the features. If some features show more DFF, it might be preferable to downweight those features rather than others, but the method cannot account for those preferences. Fourth, changing the model to remove differences will make the model better match human rating and the machine score means of some groups at the cost of making the means of human ratings and machine scores for other groups less aligned. In some sense, one group is penalized to support another group. This could be considered unfair, even though from a statistical sense, the scores would appear to be more fair. These issues will need more debate and investigation before any approach to ensuring fairness becomes standard practice. The goal of this chapter was to start those investigations and discussions by presenting these methods and showing their potential through the empirical example.

The proposed methods assume that human true scores are unbiased. Although it is true that human raters can be biased in the sense that they do not score as the rubric intends, there are methods to check for human rater bias and mitigate it. For example, a common practice is to test the accuracy of each individual rater on *validity samples*. Validity samples are responses that have been scored by multiple experts to produce what is agreed on as an accurate, unbiased rating for each response. Raters are then tested by their ability to agree with the expert ratings on the validity samples when these samples are randomly included among the responses assigned to each rater. Rater modeling (Casabianca et al., 2016; Patz et al., 2002) can also be used to identify individual raters who produce systematic errors (e.g., consistently too high or too low) relative to other raters or validity samples. Raters who demonstrate biases can be remediated or removed. As long as most raters are not biased and raters are assigned responses randomly, a few biased raters would appear more like error, and they would not introduce notable bias into the proposed methods. Additional checks such as on how raters apply the rubrics and/or reviews of the rubrics can also be used to reduce the risk of human rater bias. Since the machines are trained to predict the human raters, bias in the human raters are more likely to be coded into the machines than for the model to correct for biases in human raters.

Notes

1 See https://github.com/NAEP-AS-Challenge/info/blob/main/2021_10_4_IES_AS_Challenge_RFI_Presentation.pdf for details.

2 The plot should be non-increasing, but due to randomness of the cross-validation, there are some points where it increases.

References

Attali, Y., & Burstein, J. (2006). Automated essay scoring with e-rater v.2. *The Journal of Technology, Learning and Assessment*, *4*(3). https://ejournals.bc.edu/index.php/jtla/article/view/1650

Barocas, S., Hardt, M., & Narayanan, A. (2019). *Fairness and machine learning: Limitations and opportuni ties*. http://fairmlbook.org

Bridgeman, B., Trapani, C., & Attali, Y. (2012). Comparison of human and machine scoring of essays: Differences by gender, ethnicity, and country. *Applied Measurement in Education*, *25*(1), 2 7–40. https://doi.org/10.1080/08957347.2012.635502

Camilli, G. (2006). Test fairness. In R. L. Brennan (Ed.), *Educational measurement* (4th ed., Vol. 4, pp. 221–256). American Council on Education, Praeger.

Carroll, R. J., Ruppert, D., Stefanski, L. A., & Crainiceanu, C. M. (2006). *Measurement error in nonlinear models: A modern perspective*. Chapman and Hall, CRC Press.

Casabianca, J. M., Junker, B. W., & Patz, R. J. (2016). Hierarchical rater models. In W. J. van der Linden (Ed.), *Handbook of item response theory* (Vol. 1, pp. 477–494). Chapman and Hall, CRC Press.

Cronbach, L. J., Rajaratnam, N., & Gleser, G. C. (1963). Theory of generalizability: A liberalization of reliability theory†. *British Journal of Statistical Psychology*, *16*(2), 137–163. https://doi.org/10.1111/j.2044-8317.1963.tb00206.x

Culpepper, S. A., & Aguinis, H. (2011). Using analysis of covariance (ANCOVA) with fallible covariates. *Psychological Methods*, *16*(2), 166 –178. https://doi.org/10.1037/a0023355

Educational Testing Service (2021). *Best practices for constructed-response scoring*. ETS.

Friedler, S. A., Scheidegger, C. E., & Venkatasubramanian, S. (2016). On the (im)possibility of fairness. *arXiv Computing Research Repository (C oRR)*. https://arxiv.org/abs/1609.07236

Fuller, W. A. (1980). Properties of some estimators for the errors-in-variables model. *The Annals of Statistics*, *8*(2), 407 –422. www.jstor.org/stable/2240543

Gaines, B. R., Kim, J., & Zhou, H. (2018). Algorithms for fitting the constrained lasso. *Journal of Computational and Graphical Statistics*, *27*(4), 861 –871. https://doi.org/10.1080/10618600.2018.1473777

Gipps, C., & Stobart, G. (2004). Fairness in assessment. In *Perspectives on pupil assessment* (pp. 31–35). General Teaching Council for England.

Haberman, S. J. (2019). Measures of agreement versus measures of prediction accuracy. *ETS Research Report Series*, *2019*(1), 1–23.

Hall, A. R. (2005). *Generalized method of moments*. Oxford University Press.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). Springer-Verlag.

Holland, P. W., & Thayer, D. T. (1988). Differential item performance and the Mantel-Haenszel procedure. In H. Wainer & H. Braun (Eds.), *Test validity*. Lawrence Erlbaum Associates.

Holland, P. W., & Wainer, H. (1993). *Differential item functioning*. Lawrence Erlbaum Associates.

Hutchinson, B., & Mitchell, M. (2019). 50 years of test (un)fairness: Lessons for machine learning. *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 4 9–58. https://doi.org/10.1145/3287560.3287600

International Test Commission and Association of Test Publishers (2022). *Guidelines for technology-based assessment*. ITC.

Johnson, M. S., Liu, X., & McCaffrey, D. F. (2022). Psychometric methods to evaluate measurement and algorithmic bias in automated scoring. *Journal of Educational Measure ment*. https://doi.org/https://doi.org/10.1111/jedm.12335

Kane, M. T. (2006). Validation. In R. L. Brennan (Ed.), *Educational measurement* (4th ed., pp. 221–256). American Council on Education, Praeger.

Lange, K. (2004). *Optimization*. Springer-Verlag.

Loukina, A., Madnani, N., & Zechner, K. (2019). *The many dimensions of algorithmic fairness in educational applications* (pp. 1–10). Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications.

Madnani, N., Loukina, A., & Cahill, A. (2017). *A large scale quantitative exploration of modeling strategies for content scoring* (pp. 457–467). Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications.

Markus, K. A., & Borsboom, D. (2013). *Frontiers of test validity theory: Measurement, causation, and meaning*. Routledge.

Messick, S. (1989). Validity. In R. L. Linn (Ed.), *Educational measurement* (3rd ed., pp. 13–103). American Council on Education, National Council on Measurement in Education.

Ohio Department of Education (2018). *Machine scoring questions and ans wers*. https://education.ohio.gov/getattachment/Topics/Learning-in-Ohio/English-Language-Art/Assessments-for-English-Language-Arts/scoring_FAQ_March2018_final.pdf

Page, E. B. (1966). The imminence of . . . grading essays by computer. *The Phi Delta Kaplan*, *47*(5), 238 –243. www.jstor.org/stable/20371545

Page, E. B. (1968). The use of the computer in analyzing student essays. *International Review of Education*, *14*(2), 210 –225. https://doi.org/10.1007/BF01419938

Patz, R. J., Junker, B. W., Johnson, M. S., & Mariano, L. (2002). The hierarchical rater model for rated test items and its application to. *Journal of Educational and Behavioral Statistics*, *27*(4), 341–384.

Pearson (2017). *PARCC: Final technical report for 2017 administration*. Partnership for Assessment of Readiness for College, Careers.

Penfield, R. D. (2016). Fairness in test scoring. In N. J. Dorans & L. L. Cook (Eds.), *Fairness in educational assessment and measurement*. Routledge.

Rosseel, Y. (2012). lavaan: An R package for structural equation modeling. *Journal of Statistical Software*, *48*(2), 1–36. www.jstatsoft.org/v48/i02/

Searle, S. R., Casella, G., & McCulloch, C. E. (1992). *Variance components*. John Wiley & Sons.

Tibshirani, R. J., & Taylor, J. (2011). The solution path of the generalized lasso. *The Annals of Statistics*, *39*(3), 1335–1371. https://doi.org/10.1214/11-AOS878

Vapnik, V. N. (2000). *The nature of statistical learning theory* (2nd ed.). Springer.

Williamson, D. M., Xi, X., & Breyer, F. J. (2012). A framework for evaluation and use of automated scoring. *Educational Measurement: Issues and Practice*, *31*(1), 2–13. https://doi.org/10.1111/j.1745-3992.2011.00223.x

Yao, L., Haberman, S. J., & Zhang, M. (2019). Penalized best linear prediction of true test scores. *Psychometrika*, *84*(1), 186 –211. https://doi.org/10.1007/s11336-018-9636-7

Zhang, M., Dorans, N. J., Li, C., & Rupp, A. A. (2017). Differential feature functioning in automated essay scoring. In H. Jiao & R. W. Lissitz (Eds.), *Test fairness in the new generation of large-scale assessment* (pp. 185–208). Information Age Publishing.

Part IV

Emerging Technologies

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

10

Extracting Linguistic Signal From Item Text and Its Application to Modeling Item Characteristics

Victoria Yaneva, Peter Baldwin, Le An Ha, and Christopher Runyon

1. Introduction

One novel application of natural language processing (NLP) in assessment that has received growing interest is the modeling of item characteristics using predictors extracted from item text. Often, these attempts to capitalize on the relationship between text features and item characteristics occur in advance of pretesting, when response data have not yet been collected and the item text is the only available information about the item. It follows that the predicted item characteristics of greatest interest will be those that can *increase* the efficiency of pretesting or *reduce* its various negative side effects. In the case of efficiency – and pressure for efficiency gains will only increase with advances in automatic item generation – improvements can be made either by predicting items' probability of survival (i.e., the probability of satisfying the statistical criteria for use in scoring; Ha et al., 2019; Yaneva et al., 2020) or by eliminating (or reducing) the necessity for pretesting altogether through item difficulty prediction (Benedetto et al., 2020; Kurdi, 2020; Leo et al., 2019; Xue et al., 2020). With respect to reducing the negative aspects of pretesting, it has been shown that the timing variability of test forms can be reduced by predicting the time demands of pretest items prior to form assembly (Baldwin et al., 2021). For these activities and others, researchers have found that variables extracted from item text can predict item characteristics better than several baselines; yet the practical importance of these gains has not been convincingly demonstrated in all cases. As a result, the application of NLP to prediction problems in educational measurement remains an active and exciting area for research.

Because of its potential to illuminate and inform test development, the understanding of relationships between ancillary item data generally and various item characteristics has been of long-standing interest to assessment specialists. NLP has expanded and enriched the universe of ancillary data in novel ways, but despite this interest, it has not been widely used for this purpose. For example, except for Baldwin et al. (2021), the studies cited here were published in NLP venues, illustrating the limited exposure these methods have within educational measurement and identifying potential methodological and knowledge gaps. In this chapter, we address some of these gaps by providing an overview of several well-known NLP approaches

DOI: 10.4324/9781003278658-14
This chapter has been made available under a CC-BY-NC-ND 4.0 license.

for representing text and demonstrating how these representations can be used to solve practical measurement problems. This twofold purpose also structures the chapter.

More specifically, our overview of text representation methods starts with a summary of traditional linguistic features, moves on to introduce non-contextualized word embeddings,[1] and then concludes with a nontechnical primer on contextualized embeddings. These descriptions are targeted to readers with no background in NLP. The second part of the chapter provides an empirical illustration of these approaches by outlining the process of predicting item characteristics for multiple-choice questions (MCQs) accompanied by various relevant findings. In this context, several practical considerations are highlighted, including: the choice of pretraining data and model architecture, the encoding of different levels of dependencies, and the constraints imposed by model interpretability.

2. Representing Item Text

As mentioned, in this section we introduce three different classes of ancillary data that can be extracted from an item's text and explain how these data can be used to predict item characteristics. What we might call *ancillary* or *collateral data* in this context are generally referred to as *features* in the NLP literature. Next, these categories are presented in the following order: *human-engineered linguistic features*, *non-contextualized embeddings*, and *contextualized embeddings*, which also follows the order of their increasing abstraction (and, likewise, their chronological development). This overview is brief, merely intending to introduce those readers unfamiliar with NLP to the main approaches to text representation. For a detailed, NLP-focused review, we refer the reader to Pilehvar and Camacho-Collados (2020).

2.1 Human-Engineered Linguistic Features

Early approaches to text processing relied heavily on linguistic information extracted through human-engineered features. This extraction process requires both: (1) an initial hypothesis that a given feature will covary with a variable of interest (e.g., the hypothesis that *average noun phrase length* is related to the readability of text passages); and (2) the necessary NLP tools and resources for extracting the predictor (e.g., a *parser* and *part-of-speech tagger* that can separate a given text into relevant subparts and identify which parts constitute noun phrases).[2] Other examples of human-engineered features include *number of polysemous words*,[3] which is intended to capture semantic ambiguity; and *age of acquisition*, which is meant to capture the familiarity subjects (e.g., students) are expected to have with a given word at a given age. There are many others. Linguistic features can capture different levels of linguistic processing such as lexical, syntactic, semantic, and discourse, and they have been used to predict item difficulty in the context of reading and listening comprehension exams (e.g., Choi & Moon, 2020; Loukina et al., 2016). Beyond reading exams, linguistic features have also been shown to predict item difficulty more generally as well as the average time required to respond to different MCQs (Baldwin et al., 2021).

The extraction of linguistic features is highly reliant on NLP resources. To measure *polysemy*, first, ontologies are needed that encode semantic relationships (e.g., WordNet; Miller, 1995); to measure *age of acquisition*, normed word lists are needed (e.g., MRC psycholinguistic database; Coltheart, 1981); and so on. As can be expected, early approaches to extracting linguistic features were constrained by the availability and coverage of these kinds of resources, which were both costly and slow to develop.

Despite these challenges, the hypothesis-driven approach (where a feature is extracted only because of a researcher's hypothesis that it may have predictive power) has been successfully applied to many practical problems and is especially useful when a given application calls for interpretable features. For example, this approach allows the researcher not only to extract highly predictive features, but also to exclude ones that should not be used (e.g., text length when predicting essay scores) and have better control over model bias. This advantage of linguistic features, however, is also their limitation: *because* they are hypothesis driven, linguistic features may not always capture the most important or relevant predictors a given dataset has to offer for a given problem. For a data-driven approach, we instead must turn to a new paradigm in NLP research: dense word vector representations, also known as *word embeddings*.

2.2 Word Embeddings: Theoretical Background

The notion of word embeddings has its origins in the *distributional hypothesis*, which states that words occurring in the same contexts tend to have similar meanings (Harris, 1954). This hypothesis was later immortalized by Firth (1957) as: 'You shall know a word by the company it keeps'. A well-known illustration of this phenomenon is an experiment by McDonald and Ramscar (2001), who placed nonce words such as *wampimuk* in different contexts – e.g., 'He filled the *wampimuk* with the substance, passed it around and we all drunk some' and 'We found a little, hairy *wampimuk* sleeping behind the tree'. When presented in these contexts, *wampimuk* was consistently understood by the study participants to refer to some type of container for holding liquid or an animate creature, respectively.

The distributional hypothesis has important implications for the computational processing of language, since context can be represented numerically by encoding word co-occurrences in large collections of texts (*corpora*). In other words, if we can encode a sufficiently large number of contexts for a given word (or subword[4]), we can infer its semantic, syntactic, or pragmatic[5] properties without having to rely on external resources such as ontologies. While this was only a theoretical possibility a few decades ago, it is now practically feasible thanks to two advances: the accumulation of large amounts of electronically stored text data, which allows a sufficient number of co-occurrences to be encoded, and developments in parallel computing, which provide the computational power needed to process these large datasets. These developments were further aided by advances in deep neural network models that made it possible to condense high-dimensional and sparse co-occurrence vectors into dense vectors with fewer dimensions. These dense vectors are sometimes called *dense vector representations* but, more often, are referred to as *embeddings*. You can think of an embedding as the location of a word in an *n*-dimensional vector space, and it follows that its semantic properties can be inferred based on other nearby words in this space. The high predictive power of dense vector representations for many NLP tasks was first demonstrated by early embedding types such as Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014), which we discuss next.

2.3 Non-Contextualized Embeddings

Early embedding types are sometimes described as *non-contextualized embeddings* because they do not fully capitalize on differences in context. So, the word *hard* in 'I read Simon's book, which was hard' and 'Simon hit me with his book, which was hard' is represented by a common embedding. This limitation was later addressed by contextualized embeddings (described in the next section), in which the various uses of a given polyseme such as 'hard' are encoded separately. Nevertheless, non-contextualized embeddings are less demanding computationally and have been productively used to solve many problems.

Creating non-contextualized embeddings can be divided into two distinct stages: *data generation* and *model training*. We illustrate this process with Word2Vec.[6] During the dataset generation stage, neighboring words for each word in the corpus (i.e., the training data such as Google News or PubMed articles) are identified. For a *sliding window* (or *context window*) with *size 2*, a given word's neighbors are the two words preceding it and the two words following it. So, for example, suppose our corpus contains the preprocessed[7] sentence, 'You shall know word by company it keeps'. Data generated for the input words *know* and *word*, with a sliding window size of 2, would look as shown in Table 10.1.

Eventually, during the model-training stage (described later in this section), these data are used as input for a neural network tasked with predicting the value in the Target column (i.e., whether or not the input and output words are neighbors – sometimes called the *label*); however, note that here the target values are all 1, and so before this can be done, additional data are needed. To address this, output words are added to the dataset that are randomly sampled[8] from the vocabulary (a process called *negative sampling* that works by contrasting signal with noise). These sampled words are *not* a given input word's neighbors and so their target values are all 0, as shown in Table 10.2.

This procedure generates a large dataset of word co-occurrences (and non-co-occurrences) without relying on manual annotation or external resources, as in the case of extracting linguistic features described earlier.

Data generation is followed by the model-training stage, which begins with the creation of two matrices that are first initialized with random numbers: an embedding matrix, which will store the embeddings of the input words, and a context matrix, which will store the embeddings of the output (context) words. These are $m \times n$ matrices, where m is the number of words in the corpus vocabulary and n is the desired number of dimensions for the embeddings (e.g., 300).[9]

Both matrices are initialized with random numbers and then updated during training as follows. For each word in the dataset, the model takes one positive sample and some number[10] of

Table 10.1 Example (Partial) Training Data Samples (Before Applying Negative Sampling)

Input Word	**Output Word**	**Target** (*Are the Input and Output Words Neighbors?*)
know	you	1
know	shall	1
know	word	1
know	by	1
word	shall	1
word	know	1
word	by	1
word	company	1

Table 10.2 Example Training Data Samples for the Input Word 'Know' With Negative Sampling

Input Word	**Output Word**	**Target** (*Neighbors?*)
know	you	1
know	shall	1
know	word	1
know	by	1
know	aardvark	0
know	aarhus	0
know	. . .	0
know	truck	0

Table 10.3 Training Sample for the Input Word 'Know'

	Input Word	**Output Word**	**Target** (*Neighbors?*)
Positive Sample	know	you	1
Negative Sample	know	aardvark	0
	know	truck	0

negative samples. Table 10.3 illustrates this for the input word 'know' with the positive sample 'you' and two negative samples, 'aardvark' and 'truck'.

These samples correspond to four embeddings: one from the embedding matrix (for the input word 'know') and three embeddings from the context matrix (for the output words 'you', 'aardvark', and 'truck'). The similarity between each input word and output word then can be quantified by the dot product for each input word and output word embedding pair. Each of these three dot products then is transformed into a value ranging between zero and one using the sigmoid function:

$$p(neighbor\,|\,w,c) \sim \sigma\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{c}\right) = \frac{1}{1+e^{-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{c}}}$$

where $p(c\,|\,w)$ is the (estimated) probability that an input word w and an output word c (the context) are neighbors; $\sigma(\cdot)$ is the sigmoid function; and $\mathbf{w}^{\mathrm{T}}\mathbf{c}$ is the dot product between the embedding for the input word (from the embedding matrix) and the embedding for the output word (from the context matrix).

For this small sample of training data, this process produces three probabilities – one for each input word/output word pair. Of course, the model is still untrained (the embedding and context matrices are still in their initial random state), and so these probabilities are, at this point, meaningless (in fact, even when they mean something, these probabilities will not be of any importance to us – we care only about the embeddings matrix from the hidden layer). As a next step, then, these probabilities are subtracted from their target value (1 for neighbors and 0 otherwise), yielding errors. This produces an error vector, which then is used to adjust the embedding weights for the input and output words in the embedding and context matrices, respectively. As this iterative process is repeated (the number of iterations may depend on computational resources), the predictions and embeddings gradually improve. After training is complete, each input word has a Word2Vec embedding in the embedding matrix with a fixed number of dimensions. Provided the training set is comprehensive enough (only words in the training set will have embeddings), Word2Vec embeddings pretrained in this way can be used as predictors for various other tasks. These tasks include predicting item characteristics, which is possible when items with similar meanings (as captured by their embeddings through encoding similarities in context) are similar with respect to the item characteristic of interest.[11]

For many NLP tasks, non-contextualized word embeddings have been shown to perform as well or better than human-engineered linguistic features, without requiring annotated corpora or external resources. For example, Word2Vec embeddings and linguistic features produced comparable results when predicting item difficulties (Ha et al., 2019) and average response times (Baldwin et al., 2021) for clinical MCQs. Nevertheless, as noted, non-contextualized embeddings like Word2Vec and GloVe fail to account for, well, *context*; and in the absence of context, some meaning cannot be represented by a single embedding per word. Recent advances have addressed this shortcoming by using *contextualized* word embeddings.

2.4 Contextualized Embeddings

Most current contextualized word embeddings are produced using large models with millions of parameters known as *transformer models*.[12] Given the complexity of these models, a detailed explanation of how they work is outside of the scope of this chapter (for a more in-depth description, see Devlin et al., 2018; Wolf et al., 2020). Here, we focus on the output from these models and how it can be used for the task of predicting item characteristics.

Several well-known contextualized embedding models have been developed, including ELMo (Peters et al., 2018), GPT-2 (Radford et al., 2019), and GPT-3 (Brown et al., 2020). However, perhaps the most widely used among these at the time of this writing is BERT (Bidirectional Encoder Representations from Transformers; Devlin et al., 2018). The BERT model is pretrained on a vast collection of texts and is freely available to download.[13] In the original work, there are two versions of BERT: *BERT Large*, which performs best in a number of benchmarking tasks but has more than 300 million parameters to train; and a lighter version known as *BERT Base* (merely 110 million parameters) that performs slightly worse on some benchmarks but requires far less computational power.

In the context of predicting item characteristics, models like BERT can be trained on either generic or domain-specific texts. When medical text embeddings are desirable, for example, a corpus such as MEDLINE abstracts,[14] can be used to generate the predictive word embeddings for the task. Once trained, the model can be *fine-tuned* for the specific data and task of interest – e.g., on a given set of test items and their item characteristics, respectively. Fine-tuning typically involves adding an extra layer on top of the original deep neural network and then retraining the model on the desired task (e.g., on predicting item difficulty). In this way, the internal weights of the pretrained model are updated without discarding the knowledge gained from the original model training. Alternatively, as in the case of our experiments later in this chapter, the pretrained model can be used to generate embeddings for a dataset of interest *without* fine-tuning. In these cases, the generated embeddings are used as input for training familiar machine learning models such as regression, random forests, support vector machines, and so on. Several studies suggest that the latter approach may be more successful for some applications (e.g., Caron et al., 2020; Conneau et al., 2017; Conneau & Kiela, 2018); however, as we will describe, this was not our only motivation here.

Unlike models such as Word2Vec described earlier, BERT is trained on *two* tasks that produce *two* different types of embeddings: one for tokens (e.g., words), and one for sentences:

- *Token embeddings: Token* refers to both words and subwords. The algorithm first breaks all words into subwords and then reconstructs words from the smaller units, which gives the model the capability to handle vocabulary that is *not* included in the training data. Task one, then, is to predict each token in the training corpus by those tokens appearing before or after it, with the goal of encoding as much context as possible. Recalling the previous example, unlike non-contextualized embeddings, here the ambiguous word *hard* would be represented differently depending on whether it means 'difficult' or 'solid' in a given context. Moreover, although embeddings are at the individual word (or subword token) level, they can be pooled to represent a document (which could be an item, for example).
- *Sentence embeddings:* The goal of task two is to determine the sequence of two sentences (e.g., *does sentence B follow sentence A?*). Trained this way, the model produces embeddings for entire sentences rather than individual words or subwords, which may be beneficial for tasks where this larger context contains relevant information. In the case of test items, each item comprises several BERT sentence embeddings (depending on the number of sentences in the item) that can be pooled to form the final item embedding.

Given BERT's popularity, variations have been developed to improve aspects of the original model. For example, DistilBERT reduces the size of a BERT model by 40%, which requires significantly less computational power to train, 'while retaining 97% of its language understanding capabilities and being 60% faster' (Sanh et al., 2019). Other models such as RoBERTa (Liu et al., 2019) improve the hyperparameter tuning of the original BERT, leading to better performance on several benchmarking NLP tasks. There are versions of BERT that have been pretrained on domain-specific texts, such as Clinical BERT (Alsentzer et al., 2019), which is trained on clinical texts. Given the separation of the training and fine-tuning discussed earlier, domain-specific pretrained models allow researchers to focus on the modeling aspect of a given problem without requiring the computational power and access to data typically necessary for robust model pretraining. Finally, other transformer-based architectures include DeBERTa (He et al., 2020), Electra (Clark et al., 2020), and many others.

In the context of predicting item characteristics, contextualized embeddings have shown promising results. Ha et al. (2019) compare different predictors including linguistic features, Word2Vec embeddings, and ELMo embeddings for the task of predicting item difficulty for clinical MCQs. The results outperformed several simple baselines, and ELMo performed best among these three predictor classes (although the best results overall were obtained by the combination of linguistic features *and* ELMo, indicating that the signals they encode complement rather than completely overlap each other). Outside of the medical domain, Benedetto et al. (2021) found that BERT and DistilBERT were more successful than ELMo and several other baseline approaches in predicting item difficulty for math and IT MCQs.

2.5 Other Predictors of Item Characteristics

The previous sections gave a short overview of different ways to extract and represent linguistic information from item text that can be used to predict item characteristics for MCQs. In addition to these, there have been several other NLP-related approaches for predicting item characteristics. These approaches have so far been less successful than the ones already described based on linguistic features and embeddings; however, for greater context, a short description of these is given next.

The first approach predicts item difficulty using TF-IDF (Term Frequency–Inversed Document Frequency) representation using a tool called R2D2 (Benedetto et al., 2020). TF-IDF is a well-known early approach to text representation that relies on sparse co-occurrence vectors of words in a document. Later experiments by the same authors show that R2D2 is outperformed by contextualized embeddings such as BERT (Benedetto et al., 2021).

Another approach reported in Ha et al. (2019) used the output of an automatic question-answering system[15] for predicting item difficulty. This approach was based on the hypothesis that there is a positive relationship between the difficulty of questions for humans and their difficulty for machines. An information retrieval–based automated question-answering system was applied to a set of MCQs, and the retrieval scores from that system were used as predictors for item difficulty. While subsequent experiments showed that these predictors had low utility in the final models, exploiting question-answering systems in this way may be a promising direction for future work should the aforementioned hypothesis be true.

3. Experiments

The previous sections provided a brief overview of different approaches for extracting linguistic information from item text for the task of predicting item characteristics. In this section, we illustrate the approaches outlined earlier through several experiments for predicting different item characteristics.

Table 10.4 Example of a Practice Item

Item Stem	A 16-year-old boy is brought to the emergency department because of a 2-day history of fever, nausea, vomiting, headache, chills, and fatigue. He has not had any sick contacts. He underwent splenectomy for traumatic injury at the age of 13 years. He has no other history of serious illness and takes no medications. He appears ill. His temperature is 39.2°C (102.5°F), pulse is 130/min, respirations are 14/min, and blood pressure is 110/60 mm Hg. On pulmonary examination, scattered crackles are heard bilaterally. Abdominal examination shows a well-healed midline scar and mild, diffuse tenderness to palpation. Which of the following is the most appropriate next step in management?
Item Options	a. Antibiotic therapy* b. Antiemetic therapy c. CT scan of the chest d. X-ray of the abdomen e. Reassurance

Note: The asterisk denotes the correct answer, also known as item key.
Source: www.usmle.org/sites/default/files/2021-10/Step2_CK_Sample_Questions.pdf

3.1 Data

Data were collected between 2010 and 2015 and comprised approximately 19,000 pretest MCQs from the Step 2 Clinical Knowledge component of the United States Medical Licensing Examination (USMLE), an exam sequence taken by medical doctors as a requirement for licensure in the United States. Each exam included unscored pretest items that were presented alongside scored items. Test-takers had no way of knowing which items were scored and which were unscored pretest items. On average, each item was answered by 335 first-time examinees who were medical students from accredited[16] U.S. and Canadian medical schools.

An example test item from this exam is shown in Table 10.4. All items tested medical knowledge and were written by experienced item writers following a set of guidelines that specified a standard structure and prohibited the use of verbose language, extraneous material not needed to answer the item, information designed to mislead the test-taker, and grammatical cues such as correct answers that are longer or more specific than other options. Standards for style were also imposed, including consistent vocabulary and consistent formatting and presentation of numeric data.

Several item characteristics were computed for these items based on the responses received during pretesting. These included:

- *P-value:* The proportion of correct responses for a given item computed as:

$$p_i = \frac{\sum_{n=1}^{N} u_n}{N}$$

  where p_i is the p-value for item i, u_n is the 0–1 score (incorrect-correct) on item i earned by examinee n, and N is the total number of examinees in the sample.
- *Mean response time:* The average time (measured in seconds) that examinees spent viewing an item.
- r_b : The biserial correlation coefficient between examinees' responses on the given item and examinees' total test score. For a given item i, this may be calculated as follows:

$$r_{b_i} = \frac{(\mu_+ - \mu_X)}{\sigma_X}\left(p_i / y\right),$$

where μ_+ is the mean examinee test score *for those examinees responding correctly to item i*; μ_X is the mean examinee test score for all examinees; σ_X is the standard deviation of these scores; p_i retains its meaning from earlier; and y is the ordinate of the standard normal curve at the z-score associated with p_i. Equivalently, r_{b_i} may be expressed as the product of the Pearson product moment coefficient (between the examinee item score and test score) and

$$y^{-1}\sqrt{p_i(1-p_i)}.$$

3.2 Predictors

We report results based on each of the three types of predictors described in Section 2: human-engineered linguistic features, non-contextualized embeddings, and contextualized embeddings. The human-engineered linguistic features used here include several levels of linguistic processing and are summarized in Table 10.5. A complete list, including details on their computation, is found in Baldwin et al. (2021) and Yaneva et al. (2021).

Word2Vec (300 dimensions) was used for non-contextualized embeddings as described in Section 2.

Several models for contextualized embeddings were investigated. These included BERT Base and BERT Large (Devlin et al., 2018; described in Section 2) trained on the BooksCorpus (800 million words; Zhu et al., 2015) and English Wikipedia (2,500 million words). In addition to the BERT Base models trained on generic data, we also use a BERT Base model trained on clinical text from PubMed Central[17] and MEDLINE abstracts[18] (as in Gu et al., 2020). Training two separate BERT Base models on two types of data – generic and biomedical – allows for direct comparison of the effects of data domain for model performance on our task.

Finally, we also report results using RoBERTa (Liu et al., 2019), which is based on a more advanced architecture. This model was trained on the biomedical data described earlier to evaluate the effects of model architecture on performance. As mentioned in Section 2, RoBERTa represents a version of BERT with improved hyperparameter tuning.

Table 10.5 Summary of the Human-Engineered Linguistic Features by Level of Linguistic Processing

Linguistic Processing Level	Feature Count	Examples
Lexical	5	Word count; Average word length in syllables; Complex word count
Syntactic	29	Part of speech (POS) count; Average sentence length; Average number of words before the main verb Passive-active ratio
Semantic	11	Polysemic word count; Average senses for nouns; Average senses for verbs
Readability[a]	7	Flesch Reading Ease; Flesch-Kincaid grade level; Automated Readability Index; Gunning Fog
Cognitive[b]	14	Concreteness ratings; Imageability ratings; Familiarity ratings
Frequency[c]	10	Average word frequency; Words not in the first 2,000 most common words; Words not in the first 4,000 most common words
Cohesion	5	Temporal connectives count; Causal connectives count; Referential pronoun count
Specialized Clinical Features	8	Unified Medical Language System Metathesaurus terms count [d]

[a] See Dubay (2004) for formula definitions.

[b] *Source:* MRC Psycholinguistic Database (Coltheart, 1981).

[c] *Source:* British National Corpus (Leech et al., 2014).

[d] UMLS; Number of terms in an item that appear in the UMLS Metathesaurus (Schuyler et al., 1993).

3.3 Analysis

Several models for predicting each of the three item characteristics – p-value, mean response time, and biserial correlation (r_b) – were constructed using the three classes of ancillary data just described (linguistic features, non-contextualized embeddings, and contextualized embeddings).[19] For each of these models, 80% of the data were used for model training and the remaining 20% was used as a test set. Item characteristics were estimated using pretest data, and these empirical values were treated as truth for the purpose of model evaluation. Root Mean Squared Error (RMSE) was calculated for each model's predicted item characteristics.

To allow comparisons between the linguistic features and embeddings, the prediction step was not part of the embedding architecture but rather was done separately using several regressor algorithms from Python's *scikit-learn* library (Pedregosa et al., 2011): *linear regression*, *support vector regressor (SVR)*, *elastic net*, and *random forests (RF)*.[20] For elastic net, the alpha value was varied as a study condition and included 0.01, 0.03, and 0.05; likewise, for RF, number of trees was varied as study condition and included 100, 200, 300, and 400. Elastic net and RF were selected for their variance reduction ability in datasets with large numbers of input features.

Predictions were compared with a ZeroR baseline. This baseline is computed by taking the mean of the dependent variable (i.e., p-value, mean response time, or r_b) for the training set and treating it as the predicted value for the items in the test set. Predictions would need to outperform this baseline to be considered potentially useful.

3.4 Results

The results for modeling p-value, mean response time, and r_b are presented in Figures 10.1, 10.2, and 10.3, respectively. As can be seen, the results show that the values of the variables extracted from item text contain signal that can be used to predict item characteristics. The p-value and mean response time parameters were predicted with a significant improvement over the ZeroR baseline (especially mean response time): RMSE of .218 compared to .241 for p-value and RMSE of 23.3 compared to 32.9 for mean response time. The r_b predictions were less successful – only showing a small improvement over baseline (RMSE of .152 compared to .159) – making it the most challenging parameter to model among the parameters reported here.

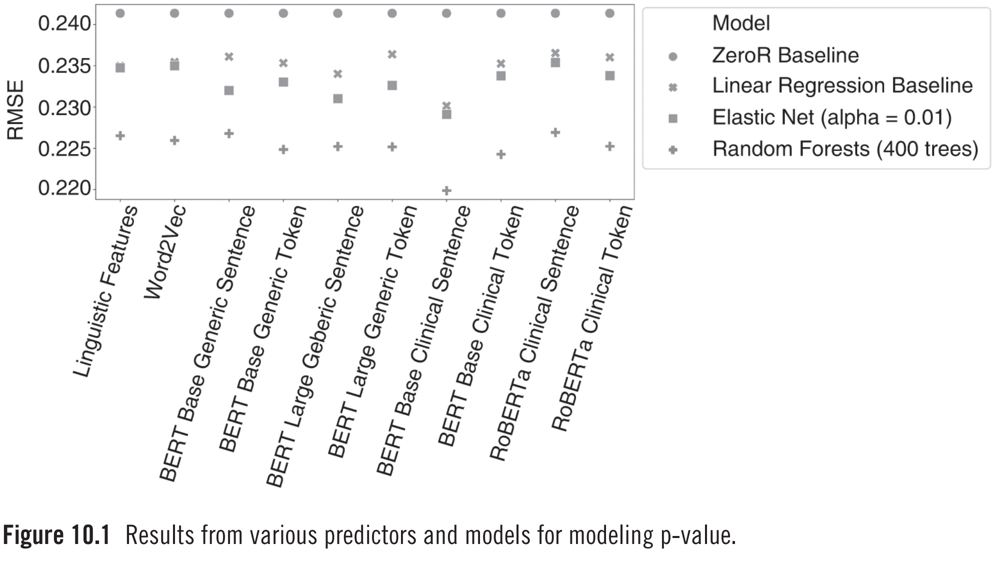


Figure 10.1 Results from various predictors and models for modeling p-value.

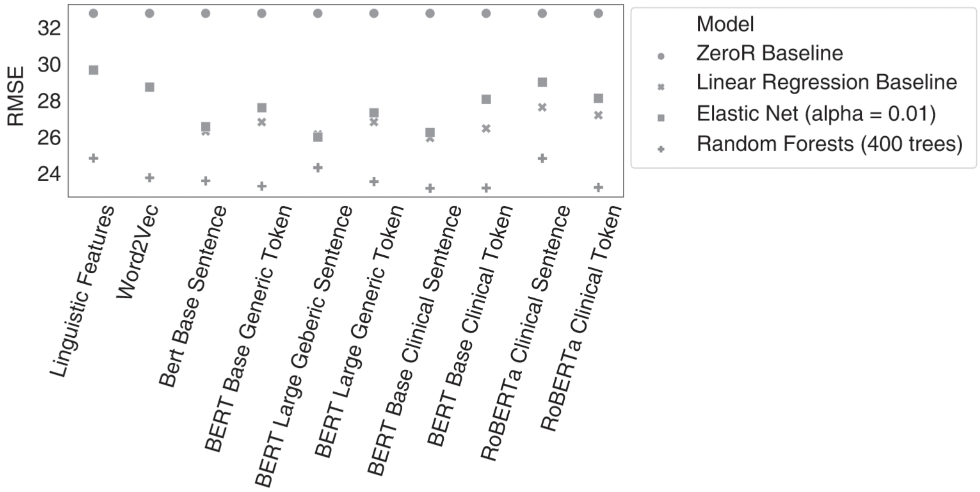


Figure 10.2 Results from various predictors and models for modeling mean response time.

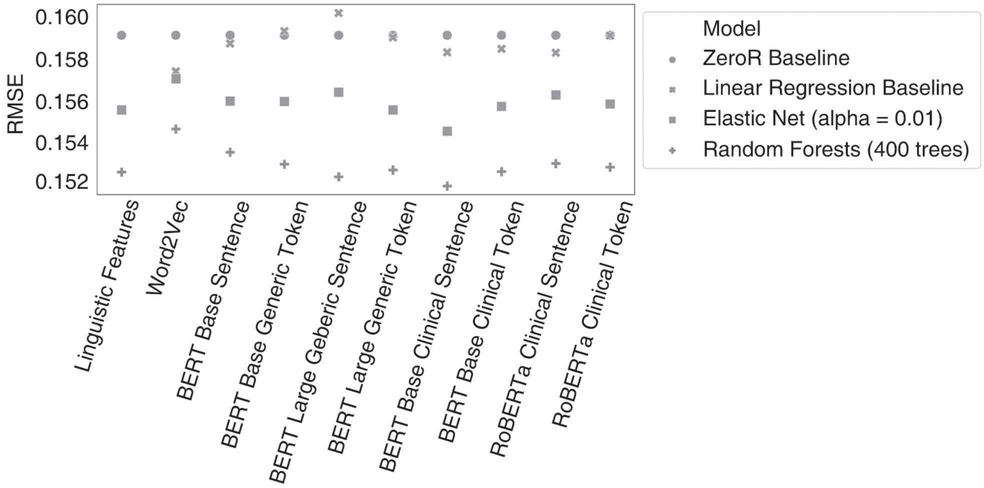


Figure 10.3 Results from various predictors and models for modeling r_b.

The sentence embeddings from the BERT Base model trained on clinical data performed best for predicting p-value and were among the best performing models for predicting response time and r_b. In contrast, models trained on generic data generally performed better with token embeddings than with sentence embeddings. The other model trained on clinical data, RoBERTa, did not perform as well as BERT Base. More than choice of model or training data, however, the regressor algorithm had the greatest effect on prediction quality, with random forests consistently outperforming all other models on all three tasks.[21]

The implications of these results for the different ways to extract signal from item text are discussed in the next section.

4. Discussion

This chapter set out to achieve two goals: to provide an overview of NLP approaches to item representation and to illustrate these approaches in the context of predicting item characteristics for clinical MCQs. The overview traced the transition from human-engineered linguistic features to non-contextualized and eventually contextualized embeddings; and the experimental results demonstrated that in general, small gains were associated with these historical developments. The experiments consistently showed that sentence embeddings from the BERT Base model trained on clinical data (BERT Base Clinical Sentence) were the best performing configuration for all tasks (although the improvements over some models were not always statistically significant).

It is conceivable that training models on generic data suffered because not all specialized medical terms from the clinical MCQs were present in the training corpus. While contextualized embeddings – like those produced by the BERT model – can process out-of-vocabulary data through subword modeling, this technique requires more computational resources[22] and may lead to lower performance. For example, as Gu et al. (2020) note, subword modeling of common medical terms such as *naloxone* first requires breaking naloxone into subword units (e.g., [na, ##lo, ##xon, ##e]) and then modeling it through these subwords. This is avoided when the training data comes from the biomedical domain, where common biomedical vocabulary such as *naloxone* are likely to be present. Gu et al. (2020) show that domain-specific pretraining using biomedical data can substantially outperform pretraining using generic data. Moreover, they note that even biomedical data alone outperforms generic + biomedical data, and they hypothesize that *the two domains are so different that negative transfer may occur if the representations are first learned on the generic data (i.e., performance may be hurt if the knowledge learned from the generic data does not apply to the specialized domain)*. The results presented here add more evidence in support of the hypothesis that domain-specific data result in superior performance. This may not be the case for predicting item characteristics in other assessment domains. Nevertheless, it highlights the use of domain-specific versus generic data as an important choice to investigate when pretraining models.

The results suggest that there is no straightforward way of deciding which architecture is best for a given task, as bigger and more robustly optimized models were not necessarily the best-performing ones. This was evident with both generic and biomedical training data, where comparisons between BERT Base and BERT Large – trained on the same generic data – did not provide clear evidence in support of the larger model; and comparison between BERT Base and RoBERTa, which were both trained on biomedical data, did not favor RoBERTa despite the additional hyperparameter tuning. This suggests that model size and parameter optimization are not necessarily the barrier preventing improvements. It is therefore advisable that researchers experiment with various architectures and make their final selection based on empirical results.

In terms of the type of encoded context, the fact that the sentence embedding from the BERT Base model trained on clinical data outperforms the token embedding from the same model shows that predicting item characteristics benefits from encoding larger context and longer dependencies as opposed to the shorter ones, characteristic of token embeddings. Since the BERT model is a bidirectional encoder[23] (as suggested by its name), it also shows the importance of capturing context from not only the prior tokens, but also the token that follows. It is interesting to observe that the sentence embeddings from RoBERTa perform consistently worse than those from BERT (and from RoBERTa token embeddings). This can be explained with a modification introduced in RoBERTa on how sentence embeddings are encoded. As described in Section 2, the BERT architecture is trained on two tasks: next token prediction and next sentence prediction. The authors of RoBERTa note that the next-sentence-prediction task, originally designed to improve performance on tasks that require reasoning about the relationships between pairs of sentences, could be removed, and that sentence embeddings can be generated without this objective. The superior performance of the BERT Base sentence embeddings suggests that reasoning about the relationships between pairs of sentences, learned from domain-specific data, is important to the task of predicting item characteristics.

One important reason for the comparative underutilization of embeddings in educational measurement is that the field is traditionally concerned with model interpretability, and embeddings offer very little information about the contribution of specific variables. While model interpretability is undoubtedly crucial for some applications, the trade-off between interpretability and accuracy may be more balanced in the area of predicting item characteristics (in fact, one may argue that accuracy is all that matters for improving pretesting or evaluating automatically generated items). Therefore, predicting item characteristics is one assessment area that can take better advantage of more sophisticated text representations such as embeddings. Should model interpretability be the focus, linguistic features, while generally not the highest-performing predictors, can provide greater insight into the relationships between various interpretable characteristics of items and various outcomes of interest. A good example of the value that linguistic features add beyond predictive performance is a study that uses linguistic features extracted from MCQs to gain insight into the cognitive complexity associated with answering the items and its relationship to item text (Yaneva et al., 2021). Apart from use cases where interpretability matters, our experiments support the now widely accepted idea that contextualized embeddings perform better than linguistic features and non-contextualized embeddings for many tasks.

The framework presented here has several limitations that merit discussion, mainly related to approaches that could have improved results but that were not shown here. These include combining different predictor types within a single model, or experimenting with other types of embedding models, including ones specifically developed for biomedical text. We also could have fine-tuned the models as opposed to using the embeddings they produced as input for a prediction model like random forests. While further research into each of these areas could be beneficial, the goal of this chapter was more modest: we merely focused on providing an overview of the most widely used, accessible, and well-known strategies in the field of NLP. This choice was motivated in part by the rapid pace of NLP research, which suggests that even bigger and more successful models for text representation will be in use by the time this chapter is published. It is our hope that a more foundational introduction to NLP approaches – particularly low-interpretability embeddings models – and their potential application to assessment problems will be more valuable to a non-NLP audience than a showcase of the latest language models. Readers are advised to view this chapter as a framework for item text representation rather than as a source of guidance about the most recent models and architectures.

The practical significance of these results will depend on specific use cases. For example, Baldwin et al. (2021) show that the use of the aforementioned approaches to predict item response time can help improve exam fairness. The results from this study indicate that if forms are assembled considering predicted response times for newly developed pretest items, overall timing variability for test forms can be reduced by 2 to 4 minutes. In contrast, the practical value of predicted p-values and Rb parameters has not yet been demonstrated, and the current results are most useful as an exploration of the type of predictive power different features or representations have with a view to optimizing the results.

One area for future research relates to recent advances in automated question answering such as those utilizing the T5 transformer model (Raffel et al., 2019). As was noted in Section 2, it may be that items that are more difficult for humans to answer also may be more difficult for machines to answer, and this relationship could be used to predict item difficulty and response time. Even if improvements in automated question answering do not lead to improvements in item parameter modeling on their own, predictors related to machine performance could potentially complement, rather than overlap with, the other approaches described in this chapter. While not yet explored in depth, combining signals from multiple sources in this way may be a promising direction for future research in modeling item characteristics.

5. Conclusion

This chapter discussed the evolution of text representation and demonstrated the use of three types of representations – linguistic features, non-contextualized word embeddings, and contextualized token and sentence embeddings – for the tasks of predicting p-values, mean response times, and r_b correlations for nearly 19,000 clinical MCQs. The empirical results suggested that, at least within the domain of clinical MCQs, it is beneficial to pre-train models on biomedical text, and that when this is done, encoding larger context and longer dependencies can improve results. Using larger models or ones with improved hyperparameter tuning does not necessarily lead to improved predictions and so, ideally, a range of architectures should be experimented with before selecting the best-performing one for a given problem. The task of modeling item parameters is one way in which the field of assessment can capitalize on the advances in text representation that have recently transformed the field of NLP and its numerous applications in our everyday lives.

Notes

1 Word embeddings, which are described in greater detail in Section 2.2, refer to several techniques for representing words based on their usage such that words with similar meanings have similar representations.

2 In most cases, the development of accurate NLP tools for the extraction of specific linguistic features first requires the availability of specific NLP resources. For example, part-of-speech-taggers are tools that automatically identify the parts of speech of words in a text, but their development and efficacy depend on the availability of resources like the Penn Treebank (Marcus et al., 1993), which contain large numbers of words manually labeled with their corresponding part of speech.

3 Words with more than one meaning.

4 Words may be divided into various subcomponents or *subwords*. For example, 'readable' can be divided into 'read' and 'able'.

5 How meaning is constructed in specific contexts.

6 Here, we describe the process using the skip-gram architecture with negative sampling, which generally works well with large datasets; however, its (in some sense) inverse architecture, *continuous bag of words*, also can be used.

7 Note that we have skipped the articles 'a' and 'the' during the preprocessing stage, where certain stopwords such as 'a', 'an', and 'the' are removed. This is optional depending on the application; however, in many tasks, stopwords are not highly informative for context.

8 The sampled distribution is sometimes called the *noise distribution*. Mikolov et al. (2013) suggest using the unigram distribution raised to the power of ¾, which reflects each word's frequency in the corpus, as the noise distribution.

9 The number of embeddings is usually defined empirically through trial and error.

10 Mikolov et al. (2013) propose 2–5 for large samples.

11 In the case of items, word embeddings must be pooled together. This can be done, for example, using element-wise averages, minimums, or maximums for all vectors (i.e., average-pooling, min-pooling, and max-pooling, respectively).

12 We note that some of the earlier contextualized embeddings such as ELMo (Peters et al., 2018) are not generated using transformer models.

13 https://github.com/google-research/bert

14 www.nlm.nih.gov/medline/medline_overview.html

15 Automated Question Answering is an NLP application, where the goal is to develop systems that can automatically answer various types of questions, including open-ended ones in reading comprehension exams, fora, or search queries; MCQs; true or false questions, etc.

16 Accredited by the Liaison Committee on Medical Education (LCME).

17 www.ncbi.nlm.nih.gov/pmc/

18 www.nlm.nih.gov/medline/medline_overview.html

19 As noted earlier, we use the embeddings as input to classic machine learning algorithms rather than finetuning BERT and RoBERTa models. This is done for two reasons: (1) preliminary experiments showed that this approach leads to better results for our data; and (2) this allows for fairer comparisons between models (especially with linguistic features, which cannot be fine-tuned).

20 Regressor algorithms were used with default parameters.

21 Performance improved with increases in the number of trees up to 400; further increases did not lead to additional meaningful gains (e.g., p-value RMSE of .216 with 1,000 trees compared to .218 with 400 trees).

22 This limitation can be addressed if the necessary computational resources are available, which may not always be the case in practice for many research institutions.

23 Models that learn information from left to right and from right to left.

References

Alsentzer, E., Murphy, J. R., Boag, W., Weng, W. H., Jin, D., Naumann, T., & McDermott, M. (2019). Publicly available clinical BERT embeddings. *arXiv*. https://arxiv.org/abs/1904.03323

Baldwin, P., Yaneva, V., Mee, J., Clauser, B. E., & Ha, L. A. (2021). Using natural language processing to predict item response times and improve test construction. *Journal of Educational Measurement*, *58*(1), 4–30.

Benedetto, L., Aradelli, G., Cremonesi, P., Cappelli, A., Giussani, A., & Turrin, R. (2021, April). *On the application of Transformers for estimating the difficulty of multiple-choice questions from text* (pp. 147–157). Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications.

Benedetto, L., Cappelli, A., Turrin, R., & Cremonesi, P. (2020, March). *R2DE: A NLP approach to estimating IRT parameters of newly generated questions* (pp. 412–421). Proceedings of the Tenth International Conference on Learning Analytics & Knowledge.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., & Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, *33*, 1877–1901.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., & Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments. *arXiv*. https://arxiv.org/abs/2006.09882

Choi, I. C., & Moon, Y. (2020). Predicting the difficulty of EFL tests based on corpus linguistic features and expert judgment. *Language Assessment Quarterly*, *17*(1), 18–42.

Clark, K., Luong, M. T., Le, Q. V., & Manning, C. D. (2020). Electra: Pre-training text encoders as discriminators rather than generators. *arXiv*. https://arxiv.org/abs/2003.10555

Coltheart, M. (1981). The MRC psycholinguistic database. *The Quarterly Journal of Experimental Psychology Section A*, *33*(4), 497–505.

Conneau, A., & Kiela, D. (2018). Senteval: An evaluation toolkit for universal sentence representations. *arXiv*. https://arxiv.org/abs/1803.05449

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., & Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. *arXiv*. https://arxiv.org/abs/1705.02364

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv*. https://arxiv.org/abs/1810.04805

DuBay, W. H. (2004). *The principles of readability*. Online Submission. DoE.

Firth, J. R. (1957). A synopsis of linguistic theory 1930–1955. In *Studies in linguistic analysis* (pp. 1–32). Philological Society. Reprinted in Palmer, F. R. (Ed.). (1968). *Selected papers of J. R. Firth 1952–1959*. Longman.

Gu, Y., Tinn, R., Cheng, H., Lucas, M., Usuyama, N., Liu, X., Naumann, T., Gao, J., & Poon, H. (2020). Domain-specific language model pretraining for biomedical natural language processing. In *ACM transactions on computing for healthcare (HEALTH)*. ACM.

Ha, L. A., Yaneva, V., Baldwin, P., & Mee, J. (2019, August). *Predicting the difficulty of multiple choice questions in a high-stakes medical exam* (pp. 11–20). Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications.

Harris, Z. (1954). Distributional structure. *Word*, *10*(23), 146–162.

He, P., Liu, X., Gao, J., & Chen, W. (2020). DeBERTA: Decoding-enhanced BERT with disentangled attention. *arXiv*. https://arxiv.org/abs/2006.03654

Kurdi, G. R. (2020). *Generation and mining of medical, case-based multiple choice questions*. The University of Manchester.

Leech, G., & Rayson, P. (2014). *Word frequencies in written and spoken English: Based on the British national corpus*. Routledge.

Leo, J., Kurdi, G., Matentzoglu, N., Parsia, B., Sattler, U., Forge, S., Donato, G., & Dowling, W. (2019). Ontology-based generation of medical, multi-term MCQs. *International Journal of Artificial Intelligence in Education*, *29*(2), 145–188.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). Roberta: A robustly optimized BERT pretraining approach. *arXiv*. https://arxiv.org/abs/1907.11692

Loukina, A., Yoon, S. Y., Sakano, J., Wei, Y., & Sheehan, K. (2016, December). *Textual complexity as a predictor of difficulty of listening items in language proficiency tests*. Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers.

Marcus, M., Santorini, B., & Marcinkiewicz, M. (1993). Building a large annotated corpus of English: The Penn treebank. *Computational linguistics-Association for Computational Linguistics (Print)*, *19*(2), 313–330.

McDonald, S., & Ramscar, M. (2001). *Testing the distributional hypothesis: The influence of context on judgements of semantic similarity* (Vol. 23, No. 23). Proceedings of the Annual Meeting of the Cognitive Science Society.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 3111–3119.

Miller, G. A. (1995). WordNet: A lexical database for English. *Communications of the ACM*, *38*(11), 39–41.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, *12*, 2825–2830.

Pennington, J., Socher, R., & Manning, C. D. (2014, October). *Glove: Global vectors for word representation* (pp. 1532–1543). Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP).

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). *Deep contextualized word representations* (pp. 2227–2237). Proceedings of NAACL-HLT.

Pilehvar, M. T., & Camacho-Collados, J. (2020). Embeddings in natural language processing: Theory and advances in vector representations of meaning. *Synthesis Lectures on Human Language Technologies*, *13*(4), 1–175. Morgan & Claypool Publishers.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, *1*(8), 9.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv*. https://arxiv.org/abs/1910.10683

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv*. https://arxiv.org/abs/1910.01108

Schuyler, P. L., Hole, W. T., Tuttle, M. S., & Sherertz, D. D. (1993). The UMLS metathesaurus: Representing different views of biomedical concepts. *Bulletin of the Medical Library Association*, *81*(2), 217.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., & Rush, A. M. (2020, October). *Transformers: State-of-the-art natural language processing* (pp. 38–45). Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations.

Xue, K., Yaneva, V., Runyon, C., & Baldwin, P. (2020, July). *Predicting the difficulty and response time of multiple choice questions using transfer learning* (pp. 193–197). Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications.

Yaneva, V., Ha, L. A., Baldwin, P., & Mee, J. (2020, May). *Predicting item survival for multiple choice questions in a high-stakes medical exam* (pp. 6812–6818). Proceedings of the 12th Language Resources and Evaluation Conference.

Yaneva, V., Jurich, D., Ha, L. A., & Baldwin, P. (2021, April). *Using linguistic features to predict the response process complexity associated with answering clinical MCQs* (pp. 223–232). Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). *Aligning books and movies: Towards story-like visual explanations by watching movies and reading books* (pp. 19–27). Proceedings of the IEEE International Conference on Computer Vision.

11

Stealth Literacy Assessment

Leveraging Games and NLP in iSTART

Ying Fang, Laura K. Allen, Rod D. Roscoe, and Danielle S. McNamara

Literacy can be broadly defined as 'the ability to understand, evaluate, use, and engage with written texts to participate in society, to achieve one's goals, and to develop one's knowledge and potential' (OECD, 2013). Literacy skills are essential for students to succeed in educational contexts and in nearly all aspects of everyday life (NICHD, 2000; Powell, 2009). However, improving these skills continues to be a challenging task in the United States. According to the 2019 National Assessment of Educational Progress, 27% of 8th grade students perform below the basic levels of reading comprehension, and 66% do not reach proficient levels. Assessments of 12th graders indicate a similar pattern, with 30% and 63% of students not reaching basic and proficiency levels, respectively.

Literacy assessments have been used extensively to help improve students' skills by providing targeted information about where students may struggle the most. In particular, accurate, valid, and reliable assessments of literacy skills are critical in order to provide students with opportunities for individualized instruction and practice, as well as timely feedback (Ingebrand & Connor, 2016; Kellogg & Raulerson, 2007). However, traditional literacy assessments typically occur *after* students have finished reading a text or writing an essay, thus rendering the delivery of timely feedback nearly impossible. In contrast to traditional approaches to assessment, *stealth assessment* offers an innovative method to assess students' literacy skills *during* learning. This type of assessment is a type of user modeling wherein the assessment is seamlessly woven into game-based learning environments to assess students unobtrusively (Shute & Ventura, 2013; Wang et al., 2015). Specifically, the evaluation of students' abilities occurs during the learning activity, rather than at summative or 'checkpoint' assessments. In addition, stealth assessments are not presented as 'quizzes' or 'tests', but instead are based on students' behaviors and performance during the tasks themselves. As such, stealth assessments can assess students' literacy dynamically and provide timely feedback throughout the learning process.

In this chapter, we describe and analyze the feasibility of implementing stealth assessment in a game-based learning environment to evaluate students' literacy skills. We first describe literacy assessments and how natural language processing (NLP) techniques have been used to assess literacy. We then provide an overview of stealth assessment and its application in digital environments to assess students' knowledge and skills. We next summarize the Interactive

DOI: 10.4324/9781003278658-15

This chapter has been made available under a CC-BY-NC-ND 4.0 license.

Strategy Training for Active Reading and Thinking (iSTART), a game-based intelligent tutoring system (ITS) designed to help students improve their reading comprehension. We describe how NLP methods embedded in iSTART are used to assess students' skills and guide the adaptation of the system (e.g., providing individualized feedback and customizing learning paths). Finally, we report two preliminary analyses demonstrating how NLP can be used to develop stealth assessments of students' literacy skills in iSTART to guide the macro-level adaptivity of the system.

1. Natural Language Processing and Its Applications

Reading comprehension is a complex process of interpreting and extracting meaningful information from written text. This endeavor requires several cognitive abilities and language skills, ranging from simple word recognition to deeper language comprehension (Vellutino, 2003). The Construction-Integration Model (Kintsch, 1988; van Dijk & Kintsch, 1983) of discourse comprehension posits that readers construct multiple levels of understanding. Individuals must integrate information explicitly stated in the text, information implied by the text, lexical knowledge, their domain knowledge, and relevant world knowledge to build a coherent representation of the text. The resulting situation model is a higher-level representation of text that captures semantic meaning (rather than specific words) and connects text information to prior knowledge. Importantly, readers construct mental models of a text *while* reading (Kintsch, 1988), and these mental models change dynamically as the properties of the text change across sentences (McNamara & Magliano, 2009).

The information contained in language (e.g., linguistic features) provides an important window into readers' underlying literacy skills and how they process language (McNamara, 2021). A powerful technique to extract and analyze the information contained in language is natural language processing (NLP) – the computerized approach to analyzing text based on a set of theories and technologies (Bird et al., 2009). NLP includes a broad category of methods used for different levels of language analysis such as speech recognition, lexical analysis, syntactic analysis, semantic analysis, and discourse analysis (Burstein, 2003; D'Mello et al., 2011; Elliot, 2003; Litman et al., 2006; McNamara et al., 2018). Speech recognition focuses on decomposing a continuous speech signal into a sequence of known words. Lexical analysis aims at interpreting the meaning of individual words, such as assigning a single part-of-speech tag to each word. Syntactic analysis focuses on the words in a sentence to reveal the grammatical structure of the sentence. This level of processing usually computes a representation of a sentence which demonstrates structural dependency relationships between the words. Semantic processing assesses the possible meanings of a sentence by focusing on the interactions among word-level meanings in the sentence. Finally, discourse analysis focuses on the nature of the discourse relations between sentences and how context impacts sentential interpretations.

NLP methods have been widely used in computer-based learning and assessment systems to support the analyses of student-constructed responses (Graesser, & McNamara, 2012; Landauer et al., 2007; Shermis et al., 2010). For example, NLP applications are used in the context of computer-based assessments of explanations and think-alouds during reading (Magliano et al., 2011), essay grading (Burstein, 2003; Landauer et al., 2003; Shermis et al., 2010), grading short answer questions (Leacock & Chodorow, 2003), and conversation-based ITSs that require students to produce written (i.e., typed) responses during interactive conversations (Graesser et al., 2008). These automated systems incorporate a variety of NLP tools and algorithms to assess the responses generated by students and to make inferences about their knowledge and skills.

2. Stealth Assessment

Stealth assessment refers to assessments that are based on evidence-centered design and woven directly and invisibly into gaming environments (Shute & Moore, 2017; Shute & Ventura, 2013). Unlike traditional assessments, test items are replaced with gaming tasks and activities such that learners may be largely unaware of being assessed. When students perform game tasks, they naturally produce rich sequences of actions, and these actions and performance during the gameplay become the evidence needed for knowledge and skills assessment.

Stealth assessments were initially proposed and used to measure higher-order skills such as creativity, problem-solving, and collaboration, which are essential for students to succeed professionally and personally (Partnership for 21st Century Learning, 2019) yet difficult to measure using traditional assessments (Wang et al., 2015). For example, stealth assessment has been embedded in a game called *Use Your Brainz* – a slightly modified version of a popular commercial game, *Plants vs. Zombies 2* – to evaluate students' problem-solving skills (Shute, Wang, et al., 2016). Specifically, students produce a dense stream of performance data during the gameplay. These data are recorded by the game system in a log file and analyzed to infer the students' problem-solving skills. Another example is *Physics Playground*, a computer game that emphasizes 2-D physics simulations. Performance-based stealth assessments are embedded in the game to evaluate students' persistence and creativity (Shute & Rahimi, 2021; Ventura & Shute, 2013; Wang et al., 2015).

In addition to domain-general skills, stealth assessments can also be used to evaluate domain-specific knowledge and skills. In *Physics Playground*, embedded stealth assessments also evaluate students' physics knowledge (Shute, Leighton, et al., 2016; Shute, Wang, et al., 2016). In *Portal 2*, a 3D puzzle-platform video game, stealth assessments are used to measure students' spatial skills (Shute et al., 2015). In sum, stealth assessments have been implemented in diverse gaming environments to evaluate students' domain knowledge and skills as well as higher-order skills that can be applied across domains.

3. iSTART: A Game-Based Learning Environment for Reading

This chapter aims to develop stealth assessments within the context of Interactive Strategy Training for Active Reading and Thinking (iSTART). iSTART is a game-based ITS that provides adaptive instruction and training to help students learn reading comprehension strategies. The game design of iSTART provides a means for stealth assessment of literacy. iSTART originated from a successful classroom-based intervention named Self-Explanation Reading Training (SERT; McNamara, 2004, 2017), which teaches students how to self-explain while reading using comprehension strategies (i.e., comprehension monitoring, paraphrasing, predicting, bridging, and elaborating). Recently, iSTART has been expanded to include strategy training modules to learn and practice summarization and question asking (Johnson et al., 2017; Ruseti et al., 2018).

iSTART instruction includes video lessons and two types of practice: regular practice (i.e., coached practice) and game-based practice. Video lessons provide students with information about comprehension strategies and prepare them for the practice. During coached practice, students generate typed self-explanations for several target sentences. iSTART leverages NLP algorithms to analyze the self-explanations and provide rapid formative feedback. Specifically, the NLP algorithms implemented in iSTART use both word-based indices and latent semantic analysis (LSA) to identify the strategies used in the self-explanations. The algorithm provides a holistic score for the quality of the self-explanation on a scale of 0 ('poor') to 3 ('great'), as well as actionable feedback to help students revise their self-explanations when their scores are below a certain threshold (McNamara et al., 2007).

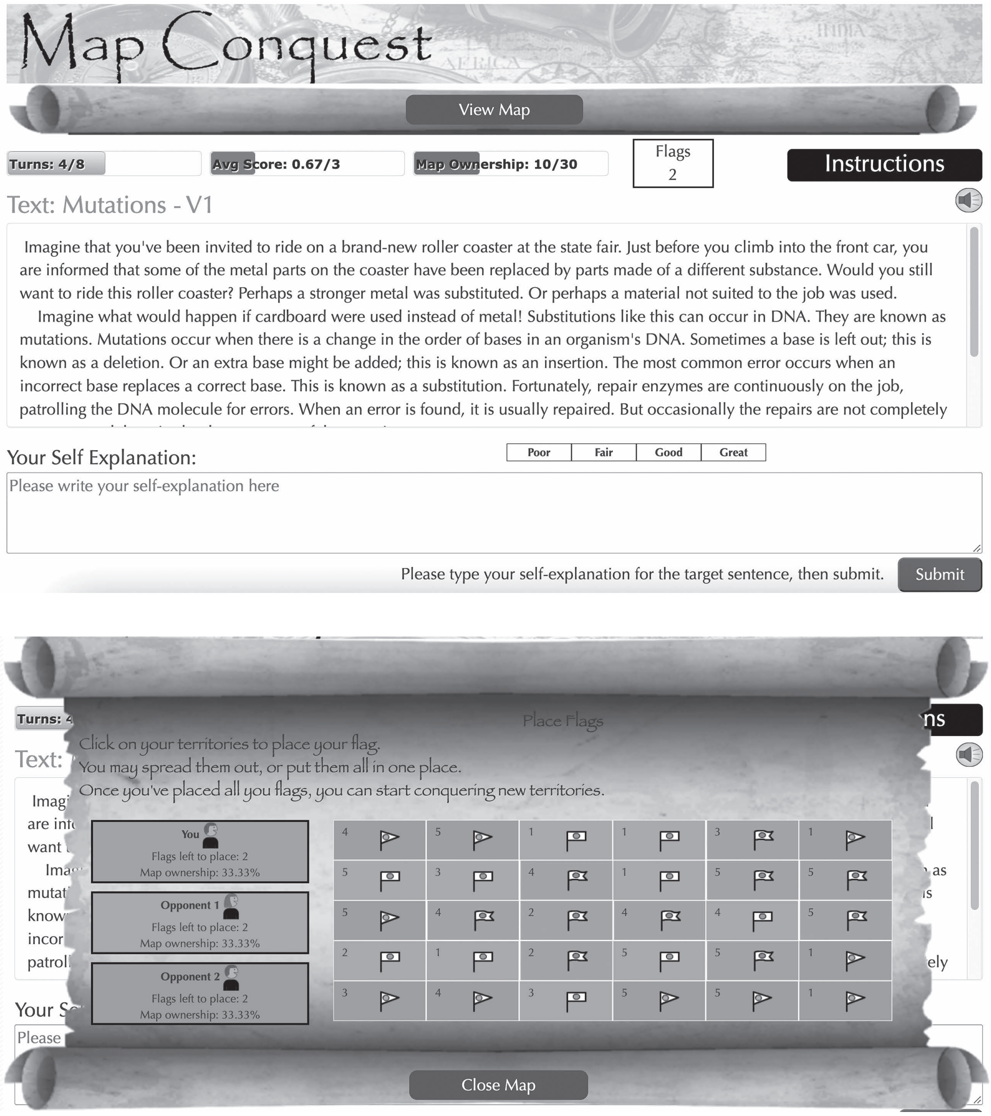
Game-based practice in iSTART was later developed to increase motivation and engagement (Jackson & McNamara, 2011, 2013). In iSTART games, students can practice reading strategies by exploring game narratives, overcoming game challenges, and interacting with game characters. iSTART generates immediate feedback during or after gameplay to help students identify their strengths and weaknesses while keeping them engaged (Jackson & McNamara, 2013). The feedback is based on the real-time assessment of students' gameplay (i.e., performance), and the assessment methods vary between two types of games in iSTART. Identification games integrate multiple-choice questions where correct and incorrect answer choices are carefully constructed to diagnose students' understanding or confusion. The assessment of students' performance occurs by matching students' selection with predetermined answers. In contrast, generative games embed open-ended questions during which students write constructed responses in the form of text, and NLP methods are used to evaluate the text-based answers (Johnson et al., 2018; McCarthy, Watanabe, et al., 2020; McNamara, 2021). Figure 11.1 shows a generative game in iSTART during which students self-explain target sentences to earn flags to put on the map and conquer neighboring lands.

4. Adaptivity in iSTART Through NLP

iSTART implements both inner-loop and outer-loop adaptivity to customize instruction to individual students. Inner-loop feedback refers to the immediate feedback students are given when they complete an individual task, and outer-loop adaptivity refers to the selection of subsequent tasks based on students' past performance (VanLehn, 2006). In iSTART regular practice and game-based practice, NLP methods have been used to facilitate inner-loop and outer-loop adaptivity.

Regarding inner-loop adaptivity, iSTART implements NLP and machine learning algorithms to assess self-explanations and then provide holistic scores and actionable, individualized feedback. The algorithm specifically relies on LSA, which is a mathematical method for representing the contextual-usage meaning of words and text segments (Landauer et al., 2007). It provides the ability to computationally represent semantic relations between ideas in text (McNamara, 2011). iSTART leverages word-based algorithms combined with LSA to drive formative feedback that guides readers on how to improve their self-explanations (McNamara, 2021; McNamara et al., 2007).

To further promote skill acquisition, iSTART complements the inner loop with outer-loop adaptations, which select practice texts based on the student model and the instruction model. An ITS typically employs three elements to assess students and select appropriate tasks: the domain model, the student model, and the instructional model (Shute & Psotka, 1996; VanLehn, 2006; Woolf, 2010). The domain model represents ideal expert knowledge and may also address common student misconceptions. The domain model is usually created using detailed analyses of the knowledge elicited from subject matter experts. The student model represents students' current understanding of the subject matter, and it is constructed by examining student task performance in comparison to the domain model. Finally, the instructional model represents the instructional strategies and is used to select instructional content or tasks based on inferences about student knowledge and skills. iSTART creates student models using students' self-explanation scores and scores on multiple-choice measures. The instructional model then determines the features of each presented task (i.e., text difficulty and scaffolds to support comprehension) using the evolving student model. For example, subsequent texts become more difficult if students' self-explanation quality on prior texts is higher. Conversely, when students' self-explanation quality is lower, the subsequent texts become easier (Balyan et al., 2020; Johnson et al., 2018). In summary, NLP methods are used in iSTART to facilitate both individualized inner-loop feedback and outer-loop task selection.



Text: Mutations - V1

Imagine that you've been invited to ride on a brand-new roller coaster at the state fair. Just before you climb into the front car, you are informed that some of the metal parts on the coaster have been replaced by parts made of a different substance. Would you still want to ride this roller coaster? Perhaps a stronger metal was substituted. Or perhaps a material not suited to the job was used.

Imagine what would happen if cardboard were used instead of metal! Substitutions like this can occur in DNA. They are known as mutations. Mutations occur when there is a change in the order of bases in an organism's DNA. Sometimes a base is left out; this is known as a deletion. Or an extra base might be added; this is known as an insertion. The most common error occurs when an incorrect base replaces a correct base. This is known as a substitution. Fortunately, repair enzymes are continuously on the job, patrolling the DNA molecule for errors. When an error is found, it is usually repaired. But occasionally the repairs are not completely

Your Self Explanation:

Please write your self-explanation here

Please type your self-explanation for the target sentence, then submit.

Click on your territories to place your flag.
You may spread them out, or put them all in one place.
Once you've placed all you flags, you can start conquering new territories.

You
Flags left to place: 2
Map ownership: 33.33%

Opponent 1
Flags left to place: 2
Map ownership: 33.33%

Opponent 2
Flags left to place: 2
Map ownership: 33.33%

4	5	1	1	3	1
5	3	4	1	5	5
5	4	2	4	4	5
2	1	2	5	5	1
3	4	3	5	5	1

Figure 11.1 Screenshots of a generative game – *Map Conquest*

5. Stealth Literacy Assessment in iSTART Using NLP

Current assessments enable iSTART to provide real-time feedback and customize learning tasks within specific games or practice. However, such assessments are task specific and may not transfer to games that incorporate different tasks or focus on other strategies. For example, students' scores on a question-asking game might not strongly correlate with their summarization skills, and thus cannot be used for the task selection between question-asking and summarization modules. As such, a task-general literacy assessment may be necessary for macro-level adaptation across iSTART modules.

Implementing stealth literacy assessments in iSTART also provides several benefits over more traditional assessments such as standardized tests. First, stealth assessments allow students to be continuously assessed without disrupting the learning process. This information then increases the opportunities for the system to adapt to students based on their specific pedagogical needs (VanLehn, 2006). In addition, stealth assessments are based on students' learning behaviors when they are engaged with learning tasks, rather than post-hoc measurements of performance. The moment-to-moment learning data can be collected by iSTART and used to assess students dynamically to capture their current state of literacy-related skills.

The potential for stealth literacy assessment in iSTART through NLP has been partially explored in prior research. For example, Allen and McNamara (2015) analyzed the lexical properties of college students' essays with TAALES, an NLP tool developed to examine the lexical properties of text (Kyle et al., 2018). Two linguistic features associated with the use of sophisticated and academic word use accounted for 44% of the variance in vocabulary knowledge scores. The findings suggest that students with greater vocabulary knowledge tend to produce essays with words acquired later in life and are more academic in nature. Similarly, Allen et al. (2015) used Coh-Metrix (McNamara et al., 2014) to calculate a set of descriptive (e.g., word count and average word length), lexical, syntactic, and cohesive indices of students' constructed responses during reading strategy training. The selected indices were used to predict students' scores in a standardized reading test. Three linguistic features (i.e., lexical diversity, semantic cohesion, and sentence length) explained 38% of variance in students' reading test scores. Their findings revealed that better readers tended to use a greater diversity of words and shorter sentences while maintaining the topic more cohesively in their self-explanations.

In contrast to previous studies that focused on either essays or self-explanations, McCarthy, Laura, et al. (2020) analyzed the linguistic properties of two types of constructed responses (i.e., self-explanation and explanatory retrievals) to compare their predictive power on reading comprehension test scores. The linguistic indices were computed using the Constructed Response Analysis Tool (CRAT, Crossley et al., 2016a). CRAT indices of self-explanations and explanatory retrievals accounted for 15% and 25% of the variance in students' comprehension test scores, respectively. The top five features in the self-explanation model included academic adjective keywords, magazine adjective keywords, fiction adjective keywords, news adjective keywords, and academic bigram keywords. In comparison, the top five variables in the explanatory retrieval model included academic bigram keywords, word imageability, academic keywords, age of acquisition for content words, and fiction keywords. Their findings indicated that the descriptive content (i.e., adjectives) of the self-explanations was most predictive of comprehension scores, whereas a wider variety of textual information, particularly lexical sophistication, were predictive of comprehension scores in explanatory retrievals.

Previous research found linguistic properties of students' constructed responses are predictive of their reading skills. However, those studies did not investigate whether the number of self-explanations affects the predictivity of the linguistic features. McCarthy, Laura, et al. (2020) explored the context of constructed responses (i.e., constructed-responses generated during reading vs. after reading), but the context was not related to iSTART training. Building on prior research, we conducted two studies using previously collected data to investigate the potential of stealth literacy (i.e., reading) assessment in iSTART through NLP methods. Specifically, we aimed to predict students' reading comprehension scores obtained from a standardized reading test using the linguistic characteristics of their self-explanations. If the linguistic properties are able to model students' reading test scores, they can serve as proxies of students' reading skills. As such, students' reading skills can be assessed during their gameplay and the assessment may guide the macro-level adaptivity of the system.

The two studies investigated three research questions: (1) To what extent can linguistic features of self-explanations predict students' reading skills as measured by a standardized reading test? (2) To what extent does the number of self-explanations affect the predictive power of linguistic features on reading skills? (3) To what extent does the context of the generated response (i.e., during training vs. before training) affect the degrees to which linguistic features predict reading skills? These research questions were investigated within Study 1, whereas Study 2 provided a 'test set' with a completely separate population of students.

5.1 Study 1

Study 1 data were collected in an investigation of the effects of metacognitive prompts in iSTART (see McCarthy et al., 2018). The original study included 121 current and recently-graduated high school students (51.2% Caucasian, 23.2% Hispanic, 11.6% African American, 7.4% Asian, and 6.6% who identified as other ethnicities; 62% female; $M_{age} = 17.59$ years) who were provided monetary compensation for their participation. The students were randomly assigned within a 2 (performance threshold vs. no threshold) by 2 (self-assessment vs. no self-assessment) between-subjects design yielding four conditions: threshold only, self-assessment only, threshold and self-assessment, and neither threshold nor self-assessment. The performance threshold and self-assessment features were two metacognitive elements implemented within iSTART generative practice. After students write their own self-explanations, an NLP algorithm immediately provides a score of 'poor' (0), 'fair' (1), 'good' (2), or 'great' (3). The performance threshold feature notifies students when their average self-explanation scores drop below 2. The self-assessment feature prompts students to rate the quality of their own self-explanations before receiving the computer-generated scores. In the current analysis, the self-explanations generated from both conditions were analyzed with the same procedure.

5.1.1 Procedure, Materials, and Measures

The participants attended five sessions within a laboratory setting. In the first session, participants completed a basic demographic questionnaire and pretests including a self-explanation test and a standardized reading test (i.e., Gates-MacGinitie Reading Test; MacGinitie & MacGinitie, 1989). For the next three sessions (two hours each), participants completed a series of activities in iSTART, including watching videos and practicing self-explanation by playing games.

In the self-explanation pretest, students self-explained target sentences in one of two texts (i.e., 'Heart Disease' or 'Red Blood Cells') that have been used in previous iSTART studies (e.g., Jackson & McNamara, 2011; McCarthy et al., 2018). There were nine target sentences in each text, which resulted in nine self-explanations. Similarly, when participants played generative games during iSTART training, they self-explained multiple target sentences within each text. Participants varied in the total number of self-explanations they generated, but most participants completed two texts, which resulted in 12 self-explanations. The 12 self-explanations associated with the first two texts generated during iSTART training, together with the 9 self-explanations produced in the pretest, were the focus of our analysis.

Students' reading comprehension skills were measured using a modified version of the Gates-MacGinitie Reading Test (4th ed.) level 10/12 form S. This test assessed students' reading comprehension ability by asking students to read short passages and then answer two to six questions about the content of the passage. There were 48 multiple-choice questions, and students were required to answer as many questions as possible within 25 minutes. The questions were designed to measure students' ability to comprehend shallow and deep level information. Students' performance on the reading test was analyzed using the raw scores.

5.1.2 Data Processing

For the purpose of calculating the linguistic properties of students' self-explanations, we aggregated all of their individual, sentence-level self-explanations. Additionally, one goal of this study was to explore how the number of self-explanations affects the predictive power of linguistic features. Therefore, multiple aggregated self-explanations were created using different numbers of self-explanations. Another goal of the study was comparing the predictivity of self-explanation generated during training versus before training. For this purpose, we created aggregated self-explanations using the self-explanations generated in different contexts separately. More specifically, we created 13 aggregated self-explanations based upon the 12 self-explanations generated during iSTART training and 9 self-explanations generated during the pretest. Figure 11.2 further illustrates how the 13 aggregated self-explanations (i.e., Files 1–13) were created using the sentence-level self-explanations. The procedure was as follows: First, we created 6 text files (i.e., Files 1–6) incrementally using the 6 self-explanations students generated from the first text (i.e., 'Ecological Pyramids'). File 1 included the first self-explanation; File 2 included the first two self-explanations; File 3 included the first three self-explanations, and so on. Next, we incrementally appended the 6 self-explanations generated from the second text (i.e., 'Gravity') to File 6, and created Files 7–12. As such, Files 7–12 comprised 7–12 aggregated self-explanations. Finally, we aggregated the 9 self-explanations generated during pretest into one file (i.e., File 13).

5.1.3 *Text Analyses*

Linguistic features of students' self-explanations were analyzed using the Tool for the Automatic Analysis of Lexical Sophistication (TAALES; Kyle et al., 2018) and Coh-Metrix

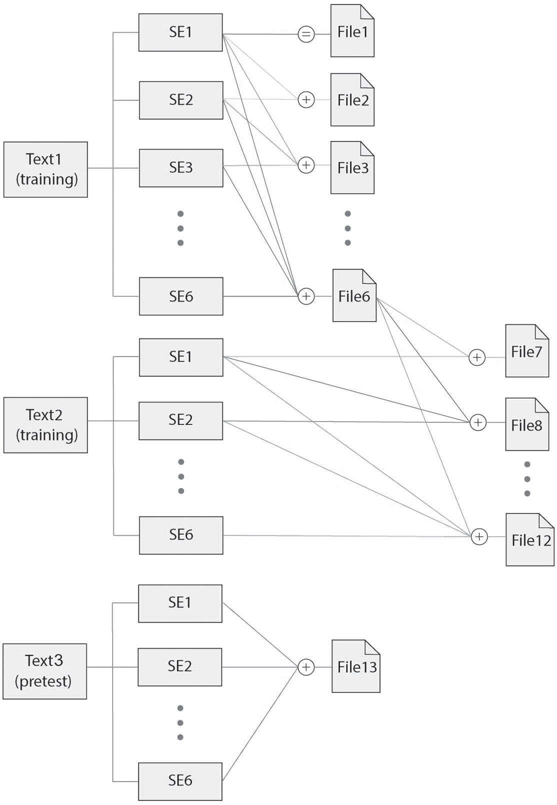


Figure 11.2 Visual illustration of the aggregated self-explanations created using 12 self-explanations generated during iSTART training and 9 self-explanations generated during pretest. Files 1–13 are the aggregated self-explanations used for further text analysis.

Note: SE = self-explanation

(McNamara et al., 2014). TAALES calculates over 200 lexical and phrasal features, such as lexical frequency (i.e., how often a word occurs in a reference corpus), n-gram frequency (i.e., how often an n-gram occurs in a reference corpus), psycholinguistic word information (e.g., familiarity, imageability, and concreteness), and word neighborhood information (e.g., the number of words in a word's orthographic neighborhood). In calculating indices related to frequency, various reference corpora are used, such as the SUBTLEXus corpus of subtitles (Brysbaert & New, 2009) and the Corpus of Contemporary American English (COCA; Davies, 2009). Coh-Metrix computes over 100 lexical, syntactic, and cohesive features. The lexical features describe the characteristics of the words that are found in a given text, such as lexical diversity (i.e., the variety of unique words in a text in relation to the total number of words). The syntactic features describe the complexity of the sentence constructions found within a text, such as the number of modifiers per noun phrase and the density of verb phrases. Cohesion features provide information about the type of connections that are made between ideas within a text, such as connectives (e.g., incidence of causal verbs and causal particles in a text), overlap in content words between sentences, and semantic overlap between sentences or paragraphs.

5.1.4 Statistical Analyses

To reduce the number of linguistic features and control for statistical assumptions, a series of pre-analytic pruning steps were undertaken. First, Pearson correlations were calculated between the features and students' reading test scores. Only features correlated higher than .20 with the reading test (i.e., Gates-MacGinitie Reading Test) scores were retained in further analysis. The procedure was performed iteratively across the 13 datasets. Second, the retained features from the 13 datasets were combined. We calculated the frequency at which each feature correlated with reading scores above .20. Thus, for a given feature, a score of '6' would indicate that the feature correlated with reading scores in 6 out of the 13 datasets. Only the features with a score greater than 6 (i.e., more than half of the datasets) were retained in the analysis. Finally, the remaining features were assessed for multicollinearity ($r > .70$). If two or more features demonstrated multicollinearity, only the feature that correlated most strongly with reading test scores was retained in the analysis.

A multiple linear regression analysis was performed to determine the extent to which the linguistic features successfully modeled students' reading test scores. The model was applied to the 13 datasets iteratively. The variance in the reading test scores explained by the linguistic features in different datasets was calculated and compared to determine the extent to which the number of self-explanations affected the predictive power of linguistic features on reading test scores. In addition, the variance explained by the linguistic features of pretest self-explanations was compared to the variance explained by the same number of training self-explanations. This latter analysis examined whether the context of generated responses (i.e., self-explanations) affected the degrees to which linguistic features predict reading skills.

5.1.5 Results

Thirteen features demonstrated consistently significant correlations with reading test scores. To avoid overfitting the model, we selected only five features that were significantly correlated with the reading test scores in the majority of the datasets (i.e., at least seven datasets). Table 11.1 shows the retained linguistic features (i.e., indices), their average correlations with the reading test scores across the 13 datasets, and the number of datasets in which a feature was significantly correlated with the reading test scores. The means and standard deviations of the five features in the dataset with 12 aggregated self-explanations are also shown in Table 11.1.

Table 11.1 Descriptive Statistics of the Five Selected Linguistic Features

NLP Tool	Linguistic Features	Average *r*	Datasets	Mean (SD)
TAALES	Word count (AW)	0.27	12	400.20 (117.87)
TAALES	BNC Written trigram Frequency (AW)	0.26	10	0.01 (0.00)
TAALES	Average distance of closest orthographic neighbors (CW)	0.31	9	1.94 (0.07)
Coh-Metrix	Causal verbs and casual particles incidence	0.32	10	41.21 (12.51)
Coh-Metrix	Temporal connectives incidence	0.24	7	19.73 (10.10)

Note: CW = Content Words, AW = All Words

A multiple linear regression analysis was performed to predict students' reading test scores with the five linguistic features. The model was applied to the 13 datasets iteratively, and the results were compared between datasets. As is shown in Table 11.2, four linguistic features were significant predictors of reading test scores in most of the datasets. Increasing the number of self-explanations improved the predictive power of the models. In addition, the predictive power of the training self-explanations was higher than that for the pretest self-explanations. Specifically, nine pretest self-explanations accounted for 21% of the variance in the reading test scores. The same number of training self-explanations accounted for 34% of the variance in the reading test scores.

5.1.6 Summary

In Study 1, we analyzed whether the linguistic properties of self-explanations could predict students' reading skills (i.e., standardized reading test scores). We also explored the extent to which this predictive relationship was affected by two factors: the number of self-explanations and the self-explaining context (i.e., during training vs. before training). The results are informative regarding when to implement stealth assessments and how many self-explanations are necessary for reliable assessments that might guide macro-level adaptivity in iSTART.

Several findings are worth noting. First, four linguistic features (i.e., *word count*, *trigram frequency*, *word neighbor distance*, and *casual verbs and particles*) were found to be significant predictors of reading test scores in most of the datasets. Additionally, the predictive power of the linguistic features *increased* as more self-explanations were included in the linguistic analysis. Finally, the linguistic features of self-explanations generated *during training* accounted for greater variance in the reading test scores compared to self-explanations generated before training. These promising results suggest the potential for efficient stealth assessment to evaluate students' reading skills in iSTART using NLP methods.

5.2 Study 2: Generalization to a New Dataset

Our second analysis further tested the value of linguistic features for estimating students' literacy skills (i.e., Gates-MacGinitie reading test scores) within a separate sample. Specifically, we obtained a new corpus of self-explanations generated by high school students and analyzed the linguistic properties of these self-explanations. We next calculated the scores of the five linguistic features previously identified in Study 1, and then performed regression models to predict students' reading test scores. Study 2 thus serves as a partial replication and attempts to generalize the findings from Study 1.

5.2.1 Test Corpus

The test corpus for Study 2 was obtained from an investigation of the effectiveness of an adaptive text selection algorithm in iSTART (McCarthy, Watanabe, et al., 2020). The original study included 113 current and recently graduated high school students (46% Caucasian, 33%

Table 11.2 Multiple Linear Regression Analysis Predicting Reading Test Scores With the Five Linguistic Features Across 13 Datasets

	Standardized Coefficients							
Dataset	Word Count	Trigram Frequency	Word Neighbor Distance	Causal Verbs and Particles	Temporal Connectives	*F*	*p*	R^2
1 SE	.23*	−.10	.13	.10	.02	1.85	.11	.08
2 SEs	.23*	−.03	.13	.15	.09	2.63	.03	.10
3 SEs	.27**	−.25**	.19*	.21*	.13	6.96	<.01	.23
4 SEs	.27**	−.27**	.21*	.19*	.13	8.24	<.01	.26
5 SEs	.25**	−.25**	.22**	.22**	.20*	9.51	<.01	.29
6 SEs	.26**	−.25**	.24**	.23**	.18*	10.52	<.01	.31
7 SEs	.24**	−.23*	.21*	.24**	.20*	7.99	<.01	.31
8 SEs	.25**	−.21*	.26**	.22*	.18*	8.28	<.01	.31
9 SEs	.26**	−.22*	.29**	.19*	.18*	9.19	<.01	.34
10 SEs	.21**	−.20*	.27**	.25**	.13	9.31	<.01	.34
11 SEs	.23**	−.19*	.34***	.21*	.17	10.24	<.01	.36
12 SEs	.22*	−.20*	.37***	.21*	.18*	11.43	<.01	.39
PreSEs	.30***	−.07	.26**	.13	.13	6.09	<.01	.21

Note: SE = self-explanation, PreSE = pretest self-explanation, * $p < .05$, ** $p < .01$, *** $p < .00$

Hispanic, 7% African American, 7% Asian, and 7% self-identified as other ethnicities; 74% female; M_{age} = 16.27 years) from the southwestern United States. Participating students were randomly assigned to one of two conditions: (1) iSTART training with *random* text selection; or (2) iSTART training with *adaptive* text selection. Participants generated self-explanations during iSTART training in both conditions. The current study analyzed the linguistic properties of the self-explanations and their association with standardized reading test scores.

5.2.2 Procedure, Materials, and Measures

All participants completed a pretest including a standardized reading test (i.e., Gates-MacGinitie Reading Test) and a self-explanation test. Participants then completed three 2.5-hour training sessions in iSTART, which included lesson videos, coached practice, and practice games wherein texts were presented randomly or adaptively. Participants generated self-explanations during coached practice and games. For consistency with Study 1, we extracted the first 12 self-explanations generated by the students. The self-explanations were associated with two texts: ‘Ecological Pyramids’ (identical to Study 1), and a second text that varied among participants based on the text selection algorithm.

In the pretest, participants were prompted to self-explain nine target sentences while reading a scientific text on either ‘Heart Disease’ or ‘Red Blood Cells’ (identical to Study 1). Participants were randomly assigned to self-explain one of the two texts.

Students’ reading comprehension skills were measured using a modified version of the Gates-MacGinitie Reading Test (4th ed.) level 10/12 form S, which asks students to read short passages and then answer questions about the content of the passages. There were 48 multiple-choice questions in the test and the time limit was 25 minutes. The questions were designed to measure students’ ability to comprehend shallow and deep-level information. The raw scores of the reading test were used as the measure of students’ reading skills (identical to Study 1).

5.2.3 Data Processing and Statistical Analyses

The procedure for self-explanation extraction and aggregation was identical to Study 1. The aggregated files were analyzed with TAALES and Coh-Metrix, and the scores of the five linguistic features (i.e., word count, written trigram frequency, average distance of closest orthographic neighbors, causal verbs and particles incidence, and temporal connectives incidence) identified in Study 1 were calculated across the 13 files. Next, the correlations between the five linguistic features and the reading test scores were calculated in each dataset. Finally, a multiple linear regression analysis was performed to predict students' reading test scores with the five linguistic features across the 13 datasets, iteratively.

5.2.4 Results

As is shown in Table 11.3, only two out of the five linguistic features were significantly correlated with the reading test scores across most of the datasets. These two features (i.e., word count and average distance of closest orthographic neighbors) also had the highest average correlation with reading test scores across the 13 datasets. The descriptive statistics of the five linguistic features are shown in Table 11.3.

A multiple linear regression analysis was performed to examine the extent to which linguistic features of self-explanations predict students' reading skills. In the linear regression model, the five linguistic features were used to predict students' reading test scores. The model was applied to the 13 datasets iteratively. Two linguistic features (i.e., word count and word neighbor distance) were found to be significant predictors of reading test scores consistently across datasets (see Table 11.4).

The variance explained by selected linguistic features was compared across datasets to examine whether the number of self-explanations (i.e., ranging from 1 to 12 self-explanations) affected the strength of the predictive models. Overall, linguistic features accounted for more variance in the standardized reading test scores when more self-explanations were considered. However, the explained variance seemed to stabilize after 9 self-explanations.

The variance explained by the linguistic features of pretest self-explanations and training self-explanations was also compared. As shown in Table 11.4, the linguistic features of both sets of self-explanations accounted for 20–22% variance in the reading test scores.

5.2.5 Summary

Study 2 explored the extent to which the five linguistic features identified in Study 1 could predict reading test scores for a new set of students (i.e., replication and generalization). In addition, we examined whether the strength of predictive models was affected by (1) the number of

Table 11.3 Descriptive Statistics of the Five Linguistic Features Selected to Predict the Reading Test Scores

NLP Tool	**Linguistic Features**	**Average *r***	**Datasets**	**Mean (SD)**
TAALES	Word count (AW)	0.25	12	579.20 (275.95)
TAALES	BNC Written trigram Frequency (AW)	0.04	0	0.01 (.00)
TAALES	Average distance of closest orthographic neighbors (CW)	0.23	9	2.02 (.09)
Coh-Metrix	Causal verbs and casual particles incidence	0.07	0	41.80 (12.86)
Coh-Metrix	Temporal connectives incidence	0.12	1	20.66 (8.51)

Note: CW = Content Words, AW = All Words

Table 11.4 Multiple Linear Regression Analysis Predicting Reading Test Scores With Linguistic Features

	Standardized Coefficients							
Dataset	Word Count	Trigram Frequency	Word Neighbor Distance	Causal Verbs and Particles	Temporal Connectives	F	p	R^2
1 SE	.16	.08	−.06	.17	.07	1.55	.18	.08
2 SEs	$.27^{**}$	−.11	.00	.20	.15	2.97	.02	.15
3 SEs	$.31^{**}$	−.05	.00	.18	$.20^{*}$	3.14	.01	.15
4 SEs	$.26^{*}$	−.03	$.20^{*}$	.13	.10	2.89	.02	.14
5 SEs	$.25^{*}$	−.02	$.28^{**}$	−.04	.08	3.10	.01	.15
6 SEs	$.25^{*}$	−.02	$.27^{**}$	−.06	.07	2.91	.02	.14
7 SEs	$.26^{*}$	−.01	$.32^{**}$	−.01	.08	3.74	<.01	.18
8 SEs	$.26^{*}$	−.01	$.32^{**}$	.04	.09	3.88	<.01	.18
9 SEs	$.25^{*}$	−.04	$.36^{***}$	.04	.14	4.57	<.01	.21
10 SEs	$.25^{**}$	−.05	$.36^{***}$	.01	.15	4.85	<.01	.22
11 SEs	$.25^{*}$	.01	$.34^{***}$	.07	.18	4.92	<.01	.22
12 SEs	$.25^{*}$	−.02	$.31^{**}$	.06	$.21^{*}$	4.43	<.01	.20
PreSEs	$.24^{*}$	−.09	$.37^{***}$	−.11	.11	4.54	<.01	.21

Note: SE = self-explanation, PreSEs = pretest self-explanation, $^{*}\ p < .05$, $^{**}\ p < .01$, $^{***}\ p < .001$

self-explanations and/or (2) the self-explaining context (i.e., during training vs. before training). The results indicated that two linguistic features – word count and word neighbor distance – were significant predictors of reading scores in most datasets. Additionally, the variance in the reading scores explained by linguistic features increased with the number of self-explanations, although the increase stabilized after nine self-explanations. Finally, there seemed to be no effect of self-explaining context.

6. Discussion

In both studies, we explored the potential of NLP-based stealth assessments to evaluate students' reading skills in iSTART. Specifically, two automated text analysis tools (TAALES and Coh-Metrix) were used to analyze the linguistic properties of students' constructed responses at multiple levels (e.g., descriptive, lexical, syntactic, and semantic). Results revealed that the linguistic properties were indeed able to predict students' scores on a standardized reading test. Additionally, selected linguistic properties were able to predict students' reading test scores on a separate dataset.

The regression analyses in Studies 1 and 2 revealed that *word count* and *orthographic word neighbor distance* indices were significant predictors of reading test scores across most datasets. Thus, better readers tended to write longer self-explanations and use words that vary more in their written representation (i.e., spelling), perhaps because they had better understanding of the target texts and more sophisticated vocabulary.

These findings demonstrate the feasibility of implementing stealth literacy assessments in iSTART using NLP methods (see also Allen et al., 2015; McCarthy, Laura, et al., 2020). In particular, iSTART can collect language data (i.e., constructed responses) when students play generative practice games. The self-explanations can be processed and analyzed with NLP methods to infer students' literacy skills. These procedures (i.e., data collection and analysis) are conducted during students' practice or gameplay, and thus students can be assessed without interrupting gameplay. As such, these stealth assessments can facilitate the adaptation of the system. For example, the students with higher literacy skills can be guided to more difficult texts, games, or modules, while the students with lower literacy skills may be directed toward videos or various games implementing practice in lower-level reading strategies.

In addition to the feasibility of stealth assessments, we also began to explore *how* to implement stealth assessments. Specifically, we examined to what extent the predictive power of linguistic features on reading skills was affected by two factors: the number of self-explanations and the self-explaining context. Both studies suggested that the regression models successfully predicted greater variance in reading test scores when they were based on a larger set of self-explanations. However, the data also suggest diminishing returns of datasets larger than nine self-explanations. We thus suggest using nine self-explanations for a quick and reliable stealth literacy assessment. As such, students' literacy skills might be usefully evaluated by playing approximately two iSTART generative games.

The regression analyses in both studies also demonstrated that the linguistic features of self-explanations generated *during* iSTART training accounted for as much as or greater variance in reading scores than self-explanations generated before training. This finding suggests the literacy assessment during iSTART training is relatively reliable, and comparable to what might be yielded during the pretest (i.e., a non-stealth assessment).

There are two important caveats to note regarding these findings. First, the degree to which the predictive accuracy of the NLP indices stabilizes across responses will depend on multiple factors including the length of the responses, the degree to which the students are engaged in the task, and the reliability of the target measure. Second, only two of the five indices generalized in the context of Study 2, with a different population (i.e., high school students) and different texts. This was a strong test of generalization; yet, the results indicate that more studies are necessary to further test these findings and examine the extent to which they hold across various contexts and populations.

Importantly, although the two studies show promising evidence for stealth literacy assessments in iSTART, NLP methods might be further explored to improve validity. The NLP tools used in current analyses included TAALES and Coh-Metrix. There are numerous other NLP tools that calculate different linguistic features or similar features using alternative methods, such as CRAT (Crossley et al., 2016a) and TAACO (Crossley et al., 2016b). Additional linguistic features offer the potential to account for additional variance in reading skills because they may capture different dimensions of literacy.

Finally, we implemented a conventional analytical approach based on linear regression in the current study. As such, the linguistic features (i.e., predictors) were selected based on their correlations with the reading test scores. Notably, there are other machine learning and nonlinear methods for feature selection, including filter, wrapper, and embedded methods (Cai et al., 2018; Kou et al., 2020), with various advantages and disadvantages to each. Filter methods are efficient, fast, and have been used extensively in text categorization. Wrapper and embedded methods can achieve better accuracy than filter methods, but they are computationally complex and require more computational time. Paired with various machine learning approaches such as linear discriminant analysis (LDA), support vector machine (SVM), random forest, or neural networks (Balyan et al., 2019), more advanced approaches are likely to yield even more accurate predictions of literacy.

In conclusion, this study offers a significant step forward in demonstrating the feasibility of stealth measures of literacy leveraging NLP. At a practical level, the implementation of reliable stealth literacy assessments can be used to strengthen the adaptivity of tutoring systems such as iSTART. Students not only can be provided with individualized feedback or texts with varying difficulty within a game, but might also be directed to beneficial instructional modules, practice activities, or games based on the stealth assessment. At a theoretical level, the success of using language production, and the features of that language, to predict students' performance on literacy assessments points to the power of NLP and to the importance of considering multiple dimensions of language to understand cognition.

References

Allen, L. K., & McNamara, D. S. (2015). You are your words: Modeling students' vocabulary knowledge with natural language processing. In O. C. Santos, J. G. Boticario, C. Romero, M. Pechenizkiy, A. Merceron, P. Mitros, J. M. Luna, C. Mihaescu, P. Moreno, A. Hershkovitz, S. Ventura, & M. Desmarais (Eds.), *Proceedings of the 8th international conference on educational data mining (EDM 2015)* (pp. 258–265). International Educational Data Mining Society.

Allen, L. K., Snow, E. L., & McNamara, D. S. (2015). Are you reading my mind? Modeling students' reading comprehension skills with natural language processing techniques. In J. Baron, G. Lynch, N. Maziarz, P. Blikstein, A. Merceron, & G. Siemens (Eds.), *Proceedings of the 5th international learning analytics & knowledge conference* (pp. 246–254). ACM.

Balyan, R., Crossley, S. A., Brown, W., Karter, A. J., McNamara, D. S., Liu, J. Y., Lyles, C. R., & Schillinger, D. (2019). Using natural language processing and machine learning to classify health literacy from secure messages: The ECLIPPSE study. *PLoS One*, *14*(2), e0212488.

Balyan, R., McCarthy, K. S., & McNamara, D. S. (2020). Applying natural language processing and hierarchical machine learning approaches to text difficulty classification. *International Journal of Artificial Intelligence in Education (IJAIED)*, *30*, 337–370.

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python: Analyzing text with the natural language toolkit*. O'Reilly Media, Inc.

Brysbaert, M., & New, B. (2009). Moving beyond Kučera and Francis: A critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for American English. *Behavior Research Methods*, *41*(4), 977–990.

Burstein, J. (2003). The e-rater scoring engine: Automated essay scoring with natural language processing. In M. D. Shermis & J. C. Burstein (Eds.), *Automated essay scoring: A cross disciplinary approach* (pp. 113–121). Lawrence Erlbaum Associates.

Cai, J., Luo, J., Wang, S., & Yang, S. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing*, *300*, 70–79.

Crossley, S.A, Kyle, K., & McNamara, D. S. (2016a). Automatic assessment of constructed response data in a chemistry tutor. In *Proceedings of the 9th international educational data mining society conference* (pp. 336–340). EDM.

Crossley, S. A., Kyle, K., & McNamara, D. S. (2016b). The tool for the automatic analysis of text cohesion (TAACO): Automatic assessment of local, global, and text cohesion. *Behavior Research Methods*, *48*(4), 1227–1237.

Davies, M. (2009). The 385+ million word corpus of contemporary American English (1990–2008+): Design, architecture, and linguistic insights. *International Journal of Corpus Linguistics*, *14*(2), 159–190.

D'Mello, S. K., Dowell, N., & Graesser, A. (2011). Does it really matter whether students' contributions are spoken versus typed in an intelligent tutoring system with natural language? *Journal of Experimental Psychology: Applied*, *17*(1), 1.

Elliot, S. (2003). IntelliMetric: From here to validity. In M. D. Shermis & J. C. Burstein (Eds.), *Automated essay scoring: A cross disciplinary approach*. Lawrence Erlbaum Associates.

Graesser, A. C., Jeon, M., & Dufty, D. (2008). Agent technologies designed to facilitate interactive knowledge construction. *Discourse Processes*, *45*(4–5), 298–322.

Graesser, A. C., & McNamara, D. S. (2012). Automated analysis of essays and open-ended verbal responses. In H. Cooper, P. Camic, R. Gonzalez, D. Long, & A. Panter (Eds.), *APA handbook of research methods in psychology: Foundations, planning, measures, and psychometrics*. American Psychological Association.

Ingebrand, S. W., & Connor, C. M. (2016). Assessment-to-instruction (A2i): An online platform for supporting individualized early literacy instruction. In S. A. Crossley & D. S. McNamara (Eds.), *Adaptive educational technologies for literacy instruction* (pp. 33–48). Routledge.

Jackson, G. T., & McNamara, D. S. (2011). Motivational impacts of a game-based intelligent tutoring system. In R. C. Murray & P. M. McCarthy (Eds.), *Proceedings of the 24th international Florida artificial intelligence research society (FLAIRS) conference* (pp. 519–524). AAAI Press.

Jackson, G. T., & McNamara, D. S. (2013). Motivation and performance in a game-based intelligent tutoring system. *Journal of Educational Psychology*, *105*(4), 1036–1049.

Johnson, A. M., Guerrero, T. A., Tighe, E. L., & McNamara, D. S. (2017). iSTART-ALL: Confronting adult low literacy with intelligent tutoring for reading comprehension. In E. Andre, R. Baker, X. Hu, M. M. T. Rodrigo, & B. du Boulay (Eds.), *International conference on artificial intelligence in education* (pp. 125–136). Springer.

Johnson, A. M., Perret, C. A., Watanabe, M., Kopp, K., McCarthy, K. S., & McNamara, D. S. (2018). Adaptive literacy instruction in iSTART and W-Pal: Implementing the outer loop. In S. Craig (Ed.), *Tutoring and intelligent tutoring systems* (pp. 221–250). Nova Science Publishers.

Kellogg, R. T., & Raulerson, B. A. (2007). Improving the writing skills of college students. *Psychonomic Bulletin & Review*, *14*(2), 237–242.

Kintsch, W. (1988). The role of knowledge in discourse comprehension: A construction-integration model. *Psychological Review*, *95*(2), 163–182.

Kou, G., Yang, P., Peng, Y., Xiao, F., Chen, Y., & Alsaadi, F. E. (2020). Evaluation of feature selection methods for text classification with small datasets using multiple criteria decision-making methods. *Applied Soft Computing*, *86*, article 105836. https://doi.org/10.1016/j.asoc.2019.105836

Kyle, K., Crossley, S., & Berger, C. (2018). The tool for the automatic analysis of lexical sophistication (TAALES): Version 2.0. *Behavior Research Methods*, *50*(3), 1030–1046.

Landauer, T. K., Laham, D., & Foltz, P. W. (2003). Automated scoring and annotation of essays with the intelligent essay assessor. In M. D. Shermis & J. Burstein (Eds.), *Automated essay scoring: A cross-disciplinary perspective* (pp. 87–112). Lawrence Erlbaum Associates.

Landauer, T. K., McNamara, D. S., Dennis, S., & Kintsch, W. (Eds.). (2007). *Handbook of latent semantic analysis*. Lawrence Erlbaum Associates.

Leacock, C., & Chodorow, M. (2003). C-rater: Automated scoring of short-answer questions. *Computers and the Humanities*, *37*(4), 389–405.

Litman, D. J., Rosé, C. P., Forbes-Riley, K., VanLehn, K., Bhembe, D., & Silliman, S. (2006). Spoken versus typed human and computer dialogue tutoring. *International Journal of Artificial Intelligence in Education*, *16*(2), 145–170.

MacGinitie, W. H., & MacGinitie, R. K. (1989). *Gates MacGinitie reading tests*. Riverside.

Magliano, J. P., Millis, K. K., Levinstein, I., & Boonthum, C. (2011). Assessing comprehension during reading with the reading strategy assessment tool (RSAT). *Metacognition and Learning*, *6*(2), 131–154.

McCarthy, K. S., Laura, K. A., & Scott, R. H. (2020). Predicting reading comprehension from constructed responses: Explanatory retrievals as stealth assessment. In *Proceedings of the international conference on artificial intelligence in education* (pp. 197–202). Springer.

McCarthy, K. S., Likens, A. D., Johnson, A. M., Guerrero, T. A., & McNamara, D. S. (2018). Metacognitive overload!: Positive and negative effects of metacognitive prompts in an intelligent tutoring system. *International Journal of Artificial Intelligence in Education*, *28*(3), 420–438.

McCarthy, K. S., Watanabe, M., Dai, J., & McNamara, D. S. (2020). Personalized learning in iSTART: Past modifications and future design. *Journal of Research on Technology in Education*, *52*(3), 301–321.

McNamara, D. S. (2004). SERT: Self-explanation reading training. *Discourse Processes*, *38*, 1–30.

McNamara, D. S. (2011). Computational methods to extract meaning from text and advance theories of human cognition. *Topics in Cognitive Science*, *2*, 1–15.

McNamara, D. S. (2017). Self-explanation and reading strategy training (SERT) improves low-knowledge students' science course performance. *Discourse Processes*, *54*(7), 479–492.

McNamara, D. S. (2021). *Chasing theory with technology: A quest to understand understanding*. Department of Psychology, Arizona State University, Manuscript submitted for Publication.

McNamara, D. S., Allen, L. K., McCarthy, S., & Balyan, R. (2018). NLP: Getting computers to understand discourse. In K. Millis, D. Long, J. Magliano, & K. Wiemer (Eds.), *Deep learning: Multi-disciplinary approaches*. Routledge.

McNamara, D. S., Boonthum, C., Levinstein, I. B., & Millis, K. (2007). Evaluating self-explanations in iSTART: Comparing word-based and LSA algorithms. In T. Landauer, D. S. McNamara, S. Dennis, & W. Kintsch (Eds.), *Handbook of latent semantic analysis* (pp. 227–241). Lawrence Erlbaum Associates.

McNamara, D. S., Graesser, A. C., McCarthy, P., & Cai, Z. (2014). *Automated evaluation of text and discourse with Coh-Metrix*. Cambridge University Press.

McNamara, D. S., & Magliano, J. P. (2009). Self-explanation and metacognition: The dynamics of reading. In J. D. Hacker, J. Dunlosky, & A. C. Graesser (Eds.), *Handbook of metacognition in education* (pp. 60–81). Lawrence Erlbaum Associates.

NICHD (2000). *National institute of child health and human development, national reading panel report: Teaching children to read: An evidence-based assessment of the scientific research literature on reading and its implications for reading instruction*. U.S. Department of Health and Human Services, Public Health Service, National Institutes of Health, National Institute of Child Health and Human Development.

Organization for Economic Cooperation and Development (2013). *OECD skills outlook 2013: First results from the survey of adult skills*. OECD Publishing.

Partnership for the 21st Century (2019). *Framework for 21st century learning*. Retrieved April 18, 2021, from http://static.battelleforkids.org/documents/p21/P21_Framework_Brief.pdf

Powell, P. R. (2009). Retention and writing instruction: Implications for access and pedagogy. *College Composition and Communication*, *60*(4), 664–682.

Ruseti, S., Dascalu, M., Johnson, A., McNamara, D. S., Balyan, R., Kopp, K., Crossley, S. A., & Trausan-Matu, S. (2018). Predicting question quality using recurrent neural networks. In C. P. Rosé, R. Martínez-Maldonado, U. Hoppe, R. Luckin, M. Mavrikis, K. Porayska-Pomsta, B. McLaren, & B. D. Boulay (Eds.), *Proceedings of the 19th international conference on artificial intelligence in education (AIED 2018)* (Part I, pp. 491–502). Springer.

Shermis, M. D., Burstein, J., Higgins, D., & Zechner, K. (2010). Automated essay scoring: Writing assessment and instruction. *International Encyclopedia of Education*, *4*, 20–26.

Shute, V. J., Leighton, J. P., Jang, E. E., & Chu, M. W. (2016). Advances in the science of assessment. *Educational Assessment*, *21*(1), 34–59.

Shute, V. J., & Moore, G. R. (2017). Consistency and validity in game-based stealth assessment. In H. Jiao & R. W. Lissitz (Eds.), *Technology enhanced innovative assessment: Development, modeling, and scoring from an interdisciplinary perspective* (pp. 31–51). Information Age Publishing.

Shute, V. J., & Psotka, J. (1996). Intelligent tutoring systems: Past, present and future. In D. Jonassen (Ed.), *Handbook of research on educational communications and technology*. Scholastic Publications.

Shute, V. J., & Rahimi, S. (2021). Stealth assessment of creativity in a physics video game. *Computers in Human Behavior*, *116*. https://doi.org/10.1016/j.chb.2020.106647

Shute, V. J., & Ventura, M. (2013). *Measuring and supporting learning in games: Stealth assessment*. MIT Press.

Shute, V. J., Ventura, M., & Ke, F. (2015). The power of play: The effects of portal 2 and lumosity on cognitive and noncognitive skills. *Computers & Education*, *80*, 58–67.

Shute, V. J., Ventura, M., & Kim, Y. J. (2013). Assessment and learning of qualitative physics in Newton's playground. *The Journal of Educational Research*, *106*(6), 423–430.

Shute, V. J., Wang, L., Greiff, S., Zhao, W., & Moore, G. (2016). Measuring problem solving skills via stealth assessment in an engaging video game. *Computers in Human Behavior*, *63*, 106–117.

van Dijk, T., & Kintsch, W. (1983). *Strategies of discourse comprehension*. Academic Press.

VanLehn, K. (2006). The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education*, *16*, 227–265.

Vellutino, F. R. (2003). Individual differences as sources of variability in reading comprehension in elementary school children. In A. P. Sweet & C. E. Snow (Eds.), *Rethinking reading comprehension* (pp. 51–81). Guilford Press.

Ventura, M., & Shute, V. (2013). The validity of a game-based assessment of persistence. *Computers in Human Behavior*, *29*(6), 2568–2572.

Wang, L., Shute, V., & Moore, G. R. (2015). Lessons learned and best practices of stealth assessment. *International Journal of Gaming and Computer-Mediated Simulations*, *7*(4), 66–87.

Woolf, B. P. (2010). *Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning*. Morgan Kaufmann.

12

Measuring Scientific Understanding Across International Samples

The Promise of Machine Translation and NLP-Based Machine Learning Technologies

Minsu Ha and Ross H. Nehm

1. Introduction

Science educators have been engaged in international comparison studies of student learning for decades. On a large scale, the Trends in Mathematics and Science Study (TIMSS) and the Programme for International Student Assessment (PISA) have documented substantial (and in many cases quite surprising) student learning and competency differences across OECD (Organization for Economic Cooperation and Development) and non-industrialized nations (e.g., OECD, 2019). These results have mobilized substantial public and private funds to investigate performance disparities, particularly in STEM fields (e.g., NRC, 2006). These findings have identified successful (e.g., Finland) and less successful educational systems and have focused attention on the curricula, pedagogies, teacher preparation approaches, and bureaucratic structures that appear to foster or constrain meaningful learning outcomes (Simola, 2005). Although TIMSS and PISA have been remarkably informative and effective at answering large-scale questions, because of time and financial constraints these assessment programs simply cannot cover the diversity or depth of science subjects or empirical questions of interest to researchers focusing on next-generation disciplinary learning (e.g., Next Generation Science Standards; NRC, 2012). Smaller-scale international comparison studies are likely to continue to generate novel insights into human learning of science that have implications for educators more broadly (e.g., Ha et al., 2019).

The rich potential of international science education comparison studies is often truncated by practical constraints, such as the reliance on closed-response item formats for educational measurement (e.g., multiple choice, true-false, Likert-scale). Yet, these assessment formats are poorly suited for generating robust inferences about next-generation science proficiencies aligned with 21st-century skills (e.g., Jones et al., 2013; Kim & Nehm, 2011; Mun et al., 2015; NRC, 2014). As the models of learning and constructs of interest in the field of science education have shifted, so too must the tasks and measures used to understand student learning of these proficiencies. In science education, for example, generating explanations, arguments, and models cannot be measured using multiple-choice item formats (NRC, 2014). The high costs of multilingual translators and assessment scorers remain insurmountable hurdles for many

DOI: 10.4324/9781003278658-16

This chapter has been made available under a CC-BY-NC-ND 4.0 license.

smaller-scale international comparison studies that are likely to offer valuable insights into science learning. Technological advances may offer opportunities for overcoming these challenges.

The proliferation of easy-to-use, open-source tools for natural language processing (NLP), machine learning (ML), and machine translation (MT) offer immense potential for advancing the scope of international measurement and assessment research by overcoming the aforementioned barriers. Currently, many of the advances in the application of NLP and ML to science learning (e.g., biology) remain restricted to certain countries and languages (e.g., USA/English: Ha & Nehm, 2016; Ha et al., 2011; Moharreri et al., 2014). Smaller-scale studies have demonstrated that open-source tools using NLP-based ML engines can reliably and accurately measure core constructs through the text-based analyses of complex assessment products (e.g., scientific explanation). But the question remains as to whether this work can be efficiently and effectively scaled in cost-effective ways in international comparison studies. This challenge – extending successful English-based NLP-ML scoring engines to multi-language corpora using machine translation – is the focus of this chapter. We begin with a summary of the value of international comparison studies and continue with an empirical demonstration of the promise of open-source NLP, machine learning, and machine translation pipelines using an educational measurement project in the life sciences. We end with a discussion of the implications of this work and argue that NLP, ML, and MT will be central to achieving a diversified educational research portfolio that balances high-risk but potentially transformative small-scale research and lower risk but high-cost large-scale international research efforts.

2. International Comparison Studies in Science Education

Although the promise of international comparison studies was proposed more than a half century ago (1950s; see Husen, 1996), governments and policy makers did not devote substantial resources to such work until the 1980s (OECD, 1992). The relationship between economic development and educational quality has been of interest to governments and policy makers for some time (NRC, 1993), and mathematics and science have been of particular interest in relation to these foci (e.g., TIMSS and PISA). For instance, both subjects have been associated with technological and economic development (NRC, 1993, p. 10) as improvements in mathematics and science curricula may enhance academic achievement, which in turn may produce a better-equipped STEM workforce and ultimately facilitate more innovative and productive scientists and engineers. International comparative studies in STEM articulate with fundamental concerns tied to economic growth, and TIMSS and PISA studies have consequently spurred a substantial number of journal articles, government reports, and academic books on the topics of science and mathematics education.

International comparison studies – whether large or small scale – inevitably confront the questions of which constructs to assess, and which assessment tasks are most valuable (and practically achievable) for measuring them. The National Research Council (1993, p. 19) proposed foundational assessment topics and formats for international comparison studies – in particular, performance-based measurements of cognitive learning outcomes. The NRC has emphasized the importance of performance-based assessments using open-ended or free-response items in several reports (NRC, 1990, 2001, 2012, 2014), as have many journal articles in science education (see Anderson et al., 2018; Chen et al., 2019; Liu et al., 2016). Enactment of these recommendations has been limited because of the significant expense of administering, translating, and scoring performance-based tests (NRC, 1990, p. 27). Nevertheless, the benefits of performance-based assessment were viewed as well worth the cost given the richness of the inferences that could be drawn from these measures.

Major educational reforms in the United States (NRC, 2012, 2013, 2014) now emphasize so-called three-dimensional science learning and assessment (i.e., the intersection of core ideas, practices, and crosscutting concepts). Three-dimensional assessment typically generates complex performance-based products (e.g., written arguments, explanations, scientific models) that in turn require complex rubrics and scoring approaches. These issues pose challenges for both large-scale international comparison studies and smaller-scale studies exploring disciplinary topics, cognitive processes, and cultural influences on learning (e.g., Ha et al., 2019). Technologies such as natural language processing and machine learning offer immense potential for the scoring of three-dimensional science proficiencies, particularly for smaller-scale investigations.

3. Open-Source Natural Language Processing and Machine Learning Tools

One long-standing goal of automated assessment methods is to develop scoring models that are capable of high-accuracy predictions of the presence or absence of particular concepts, student reasoning patterns, or epistemic stances using text-based corpora (it is important to note that more advanced applications are possible, but recent reviews suggest that the field has yet to transcend these basic applications; see Zhai et al., 2020). The proliferation of open-source tools for natural language processing (NLP) and machine learning (ML) remains underutilized for advancing the scope of large- and small-scale international measurement and assessment research in science education. In the past decade, the number of open-source NLP and ML programs has proliferated dramatically. Prohibitively expensive technologies can now be utilized by many.

Our study relied on free and relatively basic open-source NLP-ML engine known as LightSIDE (Mayfield & Rosé, 2013). The Summarisation Integrated Development Environment (or 'SIDE') was an early open-source tool for both NLP and ML applications developed by the TELEDIA lab at Carnegie Mellon University (Mayfield & Rosé, 2012; see www.cs.cmu.edu/~emayfiel/LightSIDE.pdf). SIDE was updated with a user-friendly interface (i.e., LightSIDE) that has been incorporated into open-source automated scoring programs (e.g., EvoGrader; see Section 5.2 for details). Many more advanced open-source options (e.g., Natural Language Toolkit and R packages) now exist, and so the present study illustrates the significant potential of very basic NLP open-source tools for text-rich international comparison studies.

In brief, LightSIDE has a workflow that links NLP and supervised ML methods: Text Document → Extract Features → Develop Feature Table → Build Model → Train Model. In our work, LightSIDE was used to perform text corpus feature extraction using the most basic NLP approach possible, which involves converting each text response in the corpus into a 'bag of words' (BOW; Harris, 1954). BOW simplifies text representation and reduces the dimensionality of the feature space. In contrast to many other NLP approaches (e.g., latent semantic analysis), an assumption of BOW is that words are largely independent of their text position.

Our feature extraction involved the following approaches. First, words with a corpus frequency > 1 were used to generate a 'corpus dictionary'. High-frequency neutral words (e.g., 'the', 'of', 'to'), punctuations (e.g., '?'), and rare words were removed from the corpus in order to minimize noise. The remaining words were added to the dictionary. Second, stemming was used to treat words with the same stem as a single word (e.g., 'mutat' is shared by mutation, mutated, mutagen). Third, bigram modeling (Cavnar & Trenkle, 1994) was used to create double-word terms (e.g., 'differential reproduction') in the dictionary. Notably, different concepts are typically characterized by different combinations of the aforementioned features. LightSIDE has many more advanced NLP approaches, but prior work suggested that these basic text processing techniques performed well, as discussed next.

In model building or training, mathematical operations (e.g., regression) are used to model relationships between the extracted features and the human-tagged text (i.e., presence/absence of a concept in the corpus). Many different types of training algorithms (e.g., naive Bayes classifier, support vector machine classifier) are available; based on prior efficacy studies (Ha, 2013), the algorithms in our work were trained by sequential minimal optimization (Platt, 1998). Newer methods (e.g., deep learning, ensemble methods) offer considerable promise as well. When the features from the training corpus (i.e., the corpus used to train algorithms) align with the features of the testing corpus (i.e., a new corpus not yet scored by the algorithms), robust scoring accuracy tends to occur; differences can produce scoring errors. For example, if the text 'mutation' is a common feature in the training corpus aligned with the latent construct 'biological variation' (and, correspondingly, the feature table), but 'genetic error' (a feature with equivalent meaning for mutation) is a common feature of the testing corpus, then the model may introduce measurement error. Indeed, Ha et al. (2011) found that scoring model generalization could be limited by linguistic expressions characteristic of instructors from different universities. Note that these discrepancies were found for comparisons in the same language (i.e., English). However, a recent meta-analysis of machine-human agreement (MHA) has shown that many variables impact MHAs (i.e., algorithm, subject domain, assessment format, construct, school level, and machine supervision type; see Zhai et al., 2021) making definitive conclusions about what is causing such discrepancies quite challenging. In summary, many open-source NLP and ML tools are available for education researchers, and our work utilized a very basic BOW approach to feature extraction and model building using LightSIDE. We anticipate that more advanced NLP approaches and newer ML methods may generate better results.

4. Assessment Instrument Used in Our International Comparison Studies

We used the ACORNS instrument (Assessing COntextual Reasoning about Natural Selection; Nehm et al., 2012; Opfer et al., 2012) to generate a text corpus. The ACORNS is a constructed response assessment for measuring the disciplinary core idea of evolution in the context of the scientific practice of explanation. The ACORNS has been used in many different countries (e.g., Germany, Korea, China, Indonesia), which makes it well suited for international comparison studies. For the present study, four performance tasks from the ACORNS instrument were used; each assessed the disciplinary core idea of evolution through the practice of scientific explanation. The four tasks provide different anchoring phenomena (i.e., plant vs. animal; gain vs. loss of traits) that are used to assess cognitive coherence or the ability to generate explanations across the tree of life; this is a performance expectation for mastery of natural selection (Nehm, 2018, 2019). Extensive validity and reliability evidence (cf. AERA et al., 2014) exists for the ACORNS (e.g., Beggrow et al., 2014; Ha et al., 2019; Nehm et al., 2012; Opfer et al., 2012).

Students' explanations may be hand scored using the ACORNS rubric (Nehm et al., 2010). Specifically, each explanation (i.e., text response) can be scored for the presence or absence of six normative scientific concepts (or scientifically accurate ideas) relating to the construct of natural selection (e.g., variation, heritability, competition, limited resources, differential survival, and reproduction) and three nonnormative naive ideas or misconceptions (e.g., needs/goals, use/disuse, adapt/acclimation). These ideas have been documented in many studies of student reasoning about evolutionary change and natural selection (e.g., Bishop & Anderson, 1990). Composite measures (total normative concepts, total misconceptions) were also calculated. Finally, four reasoning models (i.e., no model, mixed model [normative + misconceptions], pure naive model [all misconceptions], pure scientific model [all normative ideas]) were generated as holistic measures. These measures have been shown to align with levels of understanding aligned with key cognitive principles (Opfer et al., 2012). We used the ACORNS instrument text corpora in a series of studies of international students. Machine translation was a central part of this work and is discussed next.

5. Machine Translation: An Untapped Resource in Small-Scale International Comparison Studies

The past decade has seen transformative changes in both the scope and quality of machine translation. The most widely used machine translation tool is Google Translate (> 500 million users as of 2020). Google Translate (GT) is available for >100 languages at various levels and, according to Google, translates more than 100 billion words daily. Although there are many machine translation services currently available, one of the reasons for selecting GT as a machine translation tool was that it is free (up to a certain level), easy to use, and was found to work well according to our human translators. Prior to selecting GT as a tool, we empirically compared the efficacy of GT with 'Bing Translator' (BT). We translated a hundred Chinese and Indonesian ACORNS responses using both GT and BT. Without exception, all native speakers agreed that the efficacy of GT's translation was better than the efficacy of BT's in terms of word choice and linguistic structures. Given that little is known about the quality of GT in specific science domains (e.g., evolutionary explanations), this study also employed two expert human translators as discussed in this section. A goal of this work was to examine the extent to which free tools like GT could be leveraged to advance small-scale international comparison studies in education (in this case, biology education).

5.1 Study 1: Machine Translation, Human Translation, Machine Scoring, and Human Scoring of Assessment Products

Study 1 includes three parts (A, B, and C). Study 1A focuses on the situation in which researchers engaged in small-scale international comparison research that require robust translation of assessment products but have trained experts at their disposal to score the translated text responses. In this situation, translation effects would be potential contributors to measurement error. The overarching goal of Study 1 was to generate ACORNS scores derived from English text that was produced by free machine translation (MT) tools (Google translator; GT) and expert human translation (HT). Two expert human translators were used as the benchmark for score comparisons, and all assessment products were graded by an expert rater. Study 1A also sets up Study 1B, which compares NLP-based machine learning (ML) scoring of the different translations to expert human scoring of the different translations (see Figure 12.1).

Chinese biology undergraduate students' written explanations ($n = 640$) were collected in response to the ACORNS items described earlier. Validity evidence in support of inferences drawn from ACORNS scores in the Chinese context is provided in Ha et al. (2019). Study 1A employed two expert human translators (HT1 and HT2) who independently translated the Chinese responses into English (a different human rater produced scores). Study 1A, therefore, produced three different scores (i.e., expert human scoring of the text from human translator 1 [hereafter HS-HT1], expert human scoring of the text from HT2, and expert human scoring of the text from Google Translate).

In contrast to Study 1A, Study 1B corresponds to the situation in which a research group has the resources to robustly translate written text but does not have the expertise or resources to score the translated responses. Such situations would benefit from the application of NLP-based ML scoring. Study 1B therefore examined the efficacy of computer scoring of the three translated corpora (i.e., from HT1, HT2, and GT) discussed in Study 1A. Study 1C was meant to align with the situation in which a research group does not have the resources for either translation or grading; thus, the researchers would use GT and computer scoring (CS) in their work. Study 1C therefore compared the scoring efficacy of CS-GT to both human expert scoring of HT1 and human expert scoring of HT2. Collectively, Study 1 sought to examine a variety of situations that could confront researchers interested in international comparison studies and determine the potential applications of open-source technological tools to address resource limitations in each case. We statistically compared the correspondence (e.g., agreement percentage, Cohen's kappa) between each permutation.

5.1.1 Study 1 Sample and Methods

We collected 640 written ACORNS explanations from biology undergraduate students from two Chinese universities. In this sample, the percentage of male students was 19.4%, which is common in teacher education in China. We employed two human translators and Google translation in Study 1. The first translator was a PhD student in science education in a Teacher Education University in China. He conducted studies of Chinese students' evolutionary ideas with U.S. scholars. The second translator was a bilingual biology teacher in the United States proficient in both Chinese and English. Google translation was also used to translate the text responses. Although Google Translate provides a customized function for users to correct its word choices, we did not use any correction functions. Two expert graders (e.g., PhD in biology education and an evolutionary biologist) independently scored the translated ACORNS responses. Inter-rater reliability was measured by Cohen's kappa and was found to be >> 0.8 for all six normative scientific ideas and three nonnormative naive ideas. In cases of scoring disagreement, discussions produced consensus scores which were used for final statistical tests.

We used four measures to quantify scoring correspondences: Cohen's kappa and agreement percentage (Bejar, 1991) and precision and recall. Cohen's kappa is widely used as a measure of inter-rater agreement not only in science education. Coefficients of Cohen's kappa range from 0.0 to 1.0. Although several studies suggested different benchmarks for interpreting kappas, we used Landis and Koch's (1977) cutoff values: 0.41–0.60 ('moderate'), 0.61–0.80 ('substantial'), and 0.81–1.00 ('almost perfect'). The second measure used to quantify correspondence between different types of ACORNS scoring was the percentage of scoring agreement. Both kappa and agreement percentage were used for binary scores of each concept and the four-category score of reasoning model types (see earlier in this section).

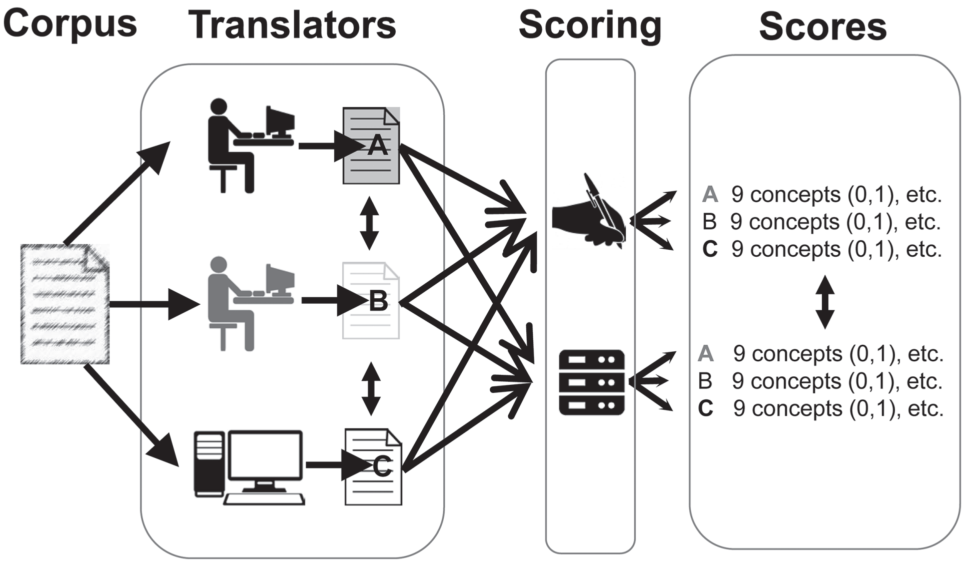


Figure 12.1 Simplified cartoon of the study design

Precision and recall were also measured; both are widely used in information retrieval studies (Su, 1994). Precision indicates the percentage of correct predictions among total positive predictions, whereas recall indicates the percentage of correct predictions among total actual positive cases. For the total key concept and naive idea scores, Pearson correlations were used to measure the correspondence between scoring approaches. Pearson correlation coefficients have been widely used in everywhere in applied statistics, and prior work considers coefficients > 0.9 Pearson to be 'identical' (Sato et al., 2005; Zhu et al., 2002).

5.2 Study 1 Findings

5.2.1 Expert Scoring of Three Different Translations of Students' Biological Explanations

Study 1A examined expert scoring of Chinese students' ACORNS responses translated into English using Google Translate (GT) and human translation (H1T and H2T; see Section 5.1). Five measures of score agreements were calculated: kappa, percentage agreement, precision, recall, and F1 (see Table 12.1). To examine the overall efficacy of human scoring of GT across nine concepts, we examined the extent to which the comparisons met acceptable agreement

Table 12.1 Kappa, Agreement, Precision, Recall, F1 Score Between Expert Scores of Two Different Translation Methods

Concept	Translator	Agreement	Kappa	Precision	Recall	F1
Variation	H1T–H2T	0.988	0.974	0.980	0.988	0.984
	H1T–GT	0.977	0.951	0.965	0.976	0.971
	H2T–GT	0.973	0.945	0.965	0.969	0.967
Heredity	H1T–H2T	0.972	0.760	0.689	0.886	0.775
	H1T–GT	0.972	0.676	0.870	0.571	0.690
	H2T–GT	0.966	0.660	1.000	0.511	0.676
Competition	H1T–H2T	0.998	0.959	0.923	1.000	0.960
	H1T–GT	0.997	0.921	0.857	1.000	0.923
	H2T–GT	0.998	0.962	0.929	1.000	0.963
Limited resource	H1T–H2T	0.977	0.924	0.919	0.958	0.938
	H1T–GT	0.961	0.871	0.898	0.891	0.895
	H2T–GT	0.975	0.918	0.958	0.911	0.934
Differential survival	H1T–H2T	0.914	0.761	0.718	0.946	0.816
	H1T–GT	0.895	0.678	0.735	0.752	0.743
	H2T–GT	**0.881**	0.672	0.856	0.665	0.748
Non-adaptive	H1T–H2T	1.000	1.000	1.000	1.000	1.000
	H1T–GT	1.000	1.000	1.000	1.000	1.000
	H2T–GT	1.000	1.000	1.000	1.000	1.000
Need/Goal	H1T–H2T	0.945	0.834	0.829	0.913	0.869
	H1T–GT	0.916	0.723	0.823	0.732	0.775
	H2T–GT	0.908	0.710	0.858	0.693	0.767
Use/Disuse	H1T–H2T	0.969	**0.221**	0.143	0.600	0.231
	H1T–GT	0.980	**0.308**	0.214	0.600	0.316
	H2T–GT	0.980	0.619	0.786	0.524	0.629
Adapt	H1T–H2T	0.969	0.816	0.794	0.877	0.833
	H1T–GT	0.950	0.726	0.671	0.860	0.754
	H2T–GT	0.959	0.786	0.753	0.873	0.809

Note: Agreement and Kappa values below benchmarks are shown in bold.

benchmarks (> 90% Agreement and 'Substantial' or 'Almost Perfect' kappa values; see Section 5.1). These comparisons produced remarkably similar findings. Specifically, 26 of 27 comparisons had Agreements > 90%, and 25 of 27 kappas were Substantial or Almost Perfect. In general, concept scoring did not differ by translation method.

In addition to concept-based comparisons, we also examined the kappa values and agreement percentages for the four-category reasoning model type variable (see above) between HS of HT and HS of GT. The agreement percentages and kappas of reasoning model types between HS of H1T and HS of H2T met the 'Almost Perfect' level (agreement = 87.5%, kappa = 0.815). The agreement percentage and kappa of reasoning model types between HS of H1T and HS of GT nearly reached the 'Almost Perfect' level (agreement = 86.3%, kappa = 0.794). Likewise, the agreement percentages and kappas of reasoning model types between HS of H2T and HS of GT nearly reached the 'Almost Perfect' level (agreement = 85.6%, kappa = 0.787). Thus, reasoning types were not substantially impacted by translation method.

Finally, the Pearson correlation coefficient of Key Concept Total Scores between HS of H1T and HS of H2T was 0.920 ($p \ll 0.01$) and the correlations between HS of H1T and HS of GT and between HS of H2T and HS of GT were 0.916 and 0.913, respectively ($p \ll 0.01$ for both). The Pearson correlation coefficient for the Naive Ideas Total Score between HS of H1T and HS of H2T was 0.937 ($p \ll 0.01$), whereas the coefficients between HS of H1T and HS of GT, and between HS of H2T and HS of GT, were 0.866 and 0.892, respectively ($p \ll 0.01$). Overall, robust kappas and correlation coefficients were found in nearly all cases, indicating the potential of expert scoring of Google translation in the assessment corpus we employed. One less promising finding was that GT was characterized by low recall values in some cases (e.g., the concepts of heredity and differential survival). Given these overall promising findings, we went on to explore the efficacy of ML-based computer scoring of these translations.

5.2.2 ML-Based Computer Scoring Results Across Three Different Translations

Study 1B examined ML-based computer scoring (i.e., EvoGrader) of the corpora produced by the three translation approaches used in Study 1A. Note that both HS and CS were performed on all three translations, thus providing a benchmark for comparison. Table 12.2 contains five measures of score correspondences for individual concepts: kappa, agreement, precision, recall, and F1. As in Study 1A, scoring performance benchmarks were used to compare approaches. First, 100% of comparisons met the 90% agreement benchmark (Table 12.2). Using kappa, only 2 out of 27 comparisons were found to be below the Substantial level (Non-adaptive and Use/disuse; Table 12.2).

We also examined the agreement percentages and kappa values for the four-category model type variable between HS and CS for the three different translations. (Recall that this variable is a holistic measure of the performance expectation for explaining evolution by natural selection.) The percentage agreements (and kappas) of reasoning model types between HS and CS of H1T, H2T, and GT were 88.3% (0.824), 94.2% (0.915), and 93.0% (0.895), respectively; all met the 'Almost Perfect' agreement level. The Pearson correlation coefficient for Key Concept Total Scores between HS and CS of H1T, H2T, and GT were, respectively, 0.936, 0.967, and 0.964 (all $p \ll 0.01$). The Pearson correlation coefficient for Naive Ideas Total Scores between HS and CS of H1T, H2T, and GT were 0.906, 0.970, and 0.960 respectively (all $p \ll 0.01$). Computer scoring of different translation methods was also found to be robust (see Table 12.2).

5.2.3 Machine Translation and Scoring of Assessment Products

Study 1C aligns with the situation in which a research group does not have the resources for either translation or grading; thus, the researchers would use GT and computer scoring (CS).

Table 12.2 Kappa, Agreement, Precision, Recall, F1 Scores Between HS and CS of Three Translated Responses by H1T, H2T, and GT

Concept	Translator	Agreement	Kappa	Precision	Recall	F1
Variation	H1	0.970	0.937	0.996	0.929	0.961
	H2	0.966	0.928	0.976	0.937	0.956
	GT	0.977	0.951	0.996	0.945	0.970
Heredity	H1	0.972	0.641	1.000	0.486	0.654
	H2	0.978	0.809	0.970	0.711	0.821
	GT	0.989	0.815	1.000	0.696	0.821
Competition	H1	0.997	0.915	0.917	0.917	0.917
	H2	0.997	0.915	1.000	0.846	0.917
	GT	0.998	0.962	1.000	0.929	0.963
Limited resources	H1	0.981	0.936	0.991	0.908	0.947
	H2	0.989	0.964	0.992	0.952	0.971
	GT	0.983	0.941	0.991	0.915	0.952
Differential survival	H1	0.909	0.679	0.908	0.612	0.731
	H2	0.952	0.873	0.943	0.871	0.905
	GT	0.961	0.875	0.957	0.848	0.900
Non-adaptive ideas	H1	0.998	0.666	1.000	0.500	0.667
	H2	1.000	1.000	1.000	1.000	1.000
	GT	0.995	**0.570**	0.400	1.000	0.571
Needs/Goals	H1	0.931	0.755	0.977	0.669	0.794
	H2	0.986	0.959	0.978	0.957	0.968
	GT	0.977	0.916	0.980	0.885	0.930
Use/Disuse	H1	0.991	**0.395**	0.400	0.400	0.400
	H2	0.988	0.794	0.842	0.762	0.800
	GT	0.991	0.745	0.900	0.643	0.750
Adapt	H1	0.986	0.914	0.914	0.930	0.922
	H2	0.975	0.873	0.797	1.000	0.887
	GT	0.988	0.940	0.922	0.973	0.947

Note: Agreement and Kappa values below benchmarks are shown in bold.

Study 1C therefore compared the scoring efficacy of GT-CS to both human expert scoring of HT1 and human expert scoring of HT2. Specifically, the accuracy of CS was compared between different translations (e.g., H1T vs. H2T, H1T vs. GT, H2T vs. GT). Table 12.3 contains the agreement values for nine concepts (CS of H1T and H2T, H1T and GT, and H2T and GT). As is apparent, score agreements were robust across translations, although less so than those in Tables 12.1 and 12.2. Specifically, 23 of 27 Agreements met the scoring benchmark of 90%, and 16 of 27 kappas met the Substantial benchmark. Notably, particular *concepts* (e.g., heredity, differential survival) accounted for lower scoring performance to a greater extent than the translation type (Table 12.3, bold values).

We also examined the agreement percentages and kappa values of the four-category model types between CS of the three different translations. The agreement percentages (kappas) of model types between CS-H1T and CS-H2T, between CS-H1T and CS-GT, and between CS-H2T and CS-GT were 76.4% (0.652), 77.3% (0.659), and 83.3% (0.754), respectively; all met the 'substantial' level. The Pearson correlation coefficient of the Key Concept Total Score between CS-H1T and CS-H2T, between CS-H1T and CS-GT, and between CS-H2T and CS-GT were 0.857, 0.869, and 0.877 ($p << 0.01$), respectively. The Pearson correlation coefficients for Naive Ideas between the same pairs were 0.835, 0.811, and 0.866, respectively.

Table 12.3 Kappa, Agreement, Precision, Recall, F1 Scores Between CSs of H1T and H2T, H1T and GT, and H2T and GT

Concept	Translator	Agreement	Kappa	Precision	Recall	F1
Variation	H1T–H2T	0.942	0.877	0.906	0.941	0.923
	H1T–GT	0.952	0.897	0.922	0.949	0.935
	H2T–GT	0.950	0.894	0.938	0.931	0.934
Heredity	H1T–H2T	0.959	**0.461**	0.364	0.706	0.480
	H1T–GT	0.977	**0.533**	0.563	0.529	0.545
	H2T–GT	0.964	**0.514**	0.813	0.394	0.531
Competition	H1T–H2T	0.992	0.779	0.818	0.750	0.783
	H1T–GT	0.992	0.796	0.769	0.833	0.800
	H2T–GT	0.997	0.915	0.846	1.000	0.917
Limited resource	H1T–H2T	0.953	0.840	0.832	0.908	0.868
	H1T–GT	0.931	0.757	0.798	0.798	0.798
	H2T–GT	0.963	0.872	0.936	0.857	0.895
Differential survival	H1T–H2T	**0.819**	**0.424**	0.408	0.736	0.525
	H1T–GT	**0.872**	**0.524**	0.521	0.701	0.598
	H2T–GT	**0.872**	0.621	0.821	0.611	0.701
Non-adaptive	H1T–H2T	0.998	0.666	0.500	1.000	0.667
	H1T–GT	0.994	**0.332**	0.200	1.000	0.333
	H2T–GT	0.995	**0.570**	0.400	1.000	0.571
Need/Goal	H1T–H2T	**0.884**	**0.604**	0.547	0.862	0.670
	H1T–GT	0.917	0.671	0.667	0.782	0.720
	H2T–GT	0.889	0.637	0.824	0.613	0.703
Use/Disuse	H1T–H2T	0.963	**−0.013**	0.000	0.000	n/a
	H1T–GT	0.977	**−0.011**	0.000	0.000	n/a
	H2T–GT	0.977	**0.472**	0.700	0.368	0.483
Adapt	H1T–H2T	0.939	0.682	0.620	0.845	0.715
	H1T–GT	0.933	0.645	0.597	0.793	0.681
	H2T–GT	0.953	0.781	0.818	0.797	0.808

Note: Agreement and Kappa values below benchmarks are shown in bold.

Table 12.4 illustrates the agreement percentages and kappa values for nine concepts between HS-H1T and HS-H2T, between HS-H1T and CS-GT, and between HS-H2T and CS-GT. Score agreements were generally robust across translation and scoring approaches, with the lowest value being 85%. Nevertheless, for some concepts (e.g., Heredity, Differential survival, Non-adaptive), GT&CS were associated with lower kappa values than HT&HS (Table 12.4, bold values). The concept use/disuse performed poorly across all translation and scoring comparisons.

We also examined the agreement percentages and kappas for the four-category model types between HS-HTs and CS-GT and compared them with those between HS-H1T and HS-H2T. The agreement percentages (kappas) of model types between HS-H1T and HS-H2T were 87.5% (0.815) and those between HS-H1T and CS-GT and between HS-H2T and CS-GT were 80.8% (0.712) and 81.7% (0.730), respectively. The Pearson coefficient of Key Concept Total Scores between HS-H1T and HS-H2T, HS-H1T and CS-GT, and HS-H2T and CS-GT were 0.920, 0.881, and 0.873, respectively (all $p << 0.01$); those for Naive Ideas were 0.937, 0.815, and 0.846 ($p << 0.01$). These findings, like those in the previous tables, demonstrate generally robust performance across translation and scoring approaches. Exceptions do occur, although they are related to concepts more so than particular translation and scoring combinations.

Table 12.4 Kappa, Agreement, Precision, Recall, F1 Scores Between HS-H1T and HS-H2T, Between HS-H1T and CS-GT, Between HS-H2T and CS-GT

Concept	Comparison	Agreement	Kappa	Precision	Recall	F1
Variation	H1T&HS–H2T&HS	0.988	0.974	0.980	0.988	0.984
	H1T&HS–GT&CS	0.953	0.901	0.959	0.921	0.940
	H2T&HS–GT&CS	0.950	0.895	0.959	0.914	0.936
Heredity	H1T&HS–H2T&HS	0.972	0.760	0.689	0.886	0.775
	H1T&HS–GT&CS	0.967	**0.574**	0.938	0.429	0.588
	H2T&HS–GT&CS	0.955	**0.506**	1.000	0.356	0.525
Competition	H1T&HS–H2T&HS	0.998	0.959	0.923	1.000	0.960
	H1T&HS–GT&CS	0.995	0.878	0.846	0.917	0.880
	H2T&HS–GT&CS	0.997	0.921	0.923	0.923	0.923
Limited resource	H1T&HS–H2T&HS	0.977	0.924	0.919	0.958	0.938
	H1T&HS–GT&CS	0.944	0.808	0.881	0.807	0.842
	H2T&HS–GT&CS	0.958	0.858	0.945	0.831	0.884
Differential survival	H1T&HS–H2T&HS	0.914	0.761	0.718	0.946	0.816
	H1T&HS–GT&CS	**0.859**	**0.547**	0.667	0.605	0.634
	H2T&HS–GT&CS	**0.845**	**0.560**	0.803	0.553	0.655
Non-adaptive	H1T&HS–H2T&HS	1.000	1.000	1.000	1.000	1.000
	H1T&HS–GT&CS	0.995	**0.570**	0.400	1.000	0.571
	H2T&HS–GT&CS	0.995	**0.570**	0.400	1.000	0.571
Need/Goal	H1T&HS–H2T&HS	0.945	0.834	0.829	0.913	0.869
	H1T&HS–GT&CS	**0.895**	0.645	0.794	0.638	0.707
	H2T&HS–GT&CS	**0.888**	0.635	0.833	0.607	0.702
Use/Disuse	H1T&HS–H2T&HS	0.969	**0.221**	0.143	0.600	0.231
	H1T&HS–GT&CS	0.980	**0.124**	0.100	0.200	0.133
	H2T&HS–GT&CS	0.970	**0.374**	0.600	0.286	0.387
Adapt	H1T&HS–H2T&HS	0.969	0.816	0.794	0.877	0.833
	H1T&HS–GT&CS	0.938	0.667	0.610	0.825	0.701
	H2T&HS–GT&CS	0.947	0.728	0.688	0.841	0.757

Note: Agreement and Kappa values below benchmarks are shown in bold.

5.3 Study 2: Extending Machine Translation and Machine Scoring to Indonesian and German Corpora

In Study 1, we found that NLP-based ML scoring models trained using American students' ACORNS assessment products were in many cases capable of effectively scoring human-translated and machine-translated Chinese ACORNS assessment products. The question arises as to whether these findings are unique to the Chinese corpus that we used in Study 1. In order to explore whether similar findings may occur in other languages, we gathered two more ACORNS corpora from Indonesian and German biology students (of comparable ages to the Chinese and American samples). Whereas in Study 1 we reported on four topics with two human translators, in Study 2 we tested two research questions given our interest in possible generalization of prior findings. In Study 2, we skipped the human scoring of Google translation and compared three different conditions: human scoring of human translation, computer scoring of human translation, and computer scoring of machine translation.

5.4 Study 2 Sample

Study 2 tests whether the findings from Study 1 (Chinese students) hold up across other samples (Indonesian and German students). We employed two new ACORNS response corpora generated by 371 Indonesian pre-service biology students and 219 German pre-service biology students. Specifically, the Indonesian participants generated 1,484 written explanations, and the German participants generated 876 written explanations. These text-based explanations were the data used in Study 2 (Nehm et al., 2013; Rachmatullah et al., 2018).

5.5 Study 2 Findings

Table 12.5 has a parallel format to the tables in Study 1. For the German ACORNS corpus, more than 90% agreement was achieved in all cases except 'limited resources', and for the Indonesian ACORNS corpus all concepts met the benchmark. Kappa measures of CS for 'Variation', 'Competition', and 'Non-adaptive idea' concepts were higher than 0.81 ('Almost Perfect' level) in both the Indonesian and German samples. We did find some differences in the kappa values between Indonesian and German scores. For example, the kappa for the 'Differential survival' concept in the German data was 0.932 (the 'Almost Perfect' level) whereas the kappa for the same concept in the Indonesian data was 0.495 ('Moderate'). On the other hand, the kappa for the 'use/disuse' concept in the Indonesian data was 0.862 ('almost perfect'), whereas the same concept in the German data was 0.576 ('moderate').

Finally, we examined the agreement percentage and kappa values for the four-category model type. The agreement percentage (and kappa) of reasoning model types between HS-HT and HS-GT were 85.6% (0.804) in the Indonesian data and 87.3% (0.798) in the German data.

Table 12.5 Kappa, Agreement, Precision, Recall, F1 Scores Between CS of Human Translation and HS of Human Translated Response

Concept	Language	Agreement	Kappa	Precision	Recall	F1
Variation	German	0.965	0.927	0.983	0.932	0.958
	Indonesian	0.962	0.882	0.964	0.854	0.909
Heredity	German	0.962	0.642	1.000	0.492	0.746
	Indonesian	0.993	0.781	1.000	0.645	0.823
Competition	German	0.998	0.975	1.000	0.953	0.977
	Indonesian	0.999	0.971	1.000	0.944	0.972
Limited resource	German	**0.870**	0.711	1.000	0.670	0.835
	Indonesian	0.989	0.970	0.997	0.959	0.978
Differential survival	German	0.966	0.932	0.995	0.939	0.967
	Indonesian	0.969	**0.495**	0.429	0.632	0.530
Non-adaptive	German	0.999	0.975	1.000	0.952	0.976
	Indonesian	1.000	1.000	1.000	1.000	1.000
Need/Goal	German	0.939	0.821	0.959	0.779	0.869
	Indonesian	0.912	0.759	0.873	0.766	0.819
Use/Disuse	German	0.981	**0.576**	0.522	0.667	0.594
	Indonesian	0.978	0.862	0.881	0.867	0.874
Adapt	German	0.966	0.719	0.894	0.627	0.760
	Indonesian	0.933	0.771	0.995	0.682	0.839

Note: Agreement and Kappa values below benchmarks are shown in bold.

The Pearson correlation coefficients for Key Concept Total Score between HS-HT and HS-GT were 0.924 (p << 0.01) in the Indonesian data and 0.960 (p << 0.01) in the German data, respectively. Naive Ideas total scores between HS-HT and HS-GT were 0.902 (p << 0.01) in the Indonesian data and 0.850 (p << 0.01) in the German data, respectively.

Our final analyses explored the efficacy of CS of Google translations of Indonesian and German students' responses by comparing the human scoring of the human translated responses (Table 12.6). First, we found that the agreements for 'Variation', 'Heritability', 'Competition', 'Non-Adaptive', and 'Use/Disuse' ideas were higher than 95% and all others were higher than 90% except for the 'Need/Goal' concept. The kappa measures for 'Variation' and 'Limited resources' were higher than 0.81 (almost perfect level) in both the Indonesian and German data. We also found discrepancies in kappa values between the Indonesian and German data. For example, the kappa for the 'Differential survival' concept in the German data was 0.841 ('Almost Perfect') whereas the kappa for the same concept in the Indonesian data was 0.275. On the other hand, the kappas for the 'Use/Disuse' concept in the Indonesian data was 0.688 ('Substantial'), whereas the same concept in the German data was 0.477 ('Moderate').

The last ACORNS categories we examined were the percentage agreement and kappa values for Model Types, Key Concepts, and Naive Ideas. The agreement percentage (and kappa) of the Model Types between HS-HT and CS-GT were 76.9% (0.681) in the Indonesian data and 82.3% (0.720) in the German data, respectively. The Pearson correlation coefficients for Key Concept Total Scores between HS-HT and CS-GT were, respectively, 0.881 (p << 0.01) in the Indonesian data and 0.942 (p << 0.01) in the German data; the Naive Ideas Total Scores between HS-HT and CS-GT were 0.792 (p << 0.01) in the Indonesian data and 0.740 (p << 0.01) in the German data. Overall, robust agreements were found in many cases, although some concepts displayed different patterns across languages.

Table 12.6 Kappa, Agreement, Precision, Recall, F1 Scores Between CS of Google Translation and HS of Human Translated Response

Concept	Language	Agreement	Kappa	Precision	Recall	F1
Variation	German	0.954	0.906	0.977	0.914	0.945
	Indonesian	0.953	0.855	0.921	0.851	0.886
Heredity	German	0.960	0.675	0.800	0.615	0.708
	Indonesian	0.986	**0.527**	0.857	0.387	0.622
Competition	German	0.979	0.759	0.857	0.698	0.777
	Indonesian	0.999	0.940	1.000	0.889	0.944
Limited resource	German	0.926	0.841	0.964	0.843	0.904
	Indonesian	0.970	0.920	0.978	0.905	0.941
Differential survival	German	0.908	0.815	0.950	0.870	0.910
	Indonesian	0.964	**0.275**	0.297	0.289	0.293
Non-adaptive	German	0.989	0.756	0.762	0.762	0.762
	Indonesian	0.998	0.799	0.667	1.000	0.833
Need/Goal	German	**0.898**	0.704	0.836	0.712	0.774
	Indonesian	**0.840**	**0.515**	0.803	0.489	0.646
Use/Disuse	German	0.968	**0.447**	0.353	0.667	0.510
	Indonesian	0.954	0.688	0.761	0.672	0.716
Adapt	German	0.934	**0.504**	0.576	0.507	0.542
	Indonesian	0.906	0.666	0.957	0.575	0.766

Note: Agreement and Kappa values below benchmarks are shown in bold.

6. Discussion

Although international comparison studies have been remarkably effective at addressing foundational educational research questions, many issues (e.g., financial constraints, expertise, and time) limit these assessment programs in their ability to fully cover the diversity or depth of science subjects and topics of interest to researchers. Moreover, the recent emphasis on next-generation disciplinary learning (e.g., Next Generation Science Standards; NRC, 2012) has shifted assessment research to more complex products, such as written explanations and arguments. Such performance assessments are much more complex to translate and score than closed-response and short answer assessments. Smaller-scale international comparison studies are likely to continue to generate novel insights into next-generation science learning more broadly (e.g., Ha et al., 2019). As is often the case, findings from small-scale studies raise important questions that lead to larger-scale research efforts. Fostering high-risk, small-scale educational research is therefore of importance to maintaining a diverse research portfolio in many domains, including science education.

A fundamental constraint in international education research is the prohibitive cost of expert translation and expert scoring of assessment products (e.g., text-based scientific explanations). This chapter explored the extent to which free and open-source tools for natural language processing (NLP), machine learning (ML), and machine translation (MT) could break these constraints and advance the scope of international measurement and assessment research focusing on written scientific explanations. Although Studies 1 and 2 were relatively limited in scope (three countries, ~2,000 explanations), together they demonstrated the immense potential of low-cost technological tools for advancing science education scholarship. This chapter may be viewed as a 'proof of concept' study for researchers seeking to investigate novel questions about next-generation learning internationally.

Given the financial constraints inherent to academic research, it is useful to explicitly estimate the extent to which free NLP, MT, and ML tools address research costs. Table 12.7 displays estimates of time and cost for scoring and translating 2,000 responses (500 students × 4 items) using the three methods explored in this chapter (human scoring of human translation, computer scoring of human translation, and computer scoring of Google translation). For our study, students were employed (given budgetary constraints), and so the costs are undoubtedly underestimated (Table 12.7). In terms of time, it took more than one month for humans to translate the ~2,000 essays into English, and it took more than two weeks for them to score the translated essays. In contrast, automated computer scoring of the Google translations took approximately one hour. In addition, limited use of Google translation was free. Scholars in many countries (particularly non-OECD) may lack the expertise and resources required to conduct small-scale studies (cf. Table 12.7). The approaches outlined in this chapter could provide one possible avenue for addressing these limitations and enriching scholarly insights into measurement, assessment, and cross-cultural learning in regions neglected from high-profile efforts (e.g., PISA, TIMSS).

As is the case with all technologies, machine translation and computer scoring have important limitations relevant to our findings. Perhaps the biggest limitation of our studies is that they did not directly examine human scoring of ACORNS responses in the original language (that is, prior to translation). This approach should be incorporated into future research designs because it can provide a more valid reference point for comparing the efficacy of different translation and scoring approaches. Training native speakers who are disciplinary experts to score responses and who demonstrate sufficient scoring reliability is expensive, however. Indeed, this factor was the limitation in this project.

The sizes of the language corpora and associated machine translation sophistication are unlikely to be uniform across languages, and differences in scoring success may be a product of

Table 12.7 Comparison Among the Estimates of Time and Cost for Scoring and Translating 2,000 Responses (500 Students × 4 Items) Using Three Methods (Human Scoring of Human Translation, Computer Scoring of Human Translation, and Computer Scoring of Google Translation)

	Scoring		Translation		Total	
Method	Time[1] (hour)	Cost[3] ($)	Time[2] (hour)	Cost[3] ($)	Time (hour)	Cost ($)
Human scoring of human translation	133.3	2000	200	3000	333.3	5000
Computer scoring of human translation	0.5	0	200	3000	200.5	3000
Computer scoring of Google translation	0.5	0	1	0	1.5	0

[1] 4 minutes for scoring one essay
[2] 6 minutes for translating one essay
[3] $15 per hour

these differences. As a result, our findings using Google translation of Chinese, German, and Indonesian responses may not generalize to other languages. Likewise, as computer translation research continues to improve, our findings may not be maintained; subtle linguistic details revealed through more sophisticated translations could produce differences in computer scoring success. On the one hand, they could improve translation quality, but it is not necessarily the case that this will improve scoring because the computer models were built using less nuanced language indicators. The stability of findings as computer translation improves is an issue in need of investigation.

Computer translation and computer scoring may instantiate human biases (Zhai et al., 2020). Uncommon dialects and cultural-linguistic practices may lack sufficient representation in a corpus, leading to differential effects on subgroups within samples. Human annotation of text may also be biased by who is performing the tagging. These and many other factors should be considered as potential sources of bias. Differential item functioning (DIF) across linguistic minorities within samples is one avenue for exploring potential biases.

Overall, our study suggests that free, automated machine scoring, and computer translation tools have great potential for researchers interested in studying more authentic scientific practices, such as explanations and arguments, across international borders. Continuing developments in computer technology will undoubtedly improve the quality of computer translations as well as the efficacy of automated scoring in the future. For these reasons, researchers need to engage more fully with these technologies, as they remain underutilized in science education and related fields and are likely to generate important insights into how culture, education, and cognition interact to generate understanding of the natural world. Pressing global scientific challenges ranging from climate change to pandemics transcend international boundaries and will require global educational efforts.

Acknowledgments

We thank the many individuals who helped us collect data in China, Indonesia, and Germany and the translators for their help converting the written responses to English. We also thank the National Science Foundation (DRL 0909999 and DUE 1322872 and 1338570) for funding parts of this work. This chapter represents the views of the authors and not the NSF. Finally, we thank the two reviewers and editors for help improving the clarity and quality of the chapter.

References

AERA, APA, & NCME. (2014). *Standards for educational and psychological testing: National council on measurement in education*. American Educational Research Association.

Anderson, C. W., de los Santos, E. X., Bodbyl, S., Covitt, B. A., Edwards, K. D., Hancock, J. B., Lin, Q., Morrison Thomas, C., Penuel, W. R., & Welch, M. M. (2018). Designing educational systems to support enactment of the next generation science standards. *Journal of Research in Science Teaching*, *55*(7), 1026–1052.

Beggrow, E. P., Ha, M., Nehm, R. H., Pearl, D., & Boone, W. J. (2014). Assessing scientific practices using machine-learning methods: How closely do they match clinical interview performance? *Journal of Science Education and Technology*, *23*(1), 160–182.

Bejar, I. I. (1991). A methodology for scoring open-ended architectural design problems. *Journal of Applied Psychology*, *76*(4), 522–532.

Bishop, B. A., & Anderson, C. W. (1990). Student conceptions of natural selection and its role in evolution. *Journal of Research in Science Teaching*, *27*(5), 415–427.

Cavnar, W. B., & Trenkle, J. M. (1994). *N-gram-based text categorization*. Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval (Vol. 161175).

Chen, J., Zhang, Y., Wei, Y., & Hu, J. (2019). Discrimination of the contextual features of top performers in scientific literacy using a machine learning approach. *Research in Science Education*, *51*(1), 129–158.

Ha, M. (2013). *Assessing scientific practices using machine learning methods: Development of automated computer scoring models for written evolutionary explanations* [Doctoral dissertation, The Ohio State University].

Ha, M., & Nehm, R. H. (2016). The impact of misspelled words on automated computer scoring: A case study of scientific explanations. *Journal of Science Education and Technology*, *25*(3), 358–374.

Ha, M., Nehm, R. H., Urban-Lurain, M., & Merrill, J. E. (2011). Applying computerized scoring models of written biological explanations across courses and colleges: Prospects and limitations. *CBE-Life Science Education*, *10*, 379–393.

Ha, M., Wei, X., Wang, J., Hou, D., & Nehm, R. H. (2019). Chinese pre-service biology teachers' evolutionary knowledge, reasoning patterns, and acceptance levels. *International Journal of Science Education*, *41*(5), 628–651.

Harris, Z. S. (1954). Distributional structure. *Word*, *10*(2–3), 146–162.

Husen, T. (1996). Lessons from the IEA studies. *International Journal of Educational Research*, 25(3), 207–218.

Jones, M. G., Paechter, M., Yen, C. F., Gardner, G., Taylor, A., & Tretter, T. (2013). Teachers' concepts of spatial scale: An international comparison. *International Journal of Science Education*, *35*(14), 2462–2482.

Kim, S. Y., & Nehm, R. H. (2011). A cross-cultural comparison of Korean and American science teachers' views of evolution and the nature of science. *International Journal of Science Education*, *33*(2), 197–227.

Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, *33*, 1159–1174.

Liu, O. L., Rios, J. A., Heilman, M., Gerard, L., & Linn, M. C. (2016). Validation of automated scoring of science assessments. *Journal of Research in Science Teaching*, *53*(2), 215–233.

Mayfield, E., & Rosé, C. P. (2012). *LightSIDE text mining and machine learning user's manual*. www.cs.cmu.edu/~emayfiel/LightSIDE.pdf

Mayfield, E., & Rosé, C. P. (2013). LightSIDE: Open source machine learning for text. In M. D. Shermis & J. C. Burstein (Eds.), *Handbook of automated essay evaluation: Current application and new directions* (pp. 124–135). Psychology Press.

Moharreri, K., Ha, M., & Nehm, R. H. (2014). EvoGrader: An online formative assessment tool for automatically evaluating written evolutionary explanations. *Evolution: Education and Outreach*, *7*(1), 1–14.

Mun, K., Lee, H., Kim, S. W., Choi, K., Choi, S. Y., & Krajcik, J. S. (2015). Cross-cultural comparison of perceptions on the global scientific literacy with Australian, Chinese, and Korean middle school students. *International Journal of Science and Mathematics Education*, *13*(2), 437–465.

National Research Council (1990). *A framework and principles for international comparative studies in education*. National Academies Press.

National Research Council (1993). *A collaborative agenda for improving international comparative studies in education*. National Academies Press.

National Research Council (2001). *Knowing what students know: The science and design of educational assessment*. National Academies Press.

National Research Council (2006). *America's lab report: Investigations in high school science*. National Academies Press.

National Research Council (2012). *A framework for K-12 science education: Practices, crosscutting concepts, and core ideas*. National Academies Press.

National Research Council (2013). *Next generation science standards: For states, by states*. National Academies Press.

National Research Council (2014). *Developing assessments for the next generation science standards*. National Academies Press.

Nehm, R. H. (2018). 14: Evolution. In M. Reiss & K. K. Kampourakis (Eds.), *Teaching biology in schools: Global research, issues and trends* (pp. 164–177). Routledge.

Nehm, R. H. (2019). Biology education research: Building integrative frameworks for teaching and learning about living systems. Disciplinary and Interdisciplinary. *Science Education Research*, *1*(1), 1–18.

Nehm, R. H., Beggrow, E. P., Opfer, J. E., & Ha, M. (2012). Reasoning about natural selection: Diagnosing contextual competency using the ACORNS instrument. *The American Biology Teacher*, *74*(2), 92–98.

Nehm, R. H., Ha, M., Großschedl, J., Harms, U., & Roshayanti, F. (2013, April 6–April 9). *American, German, Korean, and Indonesian pre-service teachers' evolutionary acceptance, knowledge, and reasoning patterns*. Paper in proceedings of the National Association for Research in Science Teaching.

Nehm, R. H., Ha, M., Rector, M., Opfer, J., Perrin, L., Ridgway, J., & Mollohan, K. (2010). *Scoring guide for the open response instrument (ORI) and evolutionary gain and loss test (EGALT)*. Technical Report of National Science Foundation REESE Project 0909999.

Opfer, J. E., Nehm, R. H., & Ha, M. (2012). Cognitive foundations for science assessment design: Knowing what students know about evolution. *Journal of Research in Science Teaching*, *49*(6), 744–777.

Organisation for Economic Co-operation and Development (1992). *The OECD international education indicators: A framework for analysis*. OECD.

Organisation for Economic Co-operation and Development (2019). *PISA 2018 results (Vol. I): What students know and can do*. OECD Publishing. https://doi.org/10.1787/5f07c754-en

Platt, J. (1998). *Sequential minimal optimization: A fast algorithm for training support vector machines*. Technical Report MSR-TR-98-14. Microsoft Research.

Rachmatullah, A., Nehm, R. H., Ha, M., & Roshayanti, F. (2018). Evolution education in Indonesia: Preservice biology teachers' evolutionary knowledge levels, reasoning models, and acceptance patterns. In H. Deniz & L. A. Borgerding (Eds.), *Evolution education around the globe* (pp. 335–356). Springer.

Sato, T., Yamanishi, Y., Kanehisa, M., & Toh, H. (2005). The inference of protein–protein interactions by co-evolutionary analysis is improved by excluding the information about the phylogenetic relationships. *Bioinformatics*, *21*(17), 3482–3489.

Simola, H. (2005). The Finnish miracle of PISA: Historical and sociological remarks on teaching and teacher education. *Comparative Education*, *41*(4), 455–470.

Su, L. T. (1994). The relevance of recall and precision in user evaluation. *Journal of the American Society for Information Science*, *45*(3), 207–217.

Zhai, X., Haudek, K. C., Shi, L., Nehm, R. H., & Urban-Lurain, M. (2020). From substitution to redefinition: A framework of machine learning-based science assessment. *Journal of Research in Science Teaching*, *57*(9), 1430–1459.

Zhai, X., Shi, L., & Nehm, R. H. (2021). A meta-analysis of machine learning-based science assessments: Factors impacting machine-human score agreements. *Journal of Science Education and Technology*, *30*(3), 361–379.

Zhu, Z., Pilpel, Y., & Church, G. M. (2002). Computational identification of transcription factor binding sites via a transcription-factor-centric clustering (TFCC) algorithm. *Journal of Molecular Biology*, *318*(1), 71–81.

13

Making Sense of College Students' Writing Achievement and Retention With Automated Writing Evaluation

Jill Burstein, Daniel F. McCaffrey, Steven Holtzman, and Beata Beigman Klebanov

1. Introduction

Writing is a critical competency in 4-year college curricula. Yet, it is acknowledged that many students lack the writing skills required in college (Graham, 2019). College retention remains a national concern (Hussar et al., 2020). To our knowledge, no existing research examines the relationship between college writing and retention. Automated writing evaluation (AWE) is the identification and generation of features that represent linguistic characteristics in a text. These features can be used to evaluate a writing sample, such as for automated essay scoring (Shermis & Burstein, 2003, 2013) This chapter discusses how AWE can be used to examine features in students' writing that may be associated with retention.

The chapter has two main goals. First, it examines the contribution of writing achievement in college retention. More broadly, the chapter demonstrates implications for using AWE for learning analytics research to inform curricular supports and interventions. Specifically, the chapter examines how AWE can be used beyond standard automated scoring and feedback, and suggests the implications for real-time student retention analytics. To do this, the chapter provides an illustration through the discussion of a study that examines the role of writing achievement as a retention factor. Through this writing achievement lens, the larger aim is to demonstrate how to build our understanding about how we can better prepare all students in writing who want to attend college, and support them so that they can graduate.

To conduct this research, we used AWE technology. While AWE typically has been used for automated essay scoring and feedback in instructional contexts, we used it to perform this writing analytics study. To do this, we generated AWE features in students' assessment and coursework writing, and examined relationships between the AWE features and broader student outcomes. Findings from this exploratory writing analytics study, discussed in the chapter, suggest implications for AWE-driven retention analytics that might be used to identify students at risk of dropping out. To situate our study in the larger retention research space, the chapter provides a general discussion of the college retention issue and previous survival analysis research.

DOI: 10.4324/9781003278658-17
This chapter has been made available under a CC-BY-NC-ND 4.0 license.

1.1 College Retention and Contributing Factors

The National Center for Education Statistics (NCES) defines college retention and graduation rates as follows:

> Retention rates measure the percentage of first-time undergraduate students who return to the same institution the following fall, and graduation rates measure the percentage of first-time undergraduate students who complete their program at the same institution within a specified period of time.
>
> (Hussar et al., 2020)

NCES reported that in 2017–2018, the percentage of students retained at approximately 2,330 four-year postsecondary degree-granting institutions varied by institutional selectivity (Hussar et al., 2020). Specifically, NCES reported that, on average, there was a retention rate of 97% at colleges with 25% acceptance rates, 62% at colleges with open admissions, and 81% across all institutions. Across all institutions, only 65% of first-time students in 2012 persisted to graduation within 6 years with variation across institution type (Hussar et al., 2020). The National Student Clearinghouse Research Center (2020) report shows for the Fall 2018 cohort entering a 4-year institution that there are disparities by race/ethnicity, with Asians (87.5%) showing greater retention than Whites (86%), and Latinx (71.8%) and Black (66.3%) populations showing lower retention rates.

Students' persistence in college may be influenced by a diverse and complex set of factors ranging from academic to wellness to finances. Stewart et al. (2015) examined college persistence at four large public research universities. Their findings showed that persistence had a statistically significant relationship to high school and first-semester college grade point average (GPA). Not surprisingly, Stewart et al. also point out that academically prepared students were more likely to persist. Implications from the Stewart et al. study suggested that a diverse set of support services such as tutoring, mentoring, counseling services, early intervention systems, and financial aid assistance would improve participants' academic deficiencies and increase persistence beyond the first year.

Hussar et al.'s (2020) report suggests a challenge with U.S. college retention. Further, the National Student Clearinghouse Research Center (2020) report suggests that college retention rates are lower for Black and Latinx students. As pointed out in Stewart et al. (2015) and Lederer et al. (2020), there is a complex set of issues that may contribute to retention. Although a variety of factors may play a role in college retention and graduation, this chapter applies a writing achievement lens to the retention equation. Specifically, we ask the question: What is the relationship between writing achievement and college retention and graduation? To answer this question, the chapter presents a survival analysis study. The study examines the relationship between AWE measures automatically derived from college students' assessment and coursework writing, and student retention.

1.2 Survival Analysis and College Retention

Survival analysis is a statistical analysis that relates the length of time that a unit (e.g., a student) survives (e.g., remains enrolled at a post-secondary institution) to a set of characteristics of the unit (e.g., a student's SAT scores or high school GPA; Hosmer & Lemeshow, 1999). Previous studies have used survival analysis to examine retention and associations with success indicators, such as outcomes measures (e.g., GPA), assessments predicting success (e.g., SAT score), and other student background information (e.g., race/ethnicity). The name, survival analysis, results from early uses of these modeling techniques to study mortality in health sciences. Two commonly used approaches for survival analysis are Kaplan-Meier (Kaplan & Meier, 1958) and Cox proportional hazards regression (Cox & Oakes, 1984). The Kaplan-Meier method models the probability that a person survives to each given time point as a function of a single discrete variable. For example, it would provide an estimate of the probability that college students in a first-year cohort with a high school GPA of below 3.0 versus 3.0 or higher remain enrolled on each day starting from their initial enrollment until the end of the subsequent 8 semesters. Cox proportional hazards regression models the hazard of failure, i.e., dropout, as a function of multiple characteristics. The hazard is the probability of failing at a given time having survived up to that time. For example, the Cox proportional hazard model would model the probability of remaining enrolled until tomorrow having remained enrolled until today given, for instance, as a function of a student's high school GPA, SAT scores, and ethnicity. Previous studies do not typically examine relationships between an academic proficiency (such as writing) or intra- or interpersonal factor subconstructs, and retention. The remainder of this section discusses prior survival analysis studies that examine college retention and broader success predictors and outcomes, high school support, and intrapersonal factors.

1.2.1 Success or Outcomes Predictors

Murtaugh et al. (1999) conducted a survival analysis study using Kaplan-Meier probabilities for univariate analysis, and Cox proportional hazards regression analysis for multivariate analysis to examine retention for 8,867 students who had enrolled at Oregon State University in the Fall terms from 1991 through 1995. Students were followed during this time, and the study used as the dependent variable the maximum time observed for a student across those years. The Kaplan-Meier analysis showed that over the study timeframe the university had an approximately 40% attrition rate. Hazards regression model findings suggested independent associations of student retention with age, residency, high school and first-quarter performance at the university, ethnicity/race, and enrollment in the university's Freshman Orientation Course. They point out that their strongest finding was the predictive value of high school GPA over SAT score. Their findings also show reduced retention with increasing age at enrollment, and a difference between the univariate and multiple-variable views of the association between ethnicity/race and retention. As a result of this study, the university implemented a new Freshman Orientation Course. Chimka and Lowe (2008) conducted survival analysis using Cox proportional hazards models to study student graduation from an engineering college. The study builds on earlier work (Chimka et al., 2007) that 'reported significance of standardized math test scores, gender and science ACT scores in explaining variation in student graduation based on main effects models of graduation controlling for descriptors such as in-state residence, hometown population, and student major'. The study followed 429 full-time students enrolled at the University of Oklahoma who entered in the Fall 1995 term; students were followed for six and a half years. Using datasets from Chimka et al. (2007), the authors evaluated survival based on standardized test scores: SAT and ACT scores. Key findings from the Cox proportional hazards model suggested that engineering students with higher English and science ACT scores were more likely to graduate.

1.2.2 High School Support Experiences

Ishitani and Snider (2006) conducted a secondary data analysis to investigate the longitudinal impact of high school experiences on college retention. These experiences were related to high school college preparation programs and teacher outreach related to college attendance. Study data were obtained from the National Education Longitudinal Study: 1988–2000 and NELS: 88/2000 Postsecondary Education Transcript Study (Adelman et al., 2003; PETS, 2000). Their sample contained 4,445 first-time students enrolled in 4-year institutions between 1992 and 1994 from the PETS dataset. These data contain information about students, such as gender, race, parents' education, income, educational expectation, high school ranking, participation in high school program (e.g., ACT/SAT prep course), and parents' involvement and first-year financial aid. Approximately 50% of students from this sample graduated in 4.2 years, 24.9% transferred, and 19.4% dropped out. Ishitani and Snider report a number of findings from an exponential model such as the following. The findings showed that Asian students were 32% less likely to drop out than White students, while Latinx, Black, and Native American students were 32%, 32%, and 42% more likely to drop out, respectively, than their White peers. This is consistent with the National Student Clearinghouse Research Center (2020) report discussed earlier. Findings showed that first-generation students were 82% more likely to drop out, and students with only one college education parent were 40% more likely to drop out. Further, students in the lower high school academic rankings were over two times more likely to drop out. With regard to high school experiences, Ishitani and Snider found that students who took ACT/SAT prep courses in high school were 33% less likely to drop out. The positive association between the participation in ACT/SAT preparation courses and college retention noted in an exponential model was shown to be stronger in years two and three (42% and 55%, respectively) using a period-specific model. Another type of support considered was high school teachers contacting parents about students' college selection. Using the period-specific model, students whose parents were contacted by high school teachers were 30% less likely to drop out in year four. The association between both these high school experiences, participation in ACT/SAT preparation courses and high school teachers contacting parents, and retention were significant after controlling for student characteristics and high school ranking.

1.2.3 Intrapersonal Factors

Alarcon and Edwards (2013) used discrete-time survival mixture analysis (DTSMA) to model student university retention with 584 students at a 4-year university who were enrolled in an introductory psychology course. DTSMA uses the conditional probability that a student will not drop out given that they did not drop out at an earlier point. They examine the relationship between retention and the factors of parent education, gender, ACT scores, conscientiousness, and trait affectivity. The authors assert that the 'dutifulness' aspect of conscientiousness is associated with motivation, and because of this, they hypothesized that this factor would be positively related to retention. They also hypothesized that negative affect would be negatively associated with retention. Conscientiousness was measured with the nine-item Conscientiousness subscale from the Big Five Inventory (John & Srivastava, 1999) (e.g., 'I see myself as someone who perseveres until the task is finished'). The 20-item Positive and Negative Affect Schedule (PANAS) (Watson et al., 1988) was used to measure affectivity. Questions ask respondents to rate an experience or emotion on a 5-point scale from very slightly or not at all (1) to extremely (5). Retention was measured based on quarterly enrollment. Findings indicated that ability and motivation were associated with university retention with the most significant predictors of retention being ACT score, and positive and negative affectivity.

This section describes previous survival analysis research representative of the types of studies that have been conducted. There does not seem to be a substantial body of work that examines specific academic competencies (and competency features), and retention. Instead, research has largely focused on success predictors (such as SAT scores and GPA), and demographic factors with some attention to intrapersonal factors. Our contribution is an exploratory survival analysis study that drills down and examines AWE features related to student writing achievement factors and retention.

1.3 Sociocognitive Model of Writing Achievement

Writing achievement is commonly viewed as a sociocognitive construct (Flower, 1994; Hayes, 2012; White et al., 2015). More specifically, it stresses the interplay between writing domain knowledge, general skills and content knowledge, and intra- and interpersonal factors. Writing domain knowledge focuses on the students' proficiency in writing processes (e.g., planning, drafting, revising) and proficiency in different aspects of writing, such as topic development, organization, argumentation, and use of English conventions. General skills are associated with, for instance, reading and critical thinking, and content knowledge addresses students' ability to discuss information in a subject area. Intrapersonal factors include aspects such as self-regulation, confidence, interest, motivation, and engagement; and interpersonal factors include, for instance, collaboration on writing tasks.

In this study, we focus on relationships between writing domain knowledge as captured by AWE, and college success indicators and retention.

2. Automated Writing Evaluation

Automated writing evaluation – AWE – is a technology that uses natural language processing (NLP) and language resources to process and analyze textual data in educational contexts. These data may include, for instance, essay writing items from educational assessments and coursework writing from naturalistic settings, such as college writing courses. Examples of NLP capabilities used for AWE are detection of argumentation (Beigman Klebanov, Gyawali, et al., 2017; Ong et al., 2014), discourse structure analysis (Burstein et al., 2003), discourse coherence quality analysis (Foltz et al., 1998; Somasundaran et al., 2014), and grammatical error detection (Leacock et al., 2010).

AWE has its roots in automated essay scoring (Beigman Klebanov & Madnani, 2020; Burstein et al., 1998; Foltz et al., 1998; Page, 1966). The technology dramatically changed the practice of high-stakes essay scoring (Burstein et al., 1998); tests such as the GMAT, GRE, and TOEFL standardized assessments use AWE or a combination of human and AWE scoring for writing. Through the emergence of increasingly sophisticated AI capabilities, AWE methods now produce advanced feedback used in a growing number of educational writing support applications, such as ETS's Criterion online essay evaluation system (Burstein et al., 2004), ETS's Writing Mentor Google Docs add-on (Burstein et al., 2018), Pearson's Write-to-Learn (Foltz & Rosenstein, 2017), Measurement Inc's MI Write (Bunch et al., 2016), and Turnitin's Revision Assistant (Woods et al., 2017).

2.1 AWE, Writing Analytics, and Broader Success Factors

While AWE has its roots in assessment and instruction, a growing body of research demonstrates how AWE can support educational data mining and writing analytics research. Such research can inform our understanding of writing achievement and its broader connections to skills, attitudes, and student outcomes. For instance, in a study with undergraduates at a U.S. university, Allen et al. (2014) examined relationships between reading comprehension and writing features based on AWE measures. The authors found statistically significant relationships between the AWE composite feature that captured characteristics of vocabulary used in college student writing and writing scores (correlation of 0.57, $p < .001$) and reading comprehension (correlation of 0.79, $p < .001$) measures. Using data from 108 university students, Allen et al. (2016) investigated how linguistic properties in students' writing can be used to model individual differences in postsecondary students' vocabulary knowledge and comprehension skills. Linguistic feature values from essays were computed using the ReaderBench framework – an automated text analysis tool that calculates linguistic and rhetorical text indices on argumentative essays in response to a standardized test prompt. The authors tested for the relationship between these automated feature scores and students' Gates-MacGinitie Vocabulary and Reading Comprehension tests and found that five features accounted for 45.3% of the variance in vocabulary scores [$F(5, 102) = 16.927$, $p < .001$; $R^2 = .453$] and three accounted for 36.3% of the variance in comprehension scores [$F(3, 104) = 19.758$, $p < .001$; $R^2 = .363$].

Perin and Lauterbach (2018) examined writing skills of low-skilled adults attending college developmental education courses by determining whether variables from an automated scoring system were predictive of writing quality. The authors applied the Coh-Metrix system (Graesser et al., 2004; McNamara et al., 2010) to students' written coursework to study whether Coh-Metrix features predicted writing skills as measured by a human holistic score rating. Coh-Metrix analyzes over 50 variables related to linguistic variables related to text cohesion, syntactic complexity, lexical diversity, and word frequency. They identified 10 Coh-Metrix variables from among these categories that were predictive of writing skill. Beigman Klebanov, Burstein et al. (2017) used AWE to extract features associated with utility value from student writing samples collected from a utility value writing intervention study in an undergraduate STEM setting (Harackiewicz et al., 2016). Utility value is the idea that a person can relate a topic to their own life. Utility value in the writing samples expressed how a biology topic related to a student's life. The authors showed that the presence of words identified as being associated with utility value (e.g., our, we, people, brother, family, children) were predictive of writing responses with higher utility value scores assigned by human raters. The automated assessment of utility value may be important because the Harackiewicz et al. (2016) study showed that in the context of the study, higher utility value scores in students' writing was correlated with course success and progression to a higher-level biology course.

In a set of papers from a study of university students, Burstein and colleagues (Burstein et al., 2017, 2019, 2020) examined relationships between writing achievement as measured by about 50 AWE features related to basic English conventions skills, discourse structure, coherence, vocabulary usage, and utility value (i.e., personal reflection) and broader academic skill and success factors, such as SAT scores, and high school GPA and cumulative college GPA. These studies use both standardized writing assessment and coursework writing samples. Varying subsets of features were found to be predictive of the success and skill factors studied. Vocabulary features were identified as statistically significant predictors (at levels between $p <.05$ and $p < 0.0001$) for many of the success and skill factors. For example, a standard deviation increase in a feature measuring vocabulary choice was associated with a 0.20 standard deviation increase in college GPA, and a standard deviation increase in a feature measuring vocabulary sophistication was associated with a 0.18 standard deviation increase in college GPA (Burstein et al., 2019). Ling et al. (2021) examined relationships between AWE feature subconstructs present in college student writing for Vocabulary, English Conventions, Organization and Development, Argumentation, Sentence Structure, and Utility-Value language (see Section 3.1.3) and writing motivation factors, including self-efficacy, writing goals (e.g., student desire to master goals, or goal avoidance), writing beliefs (e.g., the belief in the importance of convention, or that good writing is about following conventions, or the belief in the importance of content, or that writing helps people express or discover ideas), and writing affect (e.g., student enjoyment of writing). Ling et al. used essay writing data collected from the HEIghten Written Communication assessment from 325 university students from across five 4-year institutions. Students also responded to a writing motivation survey. Analysis of correlations between the modeled motivation constructs and specific features of writing identified by AWE were conducted in this study. Correlations were statistically significant (at levels between $p < .05$ and $p < .01$). for AWE features associated with Utility Value language, Vocabulary, English Conventions, and the motivation constructs identified in the survey. Writers who used more Utility Value language and had higher English Conventions scores (i.e., fewer errors) had greater beliefs that content was important to writing.

By contrast, Vocabulary and English Conventions were negatively correlated with beliefs in the importance of writing conventions. Students with lower Vocabulary and English Conventions scores tended to believe more in the importance of low-level aspects of writing. The Avoidance construct was negatively correlated with Vocabulary and English Conventions features suggesting that writers with lower Vocabulary and English Conventions scores (i.e., more errors) were more likely to respond that they avoided writing. Confidence was positively correlated with Vocabulary and English Conventions features, indicating that students with higher scores on Vocabulary and English Conventions responded that they were more confident about writing. Results were consistent with broader achievement measures (e.g., GPA) also measured in the study. Consistency with the results for the broader achievement measures suggests a meaningful relationship between motivation and writing performance. See Ling et al. (2021) for a more detailed discussion of this work.

3. Study

As mentioned earlier in the chapter, this study is motivated by the national concern about low college retention. Further, many students enter postsecondary schooling without the prerequisite writing skills generally considered necessary for college-level courses, and many continue to struggle with developing writing skills throughout their college career. These students are likely at greater risk for poor academic performance and dropout. Yet, there are no direct studies of the relationship between students' writing skills and retention. One of the challenges with such studies is collecting and evaluating writing data from students.

This study tries to address a research gap by collecting writing samples and enrollment data from a sample of students attending multiple universities to examine relationships between writing achievement and college retention. The study is an example of how AWE can play a role in understanding how the education system can better prepare all students to succeed in college. To conduct the study, writing samples were evaluated using AWE to produce an array of quantitative measures of the characteristics of the writing, such as writing conventions or vocabulary usage. One goal of the exploratory study was to determine if the information about writing skills produced by AWE features might prove useful for studying postsecondary student educational outcomes such as retention. If the AWE features prove useful, then they could support further research into the contribution of writing as a student success factor. Further, AWE features might support writing analytics research that could provide information about specific weaknesses in writing skill that potentially could be mitigated through targeted intervention. The study also provides preliminary evidence about the relationship between writing skills and college retention.

Following previous work on postsecondary retention, the study uses survival analysis to explore the relationships that exist between college retention and writing skills as measured by AWE features, while controlling for other factors known to be related to retention, including high school GPA, and ACT or SAT scores.

3.1 Methods

3.1.1 Participants

Six 4-year public universities participated in the study. One site was a historically Black college or university (HBCU), and a second site was a Hispanic-serving institution. All six universities had enrollments that were predominantly undergraduate (at least 75%) and majority female. They are rated mostly as 'inclusive' or 'selective' in terms of their acceptance policies by the Carnegie Classification. Four of the universities predominantly enrolled White students. Most students at the HBCU were Black and the majority of students at the Hispanic-serving institution were Hispanic.

For the study, a sample of courses which required at least two extended writing products were selected, and students enrolled in those courses were invited to participate. Participating students were asked to upload their written coursework from the participating course to a project web portal. Students were also asked to complete the HEIghten Written Communications[1] and Critical Thinking Assessments,[2] complete a survey on writing beliefs and motivation, and grant the study permission to collect personal data from the institutional data system. This study includes data from 476 students. Of these students, 418 students submitted one or more coursework writing assignments and 366 completed the essay in the HEIghten assessment, 308 did both. The study sample is 60% female, 4% Asian, 39% Black, 17% Hispanic, 33% White, and 8% Other Race or Ethnicity, including Unspecified. The racial/ethnic composition of the sample differed substantially across the universities. Table 13.1 contains an overview of the university student populations and the study sample.

3.1.2 Data

Data were collected from students in the Fall 2017, Spring 2018, and Fall 2018 semesters. Multiple semesters were included to increase the student sample size. Each student participated for only one semester uploading their writing assignments and completing their HEIghten tests and surveys during that semester. Institutional data were collected for students for up to five additional semesters through the end of the 2019–2020 school year. The data consists of the background data on the 476 students, including basic demographics such as gender and race/ethnicity, high school grade point average (GPA) converted to the standard 4-point scale, and SAT or ACT scores (when available). ACT scores were projected onto the SAT score scale using a concordance table (ACT, n.d.). They also include the students' college GPA at the start of their participating semester, and from the end of each semester for their participating semester until the end of the study. At each data collection (i.e., at the end of each semester during the study window), the students' enrollment status was coded as enrolled, dropped out, or graduated. The data contains seven measures of students' intrapersonal factors derived from student responses to the survey which included 50 items on writing attitudes, beliefs, and motivation. Details on the measures and the survey are found in Ling et al. (2021).

In terms of students' writing, the data included 997 coursework writing samples for which AWE features were generated (see Section 3.1.3). Assignments were from one of these courses: first-semester English composition, business, history, and STEM, and from argumentative, informative, or reflective genres (Burstein et al., 2019). Median coursework assignment word count was 753. Each writing sample was classified using a coding scheme developed by the research team as being of one of three genres: (1) argumentative; (2) informational; or (3) reflective. The data also include students' responses and scores from the HEIghten WC. The HEIghten Written Communication (WC) test measures four dimensions of a test-taker's written communication skills: knowledge of social and rhetorical situations, knowledge of conceptual strategies, knowledge of language use and conventions, and knowledge of the writing process. Each participating student completed one of two HEIghten WC test forms in a 45-minute testing session. The test consists of 24 multiple-choice items and an argumentative essay task to support their position on a topic with reasons and examples in response to a position given in the test prompt. The students' HEIghten WC essays were also evaluated using AWE.

3.1.3 Automated Writing Evaluation Features

In this study, AWE tools were used to generate 36 AWE features representing six writing subconstructs: Vocabulary (e.g., word complexity), English Conventions (e.g., grammar

Table 13.1 Characteristics of Participating Universities and Students

					Gender	Race/Ethnicity				
Site	Classification	Enrollment Profile/ Undergraduate Profile	Student Sample and Population		Percent Female	Asian	Black or African American	Hispanic/ Latinx	White	Other/ Unknown
Institution 1	Master's Colleges and Universities: Larger Programs	Very high undergraduate; 4-year, full-time, selective, lower transfer-in	Population	~9,000	58.0	1.1	6.6	7.4	77.1	7.8
			Study Sample	11	54.5	0	0	9.1	81.8	9.1
Institution 2	Master's Colleges and Universities: Larger Programs; Historically Black University	High undergraduate; 4-year, full-time, inclusive, higher transfer-in	Population	~6,000	60.6	1.5	82.0	4.5	2.0	10.1
			Study Sample	173	63.6	0.6	89.0	4.6	1.2	4.6
Institution 3	Doctoral/Professional Universities	High undergraduate; 4-year, full-time, inclusive, higher transfer-in	Population	~24,000	59.0	12.9	2.6	53.9	17.8	12.8
			Study Sample	109	51.4	15.6	3.7	63.3	9.2	8.3
Institution 4	Master's Colleges and Universities: Larger Programs	Very high undergraduate; 4-year, full-time, selective, higher transfer-in	Population	~9,000	59.2	0.8	19.1	0.5	68.9	10.7
			Study Sample	140	60.7	0.7	18.6	0.7	72.9	7.1
Institution 5	Master's Colleges and Universities: Larger Programs	Very high undergraduate; 4-year, full-time, selective, higher transfer-in	Population	~9,000	55.7	1	4.6	2.5	85.3	6.7
			Study Sample	16	68.8	6.3	0.0	6.3	81.3	6.3
Institution 6	Doctoral Universities: High Research Activity	Very high undergraduate; 4-year, full-time, more selective, higher transfer-in	Population	~17,000	63.1	2.1	4	7.5	77.7	8.7
			Study Sample	27	70.4	0	3.7	11.1	74.1	11.1
Total			Population	~74,000	59.3	3.2	19.8	12.7	54.8	9.6
			Study Sample	476	60.3	4.2	38.9	17.4	32.8	6.7

[1] Classification, enrollment profile, and student population information were obtained online from Carnegie Classification of Institutions of Higher Education. Gender and ethnicity distributions were obtained from National Center for Education Statistics IPEDS College data, 2019–2020.

errors), Organization and Development (e.g., text coherence), Argumentation (e.g., claim terms), Sentence Structure (e.g., use of clauses), and Utility-Value Language, or UVL (see Section 2; McCaffrey et al., forthcoming). See Table 13.2.

The features were derived for each coursework writing sample and each HEIghten argumentative essay response separately. To create a univariate measure for each subconstruct, the individual feature measure scores were combined into a weighted composite score. Weights equaled the loadings of the first principal component from a principal components analysis fit separately for each of the six subconstructs. Because feature distributions differed by the genre of the writing sample, the individual features were centered by genre to have mean zero prior to the calculation of the PCA weights and creation of the composite scores. The final composite scores were standardized to a mean of zero and a variance of one and averaged across writing assignments to yield one score per composite per student. Analyses were run at the student level, and separately for the HEIghten and course writing data.

4. Predicting Dropout

Participating students' enrollment was tracked for three to five semesters after their participation in the study using administrative data provided by the participating universities. A series of random effects Cox proportional hazards regression was used to model dropout as a function of the students' SAT/ACT score, high school GPA (HSGPA), university, AWE subconstruct composite score, and writing sample length (McCaffrey et al., forthcoming). The models also include random effects for the course section in which students were enrolled when participating. This accounted for possible unmodeled dropout risk factors associated with different assignments of students to different courses and section, which might occur if some courses or section were targeted for students needing remedial supports. On the basis of the previous studies of retention (Chimka & Lowe, 2008; Murtaugh et al., 1999), a model that included only university, SAT/ACT score, and high school GPA was fit.[3] Because the previous literature found race/ethnicity was also associated with dropout a variation of the model that included race/ethnicity was then fit, as was a model that included gender.

A series of separate models were then fit to explore the relationship between the composite writing features and dropout. First, six proportional hazard models (Hosmer & Lemeshow, 1999)[4] were fit to predict dropout using each of the composite features for coursework assignments separately. In these models, the probability that a student dropped out at any given time interval after enrolling in the study (e.g., at the end of the next semester) having remained enrolled until that start of the interval (i.e., the hazard function) was modeled as a function

Table 13.2 36 AWE Feature Descriptions

Two Argumentation features (ARG Features 1–2) that quantify the following:

1. Average number of claims.
2. Average number of claim verbs from an extended discourse cue lexicon from Burstein et al. (1998).

Seven English Conventions features (CNV Features 1–7) that quantify different aspects of English conventions based on:

1. Normalized, aggregate measure of grammar error counts (Attali & Burstein, 2006).
2. Normalized, aggregate measure of mechanics error counts (Attali & Burstein, 2006).
3. Normalized, aggregate measure of word usage error counts (Attali & Burstein, 2006).
4. Aggregate proxy measure for overall errors in grammar, word usage and mechanics.
5. Presence of contractions (Burstein et al., 2018).
6. Aggregate measure related to collocation and preposition use (Burstein et al., 2013).
7. Words and expressions related to a set of 13 'unnecessary' words and terms (such as 'very', 'literally', 'a total of') (Burstein et al., 2018).

Seven Organization and Development features (OD Features 1–7) that quantify the following:
1. Number of discourse text segments most likely in argumentative writing (such as, thesis statements, main points, supporting details, and conclusion statements; Attali & Burstein, 2006).
2. Length of discourse text segments most likely in argumentative writing (such as, thesis statements, main points, supporting details, and conclusion statements; Attali & Burstein, 2006).
3. Proportion of sentences directly associated with an argument (Beigman Klebanov et al., 2017).
4. Distribution of topical keywords (Beigman Klebanov & Flor, 2013).
5. Discourse quality (Somasundaran et al., 2014).
6. Keywords associated with the largest topic (Beigman Klebanov & Flor, 2013; Burstein et al., 2016).
7. Pairs of words in the text that are strongly semantically related based on the pointwise-mutual information measure (Beigman Klebanov & Flor, 2013; Burstein et al., 2016).

Three Utility Value Language features (UVL Features 1–3) based on Beigman Klebanov, Burstein, et al. (2017) that capture the instances of language that writers can use in personal reflections of the value of content to their personal lives. These measures quantify the following:
1. Use of argumentative connectives (e.g., furthermore) and narrative elements (e.g., past tense verbs) which are often used in personal stories.
2. Everyday vocabulary related to the extent and specificity of personal reflection that connects course material to the writer's personal life (e.g., family).
3. Using grammatical categories that express reference to self and other people using first-person singular (e.g., I, mine) and plural pronouns (e.g., we, ourselves), and second-person pronouns (e.g., you), possessive determiners (e.g., their), and indefinite pronouns (e.g., anyone).

Six Sentence Structure features (SSTR Features 1–6) (Madnani et al., 2016) that measure normalized counts of the following:
1. Longer prepositional phrases containing at least two adjacent prepositional phrases (e.g., 'The cat sat in the box on the table').
2. Longer sentences containing one independent clause, and at least one dependent clause.
3. Complex verbs (e.g., did leave).
4. Complex noun phrases can be one of two kinds of structures (e.g., highly qualified teacher, the teacher of the year).
5. Relative clauses and the noun referent for their pronoun.
6. Passive sentences.

One Sentence Structure feature (SSTR Feature 7) that measures the following:

7. Sentence variety (Deane, 2021).

Ten Vocabulary features (VCB Features 1–10) that quantify the following:
1. Average word length (Attali & Burstein, 2006).
2. Number of terms that belong to homonym sets (e.g., to, too, two; Burstein et al., 2004).
3. Number of inflected word forms (Burstein et al., 2017).
4. Number of derivational word forms (Burstein et al., 2017).
5. Number of pronouns.
6. Number of stative verbs (i.e., express states vs. action, e.g., feel vs. break; Burstein et al., 2017).
7. Presence of positive and negative sentiment vocabulary terms from the VADER corpus (Burstein et al., 2017).
8. Vocabulary richness, using an aggregate feature composed of a number of text-based vocabulary-related measures (e.g., morphological complexity, relatedness of words in a text; Burstein et al., 2017; Deane, 2021).
9. Verbs used metaphorically (Beigman Klebanov et al., 2015, 2016).
10. Aggregate measure related to word frequency (Attali & Burstein, 2006).

of just one of the composite features extracted from their coursework. A positive coefficient on the feature indicates that higher values of that feature in a student's coursework writing is associated with greater risk of drop in each interval. Each model also contained indicators for the student's university and the square root of the average length (number of words) of the student's written coursework, since some AWE features are correlated with the length of the writing sample.[5] A similar set of six models were then fit for the AWE features from the HEIghten essays and the square root of the length HEIghten essay. Models with one of the features scores and the baseline model variables, (i.e., university, HSGPA, and SAT/ACT scores and the associated imputation flags) and the square root of the average length of the student's coursework writing samples or HEIghten essay were then fit. Again, these models were fit separately for each of the composite features from coursework and HEIghten essays. To determine if the writing data provided unique information that was not captured in an overall measure of student performance such as their university GPA, a third series of 12 models which added the students' university GPAs at the end of their participating semesters to each model with an AWE composite feature, the square root of the average writing sample or HEIghten essay length, and the baseline model variables.

5. Results

5.1 Baseline

Across all six institutions, 332 (45%) of the 735 students in the sample had dropped out before either graduating or the end of the study's data collection. The dropout rates ranged from 24% of the sample from Institution 6 to 59% of the sample from Institution 2 (the rates were 52%, 40%, 45%, and 36% for Institutions 1, 3, 4, and 5 respectively).

Table 13.3 presents the results for the prediction of dropout using only students' background variables. The table presents the hazard ratio – the change in the probability that a student who survived to the current period drops out given a one-point increase in the background variable. For example, the hazard ratio is 0.562 for HSGPA. This means that, based on the survival model, among students who remained enrolled at the end of each semester, the probability that students with 3.0 HSGPA drop out in the coming period is only 56.2% as large as the probability that students with 2.0 HSGPA drop out. Consistent with previous research, HSGPA is a statistically significant predictor of dropout or retention, whereas SAT/ACT scores are not, when also controlling for HSGPA and institution. This does not mean that the SAT is not associated with retention more generally. It only indicates that within an institution, in our sample, the SAT did not provide additional information beyond HSGPA. There are also some differences in the dropout risk among the students in the sample from the different

Table 13.3 Dropout Models With Background Variables

	Baseline Model		**Baseline Model with Race/Ethnicity**		**Baseline Model with Gender**	
Variable	Hazard Ratio	p-value	Hazard Ratio	p-value	Hazard Ratio	p-value
Institution 1	1.793	0.179	1.894	0.152	1.811	0.172
Institution 2	2.062	0.023	1.844	0.090	2.078	0.022
Institution 3	1.701	0.086	1.380	0.374	1.695	0.088
Institution 4	2.053	0.021	2.213	0.016	2.057	0.021
Institution 5	1.496	0.264	1.524	0.267	1.478	0.281
SAT/ACT Score	1.000	0.373	1.001	0.279	1.000	0.402
Flag for Imputed SAT/ACT Score	0.998	0.990	0.993	0.969	1.000	0.998
HSGPA	0.562	0.000	0.573	0.000	0.570	0.000
Flag for Imputed HSGPA	0.831	0.370	0.855	0.475	0.830	0.369
Race/Ethnicity: Hispanic			1.154	0.726		
Race/Ethnicity: Asian			0.841	0.723		
Race/Ethnicity: Black			0.953	0.898		
Race/Ethnicity: White			0.738	0.445		
Male					1.080	0.505

Note: The hazard ratios for Institutions 1 to 5 are relative to Institution 6, so there is no hazard ratio for Institution 6. The hazard ratios for Race/Ethnicity are relative to Race/Ethnicity: Other (e.g., two or more races, Native American, or not specified) so there is no hazard ratio Race/Ethnicity: Other.

institutions. The hazard ratios in the table are relative to Institution 6. For instance, the estimated hazard ratio for Institution 1 compared with Institution 6 is 1.793. That is, the risk of dropout at time point is 1.793 times higher for sample students from Institution 1 than from Institution 6. In other words, sample students from Institution 1 are about 79% more likely to drop out at a time point having remained enrolled than sample students from Institution 6. All the hazard ratios are positive, indicating that the small sample of students from Institution 6 have the lowest dropout risk. Although there are some small differences in the risk of dropout among the racial/ethnic groups after controlling for the other variables in the model, they are not statistically significant. Similarly, gender is not associated with dropout after controlling for HSGPA, SAT/ACT, and institution. The difference between the current results for race/ethnicity and the previous literature may be related to the fact that the sample includes only students who agreed to participate and uploaded their coursework writing or completed the HEIghten test, and the fact that there was limited racial/ethnic diversity within the sample from each university.

5.2 AWE Features

Table 13.4 contains a summary of the results for models with AWE composite features. In the series of models testing the relationship between composite writing features and dropout without controlling for the baseline model variables (other than university indicators and the square root of the average sample or the HEIghten essay length), only the UVL and the Vocabulary composite features were associated with students' risk of dropping out; other composite features were not. Increased use of UVL in either coursework or the response to the standardized HEIghten assessment was associated with a higher risk of dropout. A standard deviation increase in the UVL composite feature was associated with a 28% (coursework, p = 0.007) or 27% (HEIghten, p = 0.048) increase in the hazard of dropout. An increased use of more sophisticated vocabulary in the HEIghten essay was associated with a lower risk of dropout. A standard deviation unit increase is associated with a 16% (p = 0.032) decrease in the hazard of dropout. The results remain nearly the same after controlling for the baseline model background variables of HSGPA and SAT/ACT scores, although the p-values for the HEIghten essay features are now both over 0.05 at 0.066 and 0.072 for UVL and Vocabulary, respectively.

Table 13.4 Summary of Survival Model Results for Models With Composite AWE Features

	Models Without Baseline Variables[a]				**Models With Baseline Variables**[b]			
	Coursework		**HEIghten Essay**		**Coursework**		**HEIghten Essay**	
Composite Feature	**Hazard Ratio**	**p-value**	**Hazard Ratio**	**p-value**	**Hazard Ratio**	**p-value**	**Hazard Ratio**	**p-value**
Argumentation	0.930	0.410	0.935	0.397	0.928	0.399	0.927	0.350
Organization and Development	0.993	0.966	0.961	0.629	1.015	0.931	0.986	0.861
English Conventions	0.944	0.503	1.099	0.249	0.965	0.683	1.125	0.156
Utility-Value Language	1.280	0.007	1.274	0.048	1.265	0.012	1.255	0.066
Sentence Structure	0.937	0.406	0.969	0.689	0.936	0.393	0.957	0.586
Vocabulary	0.918	0.339	0.843	0.032	0.931	0.435	0.859	0.072

Notes: [a] Results are from 12 different models, each containing one AWE composite feature from either coursework or the HEIghten essays.
[b] Results are from 12 different models, each containing one AWE composite feature from either coursework or the HEIghten essays, HSGPA, SAT/ACT scores, flags for imputed values for HSGPA, and SAT/ACT score. All models include indicator variables for university, the square root of the average sample length (for coursework) or the essay length (HEIghten), and random effect for course section.

In the model that included students' university GPA at the end of their participating semester, university GPA was strongly related to their risk of dropout, as to be expected since academic performance in university is well-known to be related to retention. However, even controlling for this measure of general university academic performance, the associations between UVL and Vocabulary and dropout remains nearly as large as in other models. For the coursework, the hazard ratio for UVL for the coursework is 1.185 (p = 0.066); and for the HEIghten essays, the hazard ratios are 1.225 (p = 0.109) and 0.883 (p = 0.132), respectively, for UVL and Vocabulary. The hazard ratios are similar to the values in Table 13.4 for the other composite features with the exception of Conventions, which has a hazard ratio of 1.199 (p = 0.033) for the HEIghten essays. A final model that included the baseline variables and the students' scores on the HEIghten multiple-choice items was fit to test whether any measure of writing would be predictive of dropout. The HEIghten multiple-choice score was not statistically significant in that model. Hence, students' written work provides information that is not captured in a general measure of writing domain knowledge.

6. Summary

It is well-known that poor performance in postsecondary education is related to dropout, and that writing is considered an essential skill for success in university-level coursework. However, no studies to date have explored the relationship between student writing and retention.

In the current study, AWE generated writing subconstruct features for students' written coursework and responses to a standardized, argumentative writing performance assessment. AWE features proved useful, predicting students' retention for up to five semesters after the study semester (i.e., the semester in which the writing samples were collected). Two of the six composite writing subconstruct measures were related to retention, and both related to vocabulary. Greater use of personalized language associated with expressing utility value (i.e., higher values of the UVL composite) in either coursework or the standardized assessment writing sample was associated with increased risk of dropout. Greater use of more sophisticated vocabulary (i.e., higher values of the Vocabulary composite) in the standardized assessment was associated with a reduced risk of dropout. The UVL and Vocabulary features appear to be providing somewhat unique information. A standardized multiple-choice writing assessment score did not predict dropout, and the relationships between the features and dropout held after controlling for student background characteristics, including high school GPA, a known correlate of dropout, and university GPA, although controlling for university GPA did weaken the relationships.

These findings suggest that students' use of vocabulary in their written work appears to be a distinct predictor of retention that is not captured in the other measures of students' background, academic performance, or even a measure of writing domain knowledge. The relationship between UVL and college writing has not been widely studied. Canning et al. (2018) show that a UV writing intervention where students are asked to explicitly connect the STEM material they are studying to their lives is an effective intervention for promoting performance and retention in STEM courses; in contrast, our findings suggest that UV-like language, when used in college writing, is associated with a negative outcome of dropping out. Qualitative reviews of a few student writing samples from study participants illustrated that UVL use reflected difficulty effectively integrating personal elements into academic writing. As discussed earlier, vocabulary has been found to be associated with various measures of academic skills and success. The results from this study extend those findings. However, these are still predictive relationships. They do not necessarily indicate the improving students' vocabulary or use of UVL in writing would reduce their risk of dropping out. The AWE features could be proxies for other factors that affect students' use of vocabulary and UVL and their success and persistence in postsecondary education.

Overall, study findings have implications for how AWE can be productively used beyond instruction and assessment. The findings suggest that information gathered from AWE applied to samples of students' written work can produce information related to risk of dropout that is not fully captured by standard sources (such as high school GPA). This insight has implications for AWE as a potential means to gather diagnostic retention analytics for stakeholders who monitor students' progress. For example, we could envision AWE integration into a learning management system in order to provide not only personalized learning for writing, but retention analytics for students, educators, and other stakeholders to signal potential obstacles in real time, so that they can be quickly addressed.

7. Discussion

AWE has primarily been used in large-scale, high-stakes writing assessments to provide an automated means for efficient, reliable, and construct-relevant essay scoring, and in writing instruction applications to provide diagnostic feedback (similar to instructor feedback) to support the writing process (i.e., reflection and revision). As discussed earlier in the chapter, AWE has been used in previous work to conduct writing analytics research that has made connections between writing achievement factors and student outcomes and success predictors. A key aim of this chapter was to demonstrate how AWE could be used more broadly to conduct learning analytics research that can help support and retain students during their college careers. Specifically, this chapter illustrates how AWE was used to explore relationships between college students' writing achievement and retention, as motivated by the low retention rates at U.S. universities. Survival analysis findings presented in this chapter suggest that linguistic information in student writing may contribute to at least some part of the fuller, low-retention story.

Our survival analysis that examined relationships between writing achievement and retention illustrates how we were able to leverage AWE and show relationships between writing achievement and retention. The AWE features provided information about student retention not found in other measures of academic success or a standardized multiple-choice writing assessment. This work, however, is situated in a larger landscape, and has powerful implications. Learning analytics relies on the collection of large dataset from students. In postsecondary institutions, learning management systems are common parts of the university ecosystem, and streamline the collection of vast amounts of student data. These data add to the standard administrative data that has been typically accessible for research (such as demographic, GPA, and enrollment data). In addition to process data, such as how students spend time engaged with these learning management systems, and navigate within the systems, student writing assignment and assessment data can be collected in learning management systems. For instance, it is standard practice in many classes for students to upload writing assignment response data into university learning management systems (such as Canvas). These writing data can then be evaluated using AWE. This is already common practice, as the writing data often are automatically directed to AWE systems, such as Turnitin, to check for plagiarism. Universities also track how students engage with remote testing platforms to mitigate cheating on assessments. At the same time, large-scale, high-volume data collection has become standard, and AI methods and the computing power required to power these methods is readily available.

Taking this a step further, as Institute for Education Sciences director Mark Schneider points out (Schneider, 2021): 'There is emerging evidence in medicine that AI is on par with human clinicians in diagnosing problems and suggesting effective treatments'. He suggests that if AI methods can be used for medical diagnosis, the educational research community should be taking steps to figure out how this can be done for education. The implication is that given the availability of big data and rapid AI advances, educators and researchers should be thinking deeply about how to use these available data, AWE methods, and AI modeling to learn more about students with the goal of providing support. To that end, postsecondary institutions could leverage the speed of data collection, access to student writing data across disciplines and genres, AWE analyses, and AI modeling to rapidly identify obstacles to success and retention. Doing so has the potential to help to remove persistent barriers that continue to hold back many students as they strive to succeed. This may better target resources for student groups who have been underserved and have documented lower retention rates. The near-universal use of learning management systems for online submission of students' written work at universities and colleges makes postsecondary institutions the natural first candidates for exploring AWE-based learning analytics. However, collecting data and applying AWE and AI methods to students' writing as early as elementary school and following students through middle and high school could have positive implications for catching problems earlier. Ultimately, this could support prerequisite skill building in writing early enough so that college success would be a more likely outcome.

Acknowledgments

Research presented in this paper was funded by the Institute of Education Science, U.S. Department of Education, Award Number R305A160115. Any opinions, findings, conclusions, or recommendations are those of the authors and do not necessarily reflect the views of the IES. We are grateful to our partner institutions who made this work possible: Bloomsburg University of Pennsylvania, Bowie State University, California State University–Fresno, Jacksonville State University, Slippery Rock University, and University of North Carolina–Wilmington.

Notes

1 www.ets.org/heighten

2 The Critical Thinking scores are not used in the analyses in this chapter.

3 Some students' institutional data did not include SAT/ACT scores or HSGPA. When fitting the survival models, these students were assigned the mean SAT/ACT score or HSGPA for students in the sample with observed SAT/ACT scores or HSGPAs. The model also includes two dichotomous variables (one for SAT/ACT, and one for HSGPA) equal to one if the student was assigned the mean because the SAT/ACT score or HSGPA was missing and zero if SAT/ACT or HSGA were not missing.

4 A proportional hazard assumes a unit change in a predictor variable multiplies the probability of dropout by a constant amount in each time period.

5 The models also accounted for the clustering of students in the same section of a course and the possibility students in the same section might have shared risk of dropout by including random course effects (Hosmer & Lemeshow, 1999).

References

ACT (n.d.). *2018 ACT/SAT concordance tables*. www.act.org/content/dam/act/unsecured/documents/ACT-SAT-Concordance-Tables.pdf

Adelman, C., Daniel, B., & Berkovits, I. (2003). *Postsecondary attainment, attendance, curriculum, and performance: Selected results from the NELS: 88/2000 postsecondary education transcript study (PETS), 2000*. ED Tabs.

Alarcon, G. M., & Edwards, J. M. (2013). Ability and motivation: Assessing individual factors that contribute to university retention. *Journal of Educational Psychology*, *105*(1), 129.

Allen, L. K., Dascalu, M., McNamara, D. S., Crossley, S., & Trausan-Matu, S. (2016). Modeling individual differences among writers using ReaderBench. In *EDULearn16: 8th international conference on education and new learning technologies* (pp. 5269–5279). IATED Academy.

Allen, L. K., Snow, E. L., & Crossley, S. A., Jackson, G. T., & McNamara, D. S. (2014). Reading components and their relation to the writing process. *L'année psychologique/Topics in Cognitive Psychology*, *114*, 663–691. https://doi.org/10.4074/S0003503314004047

Attali, Y., & Burstein, J. (2006). Automated essay scoring with e-rater® V.2. *Journal of Technology, Learning, and Assessment*, *4*(3). https://ejournals.bc.edu/index.php/jtla/article/view/1650

Beigman Klebanov, B., Burstein, J., Harackiewicz, J. M., Priniski, S. J., & Mulholland, M. (2017). Reflective writing about the utility value of science as a tool for increasing STEM motivation and retention–can AI help scale up? *International Journal of Artificial Intelligence in Education*, *27*(4), 791.

Beigman Klebanov, B., & Flor, M. (2013). *Word association profiles and their use for automated scoring of essays* (Vol. 1, pp. 1148–1158). Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Long Papers).

Beigman Klebanov, B., Gyawali, B., & Song, Y. (2017). *Detecting good arguments in a non-topic-specific way: An oxymoron?* (Vol. 2, pp. 244–249). Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Short Papers) .

Beigman Klebanov, B., Leong, C. W., & Flor, M. (2015). *Supervised word-level metaphor detection: Experiments with concreteness and reweighting of examples* (pp. 11–20). Proceedings of the Third Workshop on Metaphor in NLP.

Beigman Klebanov, B., Leong, C. W., Gutierrez, E. D., Shutova, E., & Flor, M. (2016). *Semantic classifications for detection of verb metaphors* (Vol. 2, pp. 101–106). Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Short Papers).

Beigman Klebanov, B., & Madnani, N. (2020). *Automated evaluation of writing–50 years and counting* (pp. 7796–7810). Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.

Bunch, M. B., Vaughn, D., & Miel, S. (2016). Automated scoring in assessment systems. In Y. Rosen, S. Ferrara, & M. Mosharraf (Eds.), *Handbook of research on technology tools for real-world skill development* (pp. 611–626). IGI Global. https://doi.org/10.4018/978-1-4666-9441-5.ch023

Burstein, J., Beigman Klebanov, B., Elliot, N., & Molloy, H. (2016). *A left turn: Automated feedback & activity generation for student writers* (pp. 6–13). Proceedings of Language Teaching, Learning and Technology.

Burstein, J., Chodorow, M., & Leacock, C. (2004). Automated essay evaluation: The Criterion online writing service. *AI Magazine*, *25*(3), 27.

Burstein, J., Elliot, N., Klebanov, B. B., Madnani, N., Napolitano, D., Schwartz, M., Houghton, P., & Molloy, H. (2018). Writing mentor: Writing progress using self-regulated writing support. *Journal of Writing Analytics*, *2*, 285–313.

Burstein, J., Kukich, K., Wolff, S., Lu, C., Chodorow, M., Braden-Harder, L., & Harris, M. D. (1998, August). *Automated scoring using a hybrid feature identification technique* (Vol. 1, pp. 206–210). 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics.

Burstein, J., Marcu, D., & Knight, K. (2003). Finding the WRITE stuff: Automatic identification of discourse structure in student essays. *IEEE Intelligent Systems*, *18*(1), 32–39.

Burstein, J., McCaffrey, D., Beigman Klebanov, B., & Ling, G. (2017). *Exploring relationships between writing and broader outcomes with automated writing evaluation*. Proceeding of the 12th Workshop on Innovative Use of NLP for Building Educational Applications (BEA), EMNLP.

Burstein, J., McCaffrey, D., Beigman Klebanov, B., Ling, G., & Holtzman, S. (2019). *Exploring writing analytics and postsecondary success indicators*. Companion Proceedings 9th International Conference on Learning Analytics & Knowledge (LAK19).

Burstein, J., McCaffrey, D., Elliot, N., & Beigman Klebanov, B. (2020). *Exploring writing achievement and genre in postsecondary writing*. Companion Proceedings in the 10th International Conference on Learning Analytics & Knowledge (LAK20).

Burstein, J., Tetreault, J., & Madnani, N. (2013). The e-rater® automated essay scoring system. In *Handbook of automated essay evaluation* (pp. 77–89). Routledge.

Canning, E. A., Harackiewicz, J. M., Priniski, S. J., Hecht, C. A., Tibbetts, Y., & Hyde, J. S. (2018). Improving performance and retention in introductory biology with a utility-value intervention. *Journal of Educational Psychology*, *110*(6), 834–849.

Chimka, J. R., & Lowe, L. H. (2008). Interaction and survival analysis of graduation data. *Educational Research and Reviews*, *3*(1), 29–32.

Chimka, J. R., Reed-Rhoads, T., & Barker, K. (2007). Proportional hazards models of graduation. *Journal of College Student Retention: Research, Theory & Practice*, *9*(2), 221–232.

Cox, D. R., & Oakes, D. (1984). *Analysis of survival data*. Chapman and Hall, CRC Press.

Deane, P., Wilson, J., Zhang, M., Li, C. van Rijn, P., Guo, H., Roth, A., Winchester, E., & Richter, T. (2021). The sensitivity of a scenario-based assessment of written argumentation to school differences in curriculum and instruction. *International Journal of Artificial Intelligence in Education*, *31*, 57–98. https://doi.org/10.1007/s40593-020-00227-x

Flower, L. (1994). *The construction of negotiated meaning: A social cognitive theory of writing*. Southern Illinois University Press.

Foltz, P. W., Kintsch, W., & Landauer, T. K. (1998). The measurement of textual coherence with latent semantic analysis. *Discourse Processes*, *25*(2–3), 285–307.

Foltz, P. W., & Rosenstein, M. (2017). Data mining large-scale formative writing. *Handbook of Learning Analytics*, *199*.

Graesser, A. C., McNamara, D. S., Louwerse, M. M., & Cai, Z. (2004). Coh-Metrix: Analysis of text on cohesion and language. *Behavioral Research Methods, Instruments and Computers*, *36*(2), 193–202.

Graham, S. (2019). Changing how writing is taught. *Review of Research in Education*, *43*(1), 277–303. https://doi.org/10.3102%2F0091732X18821125

Harackiewicz, J. M., Canning, E. A., Tibbetts, Y., Priniski, S. J., & Hyde, J. S. (2016). Closing achievement gaps with a utility-value intervention: Disentangling race and social class. *Journal of Personality and Social Psychology*, *111*(5), 745.

Hayes, J. R. (2012). Modeling and remodeling writing. *Written Communication*, *29*, 369–388.

Hosmer, D. W., & Lemeshow, S. (1999). *Applied survival analysis: Regression modeling of time-to-event data*. Wiley-Interscience.

Hussar, B., Zhang, J., Hein, S., Wang, K., Roberts, A., Cui, J., & Dilig, R. (2020). *The condition of education 2020*. NCES.

Ishitani, T. T., & Snider, K. G. (2006). *Longitudinal effects of college preparation programs on college retention* (Vol. 9). IR Applications, Association for Institutional Research (NJ1).

John, O. P., & Srivastava, S. (1999). *The Big-Five trait taxonomy: History, measurement, and theoretical perspectives* (Vol. 2, pp. 102–138). University of California Press.

Kaplan, E. L., & Meier, P. (1958). Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, *53*(282), 457–481.

Leacock, C., Chodorow, M., Gamon, M., & Tetreault, J. (2010). Automated grammatical error detection for language learners. *Synthesis Lectures on Human Language Technologies*, *3*(1), 1–134.

Lederer, A. M., Hoban, M. T., Lipson, S. K., Zhou, S., & Eisenberg, D. (2020). More than inconvenienced: The unique needs of US college students during the CoViD-19 pandemic. *Health Education & Behavior*, *48*(1), 14–19.

Ling, G., Elliot, N., Burstein, J. C., McCaffrey, D. F., MacArthur, C. A., & Holtzman, S. (2021). Writing motivation: A validation study of self-judgment and performance. *Assessing Writing*, *48*, 1–15.

Madnani, N., Burstein, J., Sabatini, J., Biggers, K., & Andreyev, S. (2016). *Language muse: Automated linguistic activity generation for English language learners* (pp. 79–84). Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics – System Demonstrations, Berlin, Germany, August 7–12, 2016.

McCaffrey, D., Holtzman, S., Burstein, J., & Beigman Klebanov, B. (forthcoming). What can we *learn about college retention from student writing?* To appear in Companion Proceedings in the 11th International Conference on Learning Analytics & Knowledge (LAK21).

McNamara, D. S., Louwerse, M. M., McCarthy, P. M., & Graesser, A. C. (2010). Coh-Metrix: Capturing linguistic features of cohesion. *Discourse Processes*, *47*(4), 292–330.

Murtaugh, P. A., Burns, L. D., & Schuster, J. (1999). Predicting the retention of university students. *Research in Higher Education*, *40*, 355–371. https://doi.org/10.1023/A:1018755201899

National Student Clearinghouse Research Center (2020). *First-year persistence and retention*. https://nscresearchcenter.org/wp-content/uploads/PersistenceRetention2020.pdf

Ong, N., Litman, D., & Brusilovsky, A. (2014, June). *Ontology-based argument mining and automatic essay scoring* (pp. 24–28). Proceedings of the First Workshop on Argumentation Mining.

Page, E. B. (1966). The imminence of . . . grading essays by computer. *The Phi Delta Kappan*, *47*(5), 238–243.

Perin, D., & Lauterbach, M. (2018). Assessing text-based writing of low-skilled college students. *International Journal of Artificial Intelligence in Education*, *28*(1), 56–78.

Schneider, M. (2021, March 16). *The road to learning recovery*. Institute of Education Sciences Blog. https://ies.ed.gov/director/remarks/3-16-2021.asp

Shermis, M. D., & Burstein, J. C. (Eds.). (2003). *Automated essay scoring: A cross-disciplinary perspective*. Routledge.

Shermis, M. D., & Burstein, J. (Eds.). (2013). *Handbook of automated essay evaluation: Current applications and new directions*. Routledge.

Somasundaran, S., Burstein, J., & Chodorow, M. (2014, August 23–29). *Lexical chaining for measuring discourse coherence quality in test-taker essays*. The 25th International Conference on Computational Linguistics (COLING).

Stewart, S., Lim, D. H., & Kim, J. (2015). Factors influencing college persistence for first-time students. *Journal of Developmental Education*, *38*(3), 12–20. www.jstor.org/stable/24614019

Watson, D., Clark, L. A., & Carey, G. (1988). Positive and negative affectivity and their relation to anxiety and depressive disorders. *Journal of Abnormal Psychology*, *97*(3), 346.

White, E. M., Elliot, N., & Peckham, I. (2015). *Very like a whale: The assessment of writing programs*. University Press of Colorado.

Woods, B., Adamson, D., Miel, S., & Mayfield, E. (2017, August). *Formative essay feedback using predictive scoring models* (pp. 2071–2080). Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Contributor Biographies

Author Bios

Laura K. Allen is Assistant Professor of Educational Psychology at the University of Minnesota. The principal aim of her research has been to theoretically and empirically investigate the higher-level cognitive skills that are required for text comprehension and production, as well as the ways in which performance in these domains can be enhanced through strategy instruction and training.

***

Peter Baldwin is a principal measurement scientist in the Office of Research Strategy at NBME. He has been at NBME since 2007 and previously served in the role of senior measurement scientist. He has authored numerous research papers and book chapters in the areas of equating, standard setting, automatic item generation, and natural language processing.

Prior to joining NBME, Peter was a senior research fellow for the Research, Educational Measurement and Psychometrics program at the University of Massachusetts, Amherst, where he had previously earned his doctorate. The University of Massachusetts awarded him the Psychometric Fellowship in 2004. In 2015, Peter was awarded the Alicia Cascallar Award by the National Council on Measurement in Education.

***

Beata Beigman Klebanov is Principal Research Scientist in the Research and Development division at Educational Testing Service in Princeton, New Jersey. She received her PhD in computer science from the Hebrew University of Jerusalem, Israel, and her MS degree in cognitive science from the University of Edinburgh, UK. Before joining ETS, she was a postdoctoral fellow at the Northwestern Institute for Complex Systems and Kellogg School of Management, where she applied computational approaches to the study of political rhetoric. Since joining ETS in 2011, she has worked extensively on automated assessment of various aspects of writing, including genre-specific vocabulary choices, figurative language, sentiment, argumentation, theme development, incorporation of external sources into own writing. She has also worked on developing language technologies to support learning and practice of literacy skills.

***

Jared C. Bernstein is Senior Staff Research Scientist at Google LLC and Adjunct Professor of Linguistics at Stanford University. He holds a PhD in psycholinguistics from the University of Michigan. Before joining Google, he co-founded of Analytic Measures Inc. (AMI) and Ordinate Corporation. For 30 years, Bernstein has worked on automatic spoken language processing for atypical populations such as nonnative speakers, young children, and dysarthric or hearing-impaired adults. He designed and led the development of the Versant tests in English, Spanish, Dutch, Arabic, and Chinese, as well as the prototype for a tablet-based language proficiency test for K-12 English learners (TELL). In 2014–1017, he led the design and development of NAEP's 2018 ORF Special Study.

***

Jill Burstein, PhD, is Principal Assessment Scientist at Duolingo and conducts assessment innovation research for the Duolingo English Test. Her career has been motivated by social impact, working on equitable and widely accessible education technology. Her research interests lie at the intersection of artificial intelligence (AI) and natural language processing (NLP), educational measurement, equity in education, learning analytics, and linguistics. As a leader in AI in education, Jill has led NLP teams that invented automated writing evaluation (AWE) systems used in large-scale assessment, and digital writing support applications. She has published widely and holds many patents for this work. Jill has led federally funded writing analytics research that examined relationships between AWE writing achievement measures and broader academic postsecondary outcomes. Jill is a co-founder of SIG EDU, an ACL Special Interest Group on Building Educational Applications. Jill holds a BA in Linguistics and Spanish from New York University, and MA and PhD degrees in linguistics from the Graduate Center, City University of New York.

***

Aoife Cahill is a natural language processing (NLP) expert and a director of AI research at Dataminr, the leading real-time information discovery platform. Since joining in 2021, Aoife has led a team of data scientists focused on the efficient iterative process of developing and evaluating AI technology that supports the expansion of Dataminr's internal and external products. Prior to Dataminr, Aoife led a team of research scientists and engineers working on high-stakes NLP applications in the educational domain at the Educational Testing Service (ETS). The NLP teams at ETS are known leaders in the field of developing and deploying robust, well-documented, scalable NLP prototypes that maintain fairness across user groups.

Aoife holds a PhD in computational linguistics from Dublin City University, Ireland, and has also spent time conducting NLP research in Germany, Norway, and the United States. As an active member of the computational linguistics research community, her research has been published in top-tier journals including *Computational Linguistics* and the *Journal of Research on Language and Computation*, as well as conference proceedings at the annual conference for the Association for Computational Linguistics (ACL), the International Conference on Computational Linguistics (COLING), and the Conference on Empirical Methods in Natural Language Processing (EMNLP).

***

Jian Cheng is Senior Software Engineer at Google LLC. He has a PhD in artificial intelligence from the University of Pittsburgh and earlier degrees from the Chinese Academy of Sciences and Tsinghua University. Before joining Google, he was co-founder, CEO, and CTO of Analytic Measures Inc. (AMI). Cheng's work focuses on development of machine learning, language processing, and automated scoring algorithms for assessment. Prior to AMI, he was chief scientist and director of research services at Pearson's Knowledge Technologies unit, where he developed the main automated-scoring production system behind Pearson's Versant tests, the Pearson Test of English Academic, Arizona's AZELLA tests, and Pearson's tablet-based English proficiency assessment, TELL.

***

Brian E. Clauser is the distinguished research scientist at NBME. He has published more than 100 journal papers and book chapters on issues related to differential item functioning, performance assessment, automated scoring of complex assessments, standard setting, applications of generalizability theory, test validity, and the history of educational measurement. He is a recipient of NCME Career Contribution award, a fellow of the American Educational Research Association, and past editor of both the *Journal of Educational Measurement* and the National Council on Measurement in Education (NCME) book series.

***

Ying Fang is Associate Professor of Faculty of Artificial Intelligence in Education at Central China Normal University. Her research interests include artificial intelligence in educational systems, the development, implementation, and assessment of intelligent tutoring systems, and promoting learning in electronic learning environments.

***

Le An Ha is Senior Lecturer at the Research Group in Computational Linguistics, Research Institute in Information and Language Processing, University of Wolverhampton, United Kingdom. His primary research interests include multiple-choice question generation and processing, and machine learning, including deep learning for a wide variety of NLP applications. He has worked on text categorization, short answer scoring, note scoring, response time prediction, and detection of autism, among others. He delivers specialized seminars and supervises PhD students on statistics, machine learning, and deep learning.

***

Minsu Ha is Professor in the Division of Science Education at Kangwon National University in Chuncheonsi, Republic of Korea. He holds a doctoral degree in science education from The Ohio State University in the United States. His research focuses on an assessment of biological concepts and the development of artificial intelligence (AI)–based assessment systems for open-ended assessment.

***

Polina Harik is a managing senior measurement scientist at NBME, specializing in automated scoring of complex performance tasks. Her scholarly contributions have focused on the area of applied psychometrics, generalizability analyses, and natural language processing (NLP) to evaluate and improve the quality of large-scale, high-stakes standardized examinations. Her current research interests lie at the intersection of NLP and automated scoring.

***

Steven Holtzman serves as Principal Research Data Analyst at Educational Testing Service. He has an MA in Statistics and a BA in statistics and economics from Boston University. His work at ETS concentrates on using study design, data collection, data management, and data analysis methods to help promote research in education. He has also co-authored numerous publications and presented at many conferences. Recent projects have examined noncognitive assessments, writing skills, teacher evaluation, and workforce selection assessments.

***

Amir Jafari is Lead Data Scientist at Cambium Assessment. He has a PhD in electrical and computer engineering from Oklahoma State University. He has been conducting research in the areas of neural network design, statistical modeling, artificial intelligence, and predictive modeling for the last six years. His research has encompassed a variety of application areas: machine learning, deep learning, artificial intelligence, natural language processing, and optimization.

***

Matthew S. Johnson (Matt) is Principal Research Director of the Foundational Psychometrics and Statistics Research Center at Educational Testing Service (ETS). He returned to ETS in 2018 after previously working there from 2000 to 2002, when he was Associate Research Scientist working on NAEP. Prior to rejoining ETS, he was most recently Associate Professor of Statistics and Education and Chair of the Department of Human Development at Teachers College of Columbia University in the Department of Human Development. His research focuses on the use of statistical methods in education and psychology, with a primary focus on item response theory and related models. He has served as co-editor of the *Journal of Educational and Behavioral Statistics*, served on the Editorial Board of *Psychometrika*, was program co-chair for the National Council of Measurement in Education annual conference, and served as treasurer of the Psychometric Society.

***

Geoff LaFlair is Lead Assessment Scientist at Duolingo, where his primary responsibilities include research and development of the Duolingo English Test. Prior to joining Duolingo, he was Assistant Professor in the Department of Second Language Studies at the University of Hawai'i at Mānoa and Director of Assessment in the Center for English as a Second Language at the University of Kentucky. He earned his PhD in applied linguistics, specializing in language assessment, corpus linguistics, and quantitative research methods, from Northern Arizona University. His research interests are situated at the intersection of applied linguistics, psychometrics, and natural language processing. It has been published in *Language Testing*, *Applied Linguistics*, the *Modern Language Journal*, the *Transactions of the Association for Computational Linguistics*, the *Journal of Computer Assisted Learning*, *Frontiers in Artificial Intelligence*, and *Empirical Methods in Natural Language Processing*.

***

Suzanne Lane is Professor Emeritus in the Research Methodology Program at the University of Pittsburgh. Her research and professional interests are in educational measurement and testing, with a focus on design, test validity, and equity issues pertaining to testing and on the effectiveness of education and accountability programs. Her work is published in journals such as the *Journal of Educational Measurement*, *Applied Measurement in Education*, *Educational Assessment*, and *Educational Measurement: Issues and Practice*. She has served on the Editorial Boards for the *Journal of Educational Measurement*, *Applied Measurement in Education*, *Educational Assessment*, *Educational Researcher*, and *Educational Measurement: Issues and Practice*. She is the first author of the Validity chapter in the upcoming *Educational Measurement* edited text. She was President of NCME (2003–2004), Vice President of Division D of AERA (2000–2002), AERA Fellow, member of the AERA, APA, and NCME Joint Committee for the Revision of the Standards for Educational and Psychological Testing (1993–1999), and member of the Management Committee for the next revision of the Standards (2006–2015). She was appointed to the National Assessment Governing Board that sets policy for NAEP (2020–2024). She has also served on technical advisory committees for the College Board, ETS, PARCC, U.S. Department of Education's Evaluation of NAEP, U.S. Department of Education Race to the Top Technical Review, National Research Council, and NCEO as well as for state assessment and accountability programs (CO, DE, GA, KY, NJ, NY, OK, PA, SC, TN, TX).

***

Susan Lottridge is Chief Scientist at Cambium Assessment, Inc. She has a PhD in assessment and measurement from James Madison University and master's degrees in mathematics and computer science from the University of Wisconsin–Madison. In this role, she leads CAI's machine learning and scoring team on the research, development, and operation of CAI's automated scoring software. She has worked in automated scoring for 15 years and has contributed to the design, research, and use of multiple automated scoring engines including equation scoring, essay scoring, short answer scoring, alert detection, and dialogue systems.

***

Anastassia Loukina is Engineering Manager at Grammarly, Inc. She leads a cross-functional team responsible for monitoring the quality of Grammarly AI systems. Before joining Grammarly, Anastassia was a senior research scientist in the Research and Development division at Educational Testing Service (ETS) in Princeton, New Jersey, where she worked on improving the validity, reliability, and fairness of speech-based educational applications. She published over 60 papers and book chapters, holds several patents, and frequently attends international conferences and workshops. She holds MPhil and DPhil degrees from the University of Oxford.

***

Nitin Madnani is Distinguished Research Engineer in the AI Research Labs at the Educational Testing Service (ETS) in Princeton. His NLP adventures began with an elective course on computational linguistics he took while studying computer architecture at the University of Maryland, College Park. As a PhD student at the Institute of Advanced Computer Studies (UMIACS), he worked on automated document summarization, statistical machine translation, and paraphrase generation. After earning his PhD in 2010, he joined the NLP and Speech research group at ETS, where he leads a wide variety of projects that use NLP to build useful educational applications and technologies. Examples include mining Wikipedia revision history to correct grammatical errors, automatically detecting organizational elements in argumentative discourse, creating a service-based, polyglot framework for implementing robust, high-performance automated scoring and feedback systems, and building the first-ever, fully open-source, comprehensive evaluation toolkit for automated scoring.

Dr. Madnani is co-author of *Automated Essay Scoring*, a comprehensive book published by Morgan Claypool as part of their selective Synthesis Lectures on Human Language Technologies Series. His research has appeared in leading journals (*Computational Linguistics*, *Transactions of the ACL*, *ACM Transactions on Speech and Language Processing*, *ACM Transactions on Intelligent Systems and Technology*, *Machine Translation*, *Journal of Writing Analytics*, and the *Journal of Open Source Software*) as well as in the proceedings of top-tier conferences such as Association for Computational Linguistics' annual meeting series (ACL, NAACL, EACL, EMNLP), International Conference on Computational Linguistics (COLING), Learning Analytics and Knowledge, Learning @ Scale, and the annual meetings of the American Educational Research Association (AERA) and the National Council on Measurement in Education (NCME). He currently serves as an action editor for *Transactions of the ACL* (TACL), as an executive board member of the ACL Special Interest Group on Building Educational Applications (SIGEDU), and as Chief Information Officer for ACL. He has previously served as senior area chair, area chair, or a member of the organizing committee for the NAACL/ACL/EMNLP series of conferences since 2017.

***

Halyna Maslak is a data engineer at BT, UK. She obtained her MA in technologies for translation and interpreting at the University of Wolverhampton (UK) and New Bulgarian University

(Bulgaria). She also studied English language and literature at Chernivtsi National University in Ukraine. Her research interests include NLP for educational purposes, corpus linguistics, and machine translation for low-resource languages. She is passionate about data and data-driven decisions for the business.

***

Daniel F. McCaffrey (Dan) is Associate Vice President of Psychometric Analysis and Research in the Research and Measurement Sciences unit in the Research and Development division at ETS. Prior to joint ETS, Dan was the PNC Chair in Policy Analysis at the Rand Corporation. He received his PhD degree in statistics from North Carolina State University in 1991 and his BA in mathematics and economics from Mount Saint Mary's College in 1986. His research interests include methods for causal modeling, measuring student growth and teacher and school value added, and the evaluation and use of automated scores of constructed responses. He is a fellow of the American Statistical Association and was co-editor of the *Journal of Educational and Behavioral Statistics* from 2015 to 2019.

***

Danielle S. McNamara, PhD, is Professor of Psychology at Arizona State University. She is an international expert in the fields of cognitive and learning sciences, learning engineering, reading comprehension, writing, text and learning analytics, natural language processing, computational linguistics, and intelligent tutoring systems. Her research involves the development and assessment of natural language processing tools (e.g., Coh-Metrix) and game-based intelligent tutoring systems (e.g., iSTART, Writing Pal; see soletlab.asu.edu).

***

Janet Mee is a measurement scientist at NBME with nearly 20 years of experience in research and assessment innovation in medical education. Her scholarship has focused on issues related to standard setting, practice and survey analyses, and generalizability studies. Her current interests include data science, as well as development of automated systems for scoring clinical text using natural language processing.

***

Ruslan Mitkov is Professor of Computational Linguistics and Language Engineering at the University of Wolverhampton which he joined in 1995 and where he set up the Research Group in Computational Linguistics, members of which have won awards at different NLP/shared-task competitions and conferences. In addition to being Head of the Research Group in Computational Linguistics, Prof Mitkov is also Director of the Research Institute in Information and Language Processing and Director of the Responsible Digital Humanities Lab. Dr. Mitkov is Vice President of ASLING, an international association for promoting language technology. He is a Fellow of the Alexander von Humboldt Foundation, Germany, was a Marie Curie Fellow, Distinguished Visiting Professor at the University of Franche-Comté in Besançon, France and Distinguished Visiting Researcher at the University of Malaga, Spain. In recognition of his outstanding professional/research achievements, In October 2022 Dr. Mitkov was awarded the title 'Doctor Honoris Causa' by New Bulgarian University, Sofia, the third time he has been so honored (Plovdiv University, Veliko Tarnovo University).

***

Ross H. Nehm is PI of the Biology Education Research Lab and Professor in the Department of Ecology and Evolution and the Program in Science Education at Stony Brook University (State University of New York). His lab was an early pioneer in the use of AI in studies of biology learning and assessment, and it continues to advance understanding of its potential to improve learning outcomes in undergraduate settings. Dr. Nehm completed his graduate work in biology and science education at the University of California–Berkeley and Columbia University. His major awards include an NSF CAREER award, a student mentoring award from CUNY, and a teaching award from Berkeley. He was named an Education Fellow in the Life Sciences by the U.S. National Academies and has served in academic leadership roles including as Editor-in-Chief of the journal *Evolution: Evolution Education and Outreach*, Associate Editor of *Science & Education*, Associate Editor of the *Journal of Research in Science Teaching*, Editor of *CBE-Life Sciences Education*, and a board member of several other journals. He has served on the research advisory boards of numerous federally funded science education projects, the National Science Foundation's Committee of Visitors, and many NSF panels as Chair.

***

Chris Ormerod is Principal Mathematician at Cambium Assessment. He has a PhD in applied mathematics from Sydney University in Australia and has worked in many areas including mathematical physics, systems biology, and natural language processing. His main research interests are in the application of machine learning and natural language processing to automated assessment.

***

Tharindu Ranasinghe is a lecturer at the University of Wolverhampton and a member of the Research Group on Computational Linguistics (RGCL), affiliated with the Research Institute of Information and Language Processing (RIILP). He holds a PhD in computer science from the University of Wolverhampton, which he defended in 2021. As a PhD student, he worked on applying deep learning–based text similarity models for applications in translation technology under the supervision of Ruslan Mitkov and Constantin Orasan. He serves as a program committee member of multiple conferences. He is also a co-organizer of the shared task on Hate Speech and Offensive Content Identification in English and Indo-Aryan Languages (HASOC) from 2021. His research focuses on various aspects of machine learning–driven approaches to natural language processing, with a particular interest in multilingual models and explainable machine learning. His work has diverse applications such as translation quality estimation, social media data mining, offensive language identification, information extraction, and digital humanities.

***

Rod D. Roscoe is Associate Professor of Human Systems Engineering in the Ira A. Fulton Schools of Engineering at Arizona State University. His research combines insights from learning science, computer science, and design science to improve the implementation and effectiveness of equitable educational technologies.

***

Christopher Runyon is a senior measurement scientist at NBME. His current primary research focus is the assessment of clinical reasoning. His expertise includes building psychometric applications, statistical methods in measurement, and designing automated scoring frameworks that utilize natural language processing and machine learning. His background is in cognitive science and philosophy (logic and reasoning).

***

Burr Settles is a Technical Advisory Board member for the Duolingo English Test. Previously, he was Research Director at Duolingo, where he created the Duolingo English Test. Burr has a PhD in machine learning and computational linguistics from the University of Wisconsin–Madison. He is the author of *Active Learning* (Morgan & Claypool, 2012) and a former special faculty at Carnegie Mellon University.

***

Vilelmini Sosoni is Assistant Professor at the Department of Foreign Languages, Translation and Interpreting at the Ionian University in Greece. She has taught specialized translation in the United Kingdom at the University of Surrey, the University of Westminster, Roehampton University, and the University of Wolverhampton, and in Greece at the National and Kapodistrian University of Athens, Metropolitan College, and the Institut Français d'Athènes. She also has extensive professional experience having worked as a professional translator, editor, and subtitler. She studied English language and literature at the National and Kapodistrian University of Athens and holds an MA in translation and a PhD in translation and text linguistics from the University of Surrey. Her research interests lie in the areas of the translation of institutional and political texts, corpus linguistics, audiovisual translation and accessibility, as well as machine translation and MTPE. She is the vice-president of the Hellenic Society for Translation Studies and a founding member of the Laboratory 'Language and Politics' of the Ionian University and the Greek Chapter of Women in Localization. She is also a member of the Advisory Board and the Management Board of the European Master's in Technology for Translation and Interpreting (EM TTI) funded by Erasmus+. She has participated in several EU-funded projects, notably Resonant, Trumpet, TraMOOC, Eurolect Observatory and Training Action for Legal Practitioners: Linguistic Skills and Translation in EU Competition Law, while she has edited several volumes and books on translation and published numerous articles in international journals and collective volumes.

***

Alina A von Davier, PhD, is Chief of Assessment at Duolingo and CEO and Founder of EdAstra Tech. She is Honorary Research Fellow at Oxford University and at Carnegie Mellon University. Von Davier is a researcher, innovator, and an executive leader with over 20 years of experience in EdTech and in the assessment industry. She and her team operate at the forefront of computational psychometrics. Her current research interests involve developing psychometric methodologies in support of digital-first assessments, such as the Duolingo English Test. She was awarded several prizes for her books and edited volumes in psychometrics.

***

Matthias von Davier is the J. Donald Monan, S.J., Professor in Education at the Lynch School of Education at Boston College (BC) and serves as the TIMSS & PIRLS International Study Center's executive director. His research areas include item response theory, invariance and linking, diagnostic classification and mixture models, machine and deep learning, computational statistics, model fit, and methodologies used in large-scale educational surveys.

***

Kevin Yancey is a staff AI research engineer at Duolingo and Co-Lead of the Test Scoring Team on the Duolingo English Test. He has an MEng in natural language processing from Waseda University. He specializes in the applications of natural language processing and item response theory towards second language acquisition and assessment.

***

Victoria Yaneva is a senior NLP scientist at NBME and an honorary research fellow at the University of Wolverhampton. Her interests lie in the various intersections between natural language processing (NLP) and educational measurement, with an emphasis on developing applications for high-stakes clinical exams. Examples include NLP research on predicting item characteristics from item text, automated scoring, and automated distractor generation. Another area of interest is the use of eye-tracking methodology in process validity research for clinical multiple-choice questions. She completed her doctorate in NLP at the Research Group in Computational Linguistics at the University of Wolverhampton in 2017, which was followed by a postdoctoral appointment in the same research group prior to joining NBME in 2018.

***

Index

Numbers in **bold** indicate a table. Numbers in *italics* indicate a figure on the corresponding page.

AAVE (African American Vernacular English) 45
ACORNS instrument (Assessing COntextual Reasoning about Natural Selection) 203–206, 201–213
ACT/SAT 220, 223, 224, 226; *see also* GRE; SAT
Afzal, N., and Mitkov, R. 78
age of acquisition 168
AIG *see* automated item generation
'all-MiniLM-L6-v2' model 82
all-talker model 34
Alsubait, T. 78
ANCOVA (analysis of covariance) 153
Andrew, J. J. 134
ANOVA 145
ARPABET 38
artificial intelligence (AI): AI-based AIG trained on workstations with gaming GPU 96–98; AIG and 136–137; 'AI Winter' 91; deep fakes, use of AI to detect 103; NLP approach and 137; Open AI 90, 92, 94, 95, 103; Vast.AI 97, 98
AS *see* automated scores
ASAP *see* Automated Student Assessment Prize (ASAP) dataset
ASE *see* automated scoring engines
ASR *see* automatic speech recognition
ASV *see* automatic speaker verification
Atlassian Bitbucket 8
Attali, Y., and Burstein, J. 143
audio yes/no 110, **110**, *111*, 117
automated item generation (AIG) 116, 136–137; AI and 93–105, 136–137; deep neural networks and 90; Duolingo English Test and 109–118; ML and psychometrics for 107–121; NLP and 109, 136–137; state of 90
automated scores (AS) 142
automated scoring: AI, fair scores in assessment 146–148; AI scores, proposed methods for evaluating 148–156; AI, use in 139, 141, 142; algorithms 130–131, 133; computational psychometrics and 108; deep learning in 15–26; definition of 15; Duolingo English Test 118–121; in educational measurement, evaluating fairness in 142–162; fairness in, evaluating 146, 156, 160; fairness in, existing methods for evaluating 148–151; fairness as validity concern 18, 141–142; high level flow *7*; INCITE system 63–71; models 17–26; NLP scoring system for patient notes 58–71; NLP, use in 139, 142; pipeline *7*; process 16–17; psychometric challenges with using deep learning models 21–26; role of robust software in 3–13; speech analysis and 31–55; of text/written responses, history of 58–63; unfair AI scores, remedies for 155; unfairness in testing, methods for evaluating 145–146; unfair scores, thresholds for flagging 154; validity and fairness in 127–139; *see also* ASE; AWE
automated scoring engines (ASE) 15, 16; IUA and validity arguments for 131–134
automated second language tests, validation data for **54**
Automated Student Assessment Prize (ASAP) dataset 20
automated writing evaluation (AWE) 217–232; AI and 232; automated essay scoring, roots in 221; college retention and contributing factors 218–221; composite AWE features **229**, 229–230; definition of 221; feature descriptions **226**–**227**, **229**; HEIghten assessment 229; predicting dropout 226–228, **228**; study and methods (data; participants) of 223–226; survival model results **229**, 231; Turnitin 221, 231
automatic speaker verification (ASV) 34
automatic speech recognition (ASR): 2022 applications of 35, 40–55; ASR-based scoring in a test *39*, 45; ASR-produced word sequences 33; ASR system, schematic view of the operation *36*; definition of 35; engine 120; evaluation

38–40; Microsoft ASR system 40; Moby.Read 40, 41–45; NAAL 45; operation and development 36–38; orthographic 35; phoneme node 37, *38*; second language speaking ability, assessment of 45–53; SLP and 35; SLR and 35; software 8; state node 37, *38*; transcription 8, 130; transparency and bias 53–55; Versant English Test 45; Versant Spanish Test 45–53; WER in 35, 38–40
automatic transcription 8
AWE *see* automated writing evaluation
AZELLA test, Arizona 46

Bai, Z., and Zhang, X. 34
Baldwin, P. 138, 175, 179
Balogh, J. 45, 55
Bennett, R., and Zhang, M. 131
Bernstein, Jared, et al. 40, 53; *see also* speech analysis in assessment
BERT *see* Bidirectional Elementary Representation by Transformers (BERT) model
Bidirectional Elementary Representation by Transformers (BERT) model 19, 20, 23–25, **26**, 104, 118, 172, 180n19; BERT Base 172, 175; BERT Large 172, 175; BERTScore 87; differences in exact agreement and QWK between Classical and Hybrid (Classical + BERT) engines on operational responses (Grades 3–8, *n* ≈ 500 per Item, 12 Items) **26**; differences in exact agreement and QWK between Classical and Hybrid (Classical + BERT) engines on teacher-scored responses (Grades 3, 5, 7, and 9, *n* ≈ 75 per Item, 11 Items) **25**; DISTILBert 173; QWK performance on two essays for Classical, LSTM, and Transformer (BERT)-based models across increasing training sample sizes *20*; ROBERTa and 173, 175, 178, 180n19; SBERT 82, **83**, 84; two tasks and two embeddings of 172
BLEURT 87
BLOOM 90, 91
BooksCorpus 175
Burstein, J. 108, 143

Call Home corpus 47
Camacho-Collados, J. 168
Canvas system 231
CAT *see* computer-adaptive test (CAT) phase
Cheng, Jiang 40, 41; *see also* speech analysis in assessment
Chimka, J. R., and Lowe, L. H. 219
Clauser, B. E. 135
COCA *see* Corpus of Contemporary American English
Cohen's kappa **22**, 205
Coh-Metrix 188, 190, 191, **192**, 194–196, 222, 240
Cole, B.S. 139
computer-adaptive test (CAT) phase 109–110
computational psychometrics: as integrative framework 108–109
Constructed Response Analysis Tool (CRAT) 188, 196
Corpus of Contemporary American English (COCA) 191
Cox proportion hazards regression 219, 226
CRAT *see* Constructed Response Analysis Tool
c-test **110**, 111, *112*

Das, R. K. 34
data generation 169
deep learning (DL): electronic health records and 98; models for NLP 78, 79, 80, 82, 87; psychometric challenges 21–26; psychometric challenges when using 15–26l
deep neural networks (DNNs) 90; *see also* NNs
Defense Language Institute (DLI), Monterey, California 50, 52, **53**
dense word vector representations 169; *see also* embeddings
dependencies 3–4, 8–9, 178; arbitrarily long 24; encoding 168; parameter 18
dependency-based approach 78
dependency parse 35
dependency relationship, structural 184
dependency tree model 78
Devlin, J. 118, 172; *see also* BERT
DIF *see* differential item functioning
differential item functioning (DIF) 145, 214; Mantel-Haenszel DIF procedure (MHDIF) 145–146
digital-first assessment: definition of 107, 108; machine learning and 107–121
distractors 77, 78, 80, 87, 99, 100, 137; correct-answer 77; deep neural networks used to generate 92; MCQ 78; named entity 78; plausible 138
distractor selection: feature-based and neural net (NN)–based ranking models used in 78
distributional hypothesis 169
DL *see* deep learning
DNNs *see* deep neural networks
doc2vec 82–83, **83**
Drori, I. 105
Duolingo English Test 108, 109–121; automated item generation 109–118; automated scoring 118–121; definition of 109; fairness and commonality of use of 142; item types 110, **110**; performance tasks 119; Zierky's recommendations and guidelines for review of 118

EASE *see* Enhanced AI Scoring Engine
ECD *see* evidence-centered design
educational testing services (ETS) 11–12, 18, 142, 150; *Best Practices for Constructed-Response*

Scoring 154; Criterion online essay evaluatioin system 221; e-rater system 143
EHRs *see* electronic health records
electronic health records (EHRs) 92, 96, 98
elicited speech **110**, 111, *112*, 117
ELMo 19, 172, 173, 180n12
embeddings 18–19, 24, 81–82; contextualized 168, 172, 173, 175, 176, 178–179; language model 118; medical text 172; non-contextualized 168, 175, 176, 180; position 138; sentence 172, 177, 178; token 172, 177; word 169–171, 180
Enhanced AI Scoring Engine (EASE) 20–21
ENRIE-GEN 79
entropy: ASR language model 41; cross 95–96; Kullback-Leibler divergence entropy metric 17; logarithmic 95; lowest entropy LM, development of 37, 39, *39*
e-rater system 143
evidence-centered design (ECD) 130, 134, 135

fairness: validity, technology-based assessment, and 127–139; automated scoring in educational measurement, evaluating 142–162; in testing 143–144
fake-a-talker 34
"fake collaboration" pattern 134
fake news 95, 96
fakes, use of AI to detect 103
FAN *see* Fluency Addition to NAAL
Firth, J. R. 168
Fluency Addition to NAAL (FAN) 45
frames 36, 38, 46
Franco, H. 40
functionality and evaluation metrics 3
functional test 7, 12
fuzzy similarity and matching 64–66, **68**, 70–71, 79–81

game-based: assessment 135; learning 93, 135
game the system 11, 61, 72
games: NLP and, in iStart 183–196
gaming behavior 26
gaming company, education 93
gaming GPU 100; AI-based AIG trained on workstations with 96–98
gaming responses 17; strategies 133
gaming tasks 185
Gao, Y. 78, 138
Gates-MacGinitie Reading Test 189, 191, 192, 193
Gates-MacGinitie Vocabulary 222
gauGAN NVIDIA 103
Gaussian mixture models 34, 38
generalized linear mixed model (GLMM) 118
GitHub 8, 95
GLMM *see* generalized linear mixed model
GLUE benchmarks 19, 23
GMAT 108, 142, 221
Google Translate (GT) 204, **206**, 206–207, **208**, **209**, **210**, 211–212
GPA *see* grade point average
GPT2 172; case study of automated item generation using AI 19, 90–105
GPT3 103, 121, 172; vignette written by **104**
GPT-neoX 90, 91
GPU *see* graphics or graphical processing unit
grade point average (GPA) 218–220, 222–224; HSGPA 219, 222, 226, 227, 229; university 228
Grammarly 80
graphics or graphical processing unit (GPU) 24, 91, 99, 100; AI-based AIG trained on workstations with gaming GPU 96–98
GRE 108, 142, 221
Grover 95

H1 (H1H2, H2, etc.) 21, **26**, 38, 147, 152, 153, **206**, 207, **208**, **209**
Ha, L. A., and Yaneva, V. 138
Ha, M. 203, 204
Hansen, J. H., and Hasan, T. 34
hierarchical encoder-decoder framework 78
high-level design of a neural network–based automated scoring engine *18*, 21
high performance computing (HPC) 91
Holmlund, T. 40
HPC *see* high performance computing
human annotator **22**, 22; of student readings 45
human baseline **20**
human collaborator: virtual agent or 136
human creativity 103
human editor 80
human-engineered linguistic features 168–169, 171, 175, 178
human experts 5, 9, 92, 100, 104, 105, 116
human graders 119, 120
human judgement: SBERT outperforming 82
human listeners: judgement of 33, 34, 36
human performance: automated scoring predicting 132; deep learning exceeding 21
human raters 11, 38, 144–151, 153–155, 157–158; ASE ratings compared to 133; human-human agreement 25; INCITE 63, 69; NLP to replace, possibility of 58–59, 61; semantic analysis applied to 60; of Spanish test 49, 50, 51, 52, *52*, 53
human rating true score 142
human review 117, 121
human scores, human scoring 15, 16, 18, 55; AI input 143; compared with ASR test 48; compared to Moby.Read *43*, 44; human concurrent 46; IUA and validity arguments for 131; modeling 61–62
human score scale 149

human-tagged text 202
human test scores: validation data for automated second language tests **54**
human transcription 39, 40; compared to machine transcription 40
human translation: machine translation and 204–214
human true scores 161, 162
human written items 91, 99, 100
Hutchinson, B., and Mitchell, M. 145
hyperparameter tuning 25, 100, 173, 175, 178
hypertension **101**

independent reading 41
independent writing/speaking **110**, 113, *115*, 119
INCITE 60–61, 63–71, 72n9
inference 32, 184, 186, 200; computations 82; extrapolation and decision/use 133–134, 135–137; fair 161; generalization 131, 133, 135–137, 138–139; IUA 128–129; linguistic, cultural, and substantive (LCS) patterns 130; performance 135, 137–138; scoring 132–133, 135; skill and deficit 44; stage 82
Institute for Education Science 231
integration test 7
integrative tasks 111
interpretation and use arguments (IUA) 128–139
IR: complete the paragraph **110**; complete the sentence **110**; highlight the answer **110**; identify the idea **110**
Irum, A., and Salman, A. 34
iSTART: NLP in 183–197
Ishitani, T. T., and Snider, K. G. 219, 220
item development: artificial intelligence used in 90–105; automatic generation of multiple choice test items using DNN 77–87; ML and psychometrics in 107–121
IUA *see* interpretation and use arguments

Johnson, M. S. et al. 161
Jurafsky, D., and Martin, J. 33, 36, 39

Kaggle **20**, 23
Kaplan-Meier analysis 219
kappa 120, **206**, 207, **208**, **209**, **210**, 211, 212; Cohen's **22**, 205; *see also* quadratic weighed kappa (QWK)
KERAS 98
knowledge, skills, and other attributes (KSAs) 128–131, 134–138
known-talker model 34
Koenecke, A. 55
Kolen, M. 133
KSAs *see* knowledge, skills, and other attributes
Kullback-Leibler divergence entropy metric 17
Kurdi, G. 138

Langenfeld, T. 108
Language Model (LM) 37–39; factors affecting speech model accuracy *39*
LASSO 143, 155, 157–160; regression 157, 159, *160*
latent semantic analysis 16, 20, 185, 186
LCS patterns see linguistic, cultural, and substantive (LCS) patterns
lexemes 47
lexicon 36, 37, 46, 47
linguistic, cultural, and substantive (LCS) patterns 130
linguistic expressions 203
linguistic features 184, 188; descriptive statistics of five selected features **192**; human engineered 168–169, 175; multiple linear regression analysis predicting reading test scores with linguistic features across datasets **193**, 194, **195**; of student self-explanations 189–191, 196
linguistic groups 133
linguistic minorities 214
linguistic patterns 135
linguistic signal: extracting from item text and its application 167–181
linguistic variables 222
literary assessment *see* stealth literary assessment
logarithmic entropy 95
long-short-term-memory (LSTM) 19, **20**, 20, 23, 24, 91
Lottridge, S., and Young, M. 18
LSA *see* latent semantic analysis
LSTM *see* long-short-term-memory

machine-human agreement (MHA) 203
machine-learning models 78; feature-based and neural net (NN)–based ranking models 78
machine scoring: human translation, machine translation and 204–214
machine translation: human translation, machine scoring, and 204–214; NLP and 200–214
MadLibs 91, 92
Mantel-Haenszel DIF procedure (MHDIF) 145–146
Mao, L. 134
McCarthy, K. S. 188
McDonald, S., and Ramscar, M. 169
McNamara, D. S. 188
MCQ *see* multiple-choice questions
MCTIC *see* multiple-choice test items corpus
MEDLINE 78, 172, 175
Megatron LM model 90, 95
MHA *see* machine-human agreement
MHDIF *see* Mantel-Haenszel DIF procedure
Mitkov, Ruslan, and Ha, Le An 78, 80; *see also* multiple choice test items, automatic generation of

Moby.Read 41–44
modeling human scores *see* human scores, modeling
model training 12, 169; *see also* embeddings
multiple-choice tests, generation of 78
multiple choice test items, automatic generation of 77–87; NLP methodology for 77
multiple-choice questions (MCQs) 77, 78, 81, 168; automatic generation of 78; clinical 171, 173, 178, 180; linguistic features extracted from 179; math and IT 173; pretrest 174
multiple-choice test items corpus (MCTIC) 79–80, 81–84

NAAL *see* National Assessment of Adult Literacy
NAEP *see* National Assessment of Education Progress
named entity distractors 78; *see also* distractors
named entity recognition 19, 137
National Assessment of Adult Literacy (NAAL) 45
National Assessment of Education Progress (NAEP) 45, 142, 149
National Center for Education Statistics (NCES) 45, 218
National Institute of Child Health and Human Development (NICHD) 41
NCES *see* National Center for Education Statistics
natural language processing (NLP): AIG and 109, 136–137; human raters and 58–59, 61; NN and 78; iSTART and 183–197; machine translation and 200–214; multiple choice test items, NLP methodology for 77; scoring system for patient notes 58–71
Neumeyer, L. 40
neural networks (NNs): Char- and Word-RNNs 94; DNNs 90; NLP tasks and applications 78; RNNs 91–92, 94, RNNs as imputational model of deep learning 96; training of RNNs 97, 98; as universal functional approximators 94
NICHD *see* National Institute of Child Health and Human Development
NLG 90, 103
null hypothesis 154, 157

OECD *see* Organization for Cooperation and Development
Open AI 90, 92, 94; Codex 103; parameter model 95; WebText 95; *see also* GPT
oral reading fluency (ORF) 41; ASR-based testing 45; automated testing of 44; Moby.Read 42
Organization for Cooperation and Development (OECD) 200, 213
orthographic word form 36
orthographic word neighbor and word neighbor distance 191, **192**, 194, 195
OWL *see* Web Ontology Language (OWL)

Page, E. 58–59, 61, 143
parser 168
part-of-speech tagger 168
phonics 41, 44, 45
picture annotations 93, 96
picture description writing **110**, 113, 114, *115*, 119
Pilehvar, M. T. 168
"pills" utterance 36, 37, *38*
PISA *see* Programme for International Student Assessment
polyglot framework 239
polyseme 169
polysemous words: number of 168
PRMSE *see* proportion reduction of mean squared errors
Programme for International Student Assessment (PISA) 60, 91, *97*, 142, 200, 201, 213
ProphetNet 79
proportion reduction of mean squared errors (PRMSE) 17, 157, 159; line plots of *160*
psychometricians 155
psychometrics 127; characteristics, items 137; computational, for digital-first assessments 107–121; deep learning and challenges when using 15–26; fairness and 143; key concepts 144–145, 148; standard assumptions 146
PubMed 90, 100, 101, 169, 175
Python 10, 80, 81, 100; *scikit-learn* library 176; unit test *7*
PyTorch 98

quadratic weighed kappa (QWK) 17, 150, 157; Cohen's QWK **22**, 148; differences in exact agreement and QWK between Classical and Hybrid (Classical + BERT) engines on operational responses (Grades 3–8, $n \approx 500$ per Item, 12 Items) **26**; differences in exact agreement and QWK between Classical and Hybrid (Classical + BERT) engines on teacher-scored responses (Grades 3, 5, 7, and 9, $n \approx 75$ per Item, 11 Items) **25**; Haberman's generalization for continuous measure 149; performance on two essays for Classical, LSTM, and Transformer (BERT)-based models across increasing training sample sizes *20*
Quellmalz, E. 136
Question Answering (QA) NLP application 78
QWK *see* quadratic weighted kappa

Radford, A. 90
Ranasinghe, T. 81–82

RC *see* reference corpus
reading instruction cycle in grades 1–3 *42*
reference corpus (RC) 80–81, 82
REML 145
Reynolds, D. 34
RNNs 91–92, 94, Char- and Word-RNNs 94; as imputational model of deep learning 96; training of 97, 98
ROBERTa 173, 175, 178, 180n19

sample item-presentation page with a listing of features and functions *42*
SAT score 218–220, 222, 223; ACT/SAT 220, 223, 224, 226; *see also* GPA
SBERT 82, **83**, 84
Schneider, Mark 231
second language (L2) 120
second language listening and speaking 31
second language speaking ability, assessment of 45–53
Semantic Textual Similarity (STS) benchmark 82, 87n7
Shannon, C. 93
Shin, J. 78
Siamese network structures 82
SIDE *see* Summarisation Integrated Development Environment
simulation-based assessments 134; IUA and 134–136
simulation-based science classroom 136
simulation tasks 127, 136, 139
Singh, A. 100
Singh Bhatia, A. 78
SLP *see* spoken language processing
Slaney, M. 40
Spearman correlation coefficient 82
speech analysis in assessment 31–55
speech recognition and response scoring within a computer-based assessment with items that elicit spoken responses *41*
spoken language processing 31, *44*; AMI 43; automatic speech recognition in 35–36
stealth assessment 185, 188, 192
stealth literary assessment 183–196; iStart and 185–196; NPL and 184
SUBTLEXus corpus of subtitles 191
Summarisation Integrated Development Environment (SIDE) 202
support vector machine (SVM) 143, 196
support vector regressor (SVR) 143, 196
surrogates 58, 59; use of, in scoring 60–61
Susanti, Y. 78
SVC 143
SVM see support vector machine
SVR *see* support vector regressor

TAACO 186
TAALES 188, 191, **192**, 194–196
TabNine 92, 98, 100
Taghipour, K., and Ng., H.-T. 20
TensorFlow (Google) 94, 98, 99, 100
text-based corpora 202
text yes/no 110, **110**, *111*, 117
TIMSS *see* Trends in Mathematical Science Study
TM *see* translation memory
TOEFL 45, **54**, 108, 142, 221
TOEIC 45
transformer and transformer-based language model 19, *20*, 23, 24, 90, 94–97, 103–104, 117, 172; DeBERTa 173; Electra 173; T5 179; potential of 96; pretrained 100–101, **101**; retraining 92; training of 97; very large 90; *see also* BERT; GPT
transformer neural network structure 92
Transformer-XL 95
translational memory (TM) 79, 80, 81, 84, **85**, **86**
Trends in Mathematical Science Study (TIMSS) 104–105, 200, 201, 213
Turnitin 221, 231

unit test 7
universal background model (UBM) 34
universal function approximators, NNs as 94
Universal Sentence Encoder 87

validation approaches, three key elements 17
validity: accuracy, fairness, and validity 5–6, 17–18; automated scoring, validity and fairness in 3, 9–10, 12, 127–139; criterion validity coefficients 107; fairness as validity concern 18, 141–142; IUA and validity arguments for automatic scoring engines 131–134; reliability, generalizability, and validity 107, 109; technology-based assessment, fairness, and 127–139; validity argument 21, 53, 144; Williamson's understanding of 5
variance 33, 44, 130, 135; asymptotic 154; construct-irrelevant 118, 120, 127, 128, 130, 132–136, 159–160; covariance 153; marginal 151; mean and 149–150; residual 152; in reading test scores 191–192, 194–196; rubric-irrelevant 147; in vocabulary scores 188, 222
variance component methods 145
variance reduction abilities 176
Vast.AI 97, 98
Vaswamni, A. 94
Versant English Test 45
Versant Spanish Test 45–53
voice-spoofing technologies 34
von Davier, Matthias 92, 109, 138

wampimuk experiment 169
waveform 32; nonnative speaker's 22-second spoken response *32*; spectrogram of 'two pills', spoken slowly in isolation *37*
Web Ontology Language (OWL) 78
WER *see* word error rate
Williamson, D. M. 5, 17, 18, 131
Witt, S. M. and Young, S. J. 40
Word2Vec 19, 24, 82, 169, 171–173, 175; *see also* Doc2vec
word embeddings 169–171
word error rate (WER) 35, 38–40
word-node 'pills' *38*

Yamamoto, K. 60
Yaneva, V. 138, 175

Zhang, M. 131
Zhang, X. 34
Zierky, M. J. 118