Leveraging Natural Language Processing Methods to Evaluate Automatically Generated Cloze Questions: A Cautionary Tale

by

Guher Gorgun

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Measurement, Evaluation, and Data Science

Department of Educational Psychology
University of Alberta

Abstract

The purpose of this dissertation is to employ three prominent natural language processing methods to assess the feasibility of automatically evaluating cloze questions generated by automatic item generation (AIG) methods. AIG methods have been developed to address the need for a large number of items for computerized assessments as well as online learning environments. Yet, traditional methods for evaluating item quality are limited to evaluating each generated item and providing information about the quality of the generated items. In this study, we first provided an exhaustive overview of item quality criteria and evaluation methods used by AIG researchers. This allowed us to portray the current evaluation practices, their advantages, and limitations. We proposed a taxonomy of current evaluation methods used for AIG studies, namely, metric-based evaluations, human evaluators, and post-hoc evaluations. Given that current evaluation methods have several limitations and typically cannot be used for evaluating all generated items, we examined three natural language processing methods for evaluating item quality automatically. As such, this is a proof-of-concept study investigating the feasibility of various natural language processing methods for item evaluation. In this study, we used automatically generated cloze questions evaluated by crowdsource workers to investigate the utility of three prominent natural language processing methods for item evaluation. Thus, we examined the capacity of incorporating NLP and ML methods in item evaluation process for automatically generated items to render item evaluation more feasible. These methods included training three machine learning classifiers (i.e., random forest, support vector machine, and logistic regression) using linguistic features extracted

from item stems and keyed responses (Study 1), fine-tuning a large-language model, namely BERT (Study 2), and instruction-tuning a generative large-language model, namely, Llama-2 (Study 3). In Study 1, the best-performing classifier was the logistic regression, followed by the random forest and support vector machine. Nonetheless, the results of ML classifiers highlighted that they are quite limited in predicting item quality. In Study 2, we fine-tuned BERT-Large and BERT-Base and found an improvement in item quality prediction compared to Study 1 results. In Study 3, the performance of the instruction-tuned Llama-2 model surpassed all other methods and achieved an acceptable performance for identifying item quality. Overall, the findings suggested the promise of tuning generative large-language models by providing specific instructions regarding item quality for automatically evaluating the quality of generated items.

Preface

This thesis is an original work by 'Guher Gorgun'.

Chapter 2 of this thesis is published as Gorgun G. & Bulut O. (2024). Exploring quality criteria and evaluation methods in automated question generation: A comprehensive survey. *Education and Information Technologies.* https://doi.org/10.1007/s10639-024-12771-3. I was responsible for conceptualization, methodology, formal analysis, and original manuscript preparation. O. Bulut supervised the study and contributed to the original manuscript–reviewing & editing.

Chapter 2 of this thesis has been presented as Gorgun, G., & Bulut, O. (2024, February). *Current evaluation methods are a bottleneck in automatic question generation.* Poster presented at the Workshop on AI for Education: Bridging Innovation and Responsibility at the AAAI Annual Conference on AI, Vancouver, BC. I was responsible for conceptualization, methodology, formal analysis, and original proposal preparation. O. Bulut supervised the study and contributed to the original proposal preparation–reviewing & editing.

Study 3 of Chapter 4 has been presented as Gorgun, G. & Bulut, O. (2024, April). *Using large language models for evaluating item quality in large-scale assessments.* Paper presented at the annual meeting of the National Council on Measurement in Education, Philadelphia, PA. I was responsible for conceptualization, methodology, formal analysis, and original proposal preparation. O. Bulut supervised the study and contributed to the original proposal preparation–reviewing & editing.

Acknowledgements

In addition to my dissertation committee, I would like to extend my thanks to the following people who immensely supported me during my doctoral journey.

I am grateful for the support and academic freedom my supervisor, Dr. Okan Bulut, provided during my doctoral studies. It was extremely fulfilling to be able to pursue my research interests. I am also grateful for the rich research opportunities and experiences he has provided. These opportunities helped me to become a better researcher and to find my voice as a researcher.

I also would like to express my deep appreciation to Dr. Carrie Demmans Epp. She has tremendously supported my professional development with the resources, guidance, and mentorship she has provided, even though she has no obligation to do so. I am always astonished by the resources and suggestions she has provided whether it be for research or professional development. I also truly appreciate being a part of the EdTeKLA Research Group and receiving invaluable feedback from the research group members.

I am thankful for the summer research internship opportunity at Educational Testing Service (ETS) which has been extremely insightful in understanding the complexities of automatic item generation and evaluation. I am grateful for my supervisors, Drs. Ikkyu Choi and Jiyun Zu, for providing mentorship and the opportunity to get to know other fantastic researchers at ETS.

I would like to thank Dr. Kim Colvin for guiding and supporting me during my graduate studies. Thanks to her mentorship and guidance, I made informed decisions regarding my academic and professional development.

I am also extremely grateful for Dr. Sevilay Kilmen's support. She has always encouraged and motivated me to become a better researcher.

I owe thanks to my colleagues and friends in the Measurement, Evaluation, and Data Science program who made me feel at home and provided support and intellectual growth.

I also would like to acknowledge the financial support I have received during my doctoral studies from Alberta Innovates, the University of Alberta, Kansas University, and Mitacs.

Of course, this journey would not be such a pleasant and rewarding one without the support of my parents, Gaye and Nejat Görgün. I am so grateful to them for supporting me and my aspirations and providing love, encouragement, and a willingness to be all ears when I need them.

Contents

List of Tables

List of Figures

Abbreviations

AI Artificial Intelligence.

AIG Automatic Item Generation.

AUC Area Under the Curve.

BERT Bidirectional Encoder Representations from Transformers.

BiDAF Bidirectional Attention Flow.

BLEU BiLingual Evaluation Understudy.

BP Brevity Penalty.

CAT Computerized Adaptive Test.

CTT Classical Test Theory.

Det Determiner.

DIF Differential Item Functioning.

ELMo Embeddings from Language Model.

FN False Negative.

FP False Positive.

FPR False Positive Rate.

GloVe Global Vectors for Word Representation.

GPT Generative Pre-trained Transformer.

GPU Graphics Processing Unit.

IRT Item Response Theory.

LCS Longest Common Subsequence.

Llama Large Language Model Meta AI.

LLM Large-Langauge Model.

LR Logistic Regression.

LSA Latent Semantic Analysis.

LSTM Long-Short Term Memory.

M Mean.

METEOR Metric for Evaluation of Translation with Explicit ORdering.

ML Machine Learning.

MTLD Measure of Textual Lexical Diversity.

NLG Natural Language Generation.

NLP Natural Language Processing.

NLTK Natural Language Toolkit.

NP Noun Phrase.

P Precision.

PEFT Parameter-Efficient Fine-Tuning.

PP Prepositional Phrase.

QLoRA Quantized Low Rank Adapters.

R Recall.

RF Random Forest.

RoBERTa A Robustly Optimized BERT Pretraining Approach.

ROC Receiver Operator Characteristic Curve.

ROUGE Recall-Oriented Understudy for Gisting Evaluation.

S Sentence.

SD Standard Deviation.

sense2vec Senses-to-Vector.

seq2seq Sequence-to-Sequence Model.

SME Subject Matter Expert.

SQuAD Stanford Question Answering Dataset.

SVM Support Vector Machine.

T5 Text-to-Text Transfer Transformer.

TN True Negative.

TP True Positive.

TPR True Positive Rate.

VP Verb Phrase.

word2vec Words-to-Vector.

Chapter 1

Introduction

Assessment is an indispensable tool in education that allows making inferences about various agents (e.g., teachers, students) and institutions (e.g., schools, educational districts) of education (Nagy, 2000; Newton, 2007). A well-designed assessment (Palomba & Banta, 1999) supplies essential information about the performance of test-takers. Stakeholders, then, can use this information to make inferences and conclusions about schools, teachers, students, or educational jurisdictions. For example, assessment results can be used for grading, award decisions, monitoring students' progress, or identifying at-risk students. Additionally, international (e.g., Programme for International Student Assessment) (Bulle, 2011) and national large-scale assessments (e.g., National Assessment of Educational Progress) (Rampey et al., 2009) can be employed for evaluating the performance of educational systems across countries or states, assessing accountability of schools and school districts, granting accreditation to educational institutions, or understanding teacher effectiveness (Cole et al., 2008; Ewell, 2008; Flowers et al., 2001; Goertz & Duffy, 2003; Heubert & Hauser, 1999; Linn, 2003; Zilberberg et al., 2013). Educational assessments, therefore, should have high quality (e.g., validity and reliability) because they help measure learning and teaching, which are not directly observable and visible in nature.

Typically, educational assessments are composed of presumably high-quality items to measure learning and teaching precisely (Livingston, 2018). One can conceptualize

items within a given assessment as building blocks of that assessment. Items are also primary indicators, providing validity evidence about the usefulness of an assessment (Haladyna & Rodriguez, 2013). That is, it is not possible to create a high-quality assessment before ensuring that the items included in the assessment are of high quality. For example, items included in an assessment should reflect what is being measured, providing one strand of validity evidence (Smith, 2003). An assessment needs multiple items to cover what is being measured and taught. Furthermore, we also know that as the number of items in an assessment increases, the confidence in an assessment augments, and the standard error of measurement decreases (Smith, 2003). However, we should underscore that merely increasing the number of items in an assessment will not be sufficient to enhance the validity and reliability of an assessment. Those items should be of high quality to contribute to the overall quality of the assessment. Thus, we should assess, evaluate, and understand each item's quality when creating a high-quality assessment.

Over the last two decades, the demand for a large number of items has increased substantially, thanks to technology incorporation into assessment and learning environments. We classify the need for a large item bank into two reasons: (1) advances in assessment practices and (2) innovations in learning environments. With the increased access and use of computers due to their flexibility and efficiency in test administration, scoring, and reporting, computerized assessments have been widely used in school, statewide, and international assessments (Drasgow & Olson-Buchanan, 1999; Mills et al., 2019; Ras & Brinke, 2015). Computerized assessments require a large item repository (i.e., item bank) (Wright & Bell, 1984) in order to increase precision in estimation, adaptivity, and diversity while controlling for item exposure. Additionally, with the advances in computer technology, computerized adaptive tests (CATs) have become more widespread. CATs may shorten the test length drastically while depicting an accurate picture of examinees' ability levels (Bulut & Kan, 2012; Weiss, 1982, 2004; Weiss & Kingsbury, 1984). Because adaptive tests, unlike fixed-

length assessments, select items matching the interim ability of the examinee, those tests necessitate a large item bank (Breithaupt et al., 2010) with diverse items in terms of difficulty and discrimination.

In addition to the changing nature of test administration practices, formative assessments gained extensive attention from the education community because of the way policymakers, researchers, and practitioners have conceptualized current educational objectives (e.g., the importance of lifelong learning and critical thinking skills rather than factual knowledge) and its relationship to learning and knowledge-based societies (Dębska & Kubacka, 2012; OECD, 2009; Su, 2015). Unlike summative assessments which solely focus on the outcomes of learning, formative assessments (also known as assessments for learning) allow us to recognize the process during learning. Formative assessments essentially involve feedback about instruction and student learning allowing one to monitor students' progress and evaluate the effectiveness of the instruction (Black & Wiliam, 1998; Sadler, 1989). Just like any type of assessment, formative assessments also require diverse, multiple, and high-quality items to accurately measure learning and instruction.

With the changing nature of learning environments, one can observe that formative assessments have been embedded in online, interactive, and digital learning systems (Corbett et al., 1997; VanLehn, 2011). Digital learning environments can provide real-time, personalized, and immediate feedback (Feng & Heffernan, 2005) with the possibility of tutoring services such as through conversational agents (Graesser et al., 2004, 2014; Sosnowski & Yordanova, 2020) or scaffolding exercises (Feng et al., 2006; Razzaq & Heffernan, 2006) that break down the assessment item into smaller chunks of knowledge components. Such learning environments need to include a vast number of formative assessment items that are typically aligned with the program of studies or the curriculum such as the common core state standards (Kober & Rentner, 2012). Digital learning environments include thousands of items to better cater to student and teacher needs. As such, diverse and more practice items could be provided to

learners in these environments to enhance learning and knowledge retention (Kochmar et al., 2022).

However, supplying large quantities of high-quality items to computerized assessments and online learning environments is not easy. Traditional item development is an iterative process that involves several steps, such as item writing, item evaluation, and field testing. Subject matter experts are profoundly involved in each step. The item development process starts with subject matter experts writing and developing individual assessment items. After items are developed, subject matter experts review, revise, and edit newly developed items based on their expertise (Gierl et al., 2021a; Lane et al., 2016). Typically, qualitative ratings have been used to attain standardization in human evaluations at this stage (Gierl et al., 2016). After consensus has been reached in terms of item quality ratings and refinements have been made to the newly developed items, items are administered to a representative sample (i.e., field-testing or pretesting) to obtain empirical item quality indices (e.g., statistical analysis performed for conducting item analysis) (Gierl et al., 2021b, 2022). Field testing and item analysis are again followed by an expert review to evaluate whether items can be used for operational test settings, or revisions and refinements should be made to improve the item quality for operational use.

Nonetheless, traditional item development has several pitfalls, rendering the traditional item development process infeasible for producing a large number of high-quality items for computerized assessments and online learning environments (Gierl & Haladyna, 2012; Gierl & Lai, 2012; Gierl et al., 2021a). First, in traditional approaches to item development, the unit of item production and analysis are individual items. That is, each item is written and evaluated on an individual basis, leading to extended periods of time devoted to item development and validation (Romberg et al., 1982). One may assume that once a large quantity of items is developed by subject matter experts and test developers, they can be used an indefinite number of times. Yet, we need to underscore that it is an unrealistic expectation about items

and item banks. Item banks need to be maintained and amended to control for item exposure or item drift (Guo et al., 2017) as well as to keep up with rapid changes observed in some subject areas such as medicine (Gierl & Haladyna, 2012). As item exposure increases, test security may decrease as more and more examinees take the same set of items (Barrada et al., 2009). In addition to being time-consuming, the traditional item development process is also expensive. Rudner (2010), for instance, asserted that the development of a single item costs around US$1,500 to US$2,500 (p. 157). Consider that an intelligent tutoring system requires thousands of items to adequately monitor students' progress and provide tutoring services. Item development and refinement alone will cost hundreds of dollars. Therefore, traditional item development is quite limited in responding to today's educational assessment needs and challenges.

To address these concerns, a novel approach to item development—automatic item generation (AIG)—has been proposed, which focuses on diminishing the resource-intensive process of item creation by utilizing computing power and automation (Gierl et al., 2023). Both selected- and constructed-response items can be generated by AIG methods (Becker et al., 2012; Gierl et al., 2016; Seyler et al., 2017; Wang, Lan, & Baraniuk, 2021; Yang et al., 2021). The two ends of the spectrum of AIG methods approach item generation quite differently. We provided an overview of both ends of automatic item generation in Figure 1.1. At the one end of the spectrum, template-based AIG (Bejar et al., 2002; Gierl & Haladyna, 2012; Gierl et al., 2023; Irvine & Kyllonen, 2002), utilizes item templates and cognitive models for item generation. The workflow of template-based AIG starts by identifying the content to be used for item generation (Gierl & Lai, 2013). This first step is labeled as the cognitive model (Gierl & Haladyna, 2012) of AIG where skills, knowledge components, and abilities are determined, which will be assessed by the test. The following step (Step 2) is referred to as an item model (Laduca et al., 1986), where item templates are developed for manipulating the content to be measured by the test. During the second step, the

stem, the set of alternatives, including both distractors and keyed responses, and the auxiliary information such as tables, figures, and graphs are created by subject matter experts. Note that subject matter experts are profoundly engaged in both Step 1 and Step 2. In Step 3, using the computer technology as well as the cognitive models and item templates created in Steps 1 and 2, new items are generated.

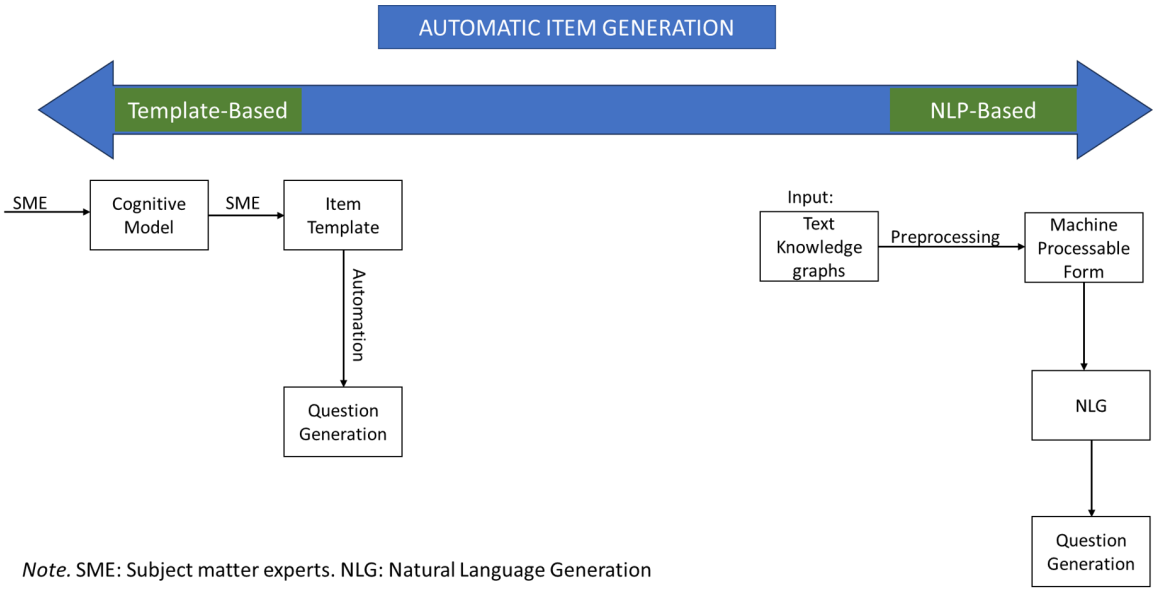


Figure 1.1: An Overview of Template-Based and NLP-Based AIG Branches.

On the other end of the spectrum of item generation is natural language processing (NLP)-based AIG which utilizes NLP techniques to generate items, reducing the dependency on subject matter experts (Kurdi et al., 2020; Lu & Lu, 2021; Mulla & Gharpure, 2023). Note that NLP-based approaches may combine NLP techniques with computer vision (Sarrouti et al., 2020) or deep learning (Kumar et al., 2018) to generate questions. There are multiple approaches proposed by NLP-based AIG. Broadly put, NLP-based approaches can be grouped under two headings: 1) syntactic models and 2) semantic models. While the former generates items by leveraging syntactic elements of an input text, the latter focuses on contextual features such as named entity recognition (Lu & Lu, 2021).

Unlike template-based approaches that heavily depend on subject matter experts

for the creation of cognitive models and item templates (Bezirhan & von Davier, 2023), NLP-based approaches are considered fully automated approaches, significantly reducing the degree of human involvement during the item generation process. Furthermore, in templated-based AIG models computer aid and automation come at the last stage (i.e., Step 3); hence, several researchers have referred to this branch of item generation as semi-automatic AIG. Besides the differences in terms of the degree of automation between template-based and NLP-based AIG, template-based AIG is also not applicable to some content areas such as reading comprehension (Gierl et al., 2023). Yet, dependency on subject matter experts in template-based AIG also has substantial advantages compared to NLP-based AIG, such as conserving the generated items against certain item quality deficiencies (e.g., sounding unnatural or being irrelevant to the target content) or having better control when creating alternatives for a multiple-choice item.

1.1 Problem Statement

Despite the palpable advantages of AIG (e.g., increasing test security and generating numerous items easily and in a cost-effective way) (Gierl et al., 2021a; Kosh et al., 2019), there are several limitations of AIG approaches that restrict the use of generated items in operational test settings. An obvious advantage of AIG (i.e., generating hundreds, if not thousands, of test items) may become a subtle drawback of the items generated via automated approaches. The quality of generated items is typically unknown. Using the traditional item evaluation methods, the generated items cannot be evaluated effectively and efficiently by employing subject matter experts or by administering all of the items to a representative sample (i.e., it is not realistic to administer, say, 20,000 items in a reasonable time). Additionally, item parameters (e.g., item difficulty and discrimination) cannot be obtained for all items generated using traditional approaches used for item analysis. These inhibit the use of generated items in operational testing, whether it be an assessment on its own or an assessment

embedded in an online learning environment. For example, in a computerized adaptive test, we need to know item parameters to select items that maximize information about examinees (Weiss & Kingsbury, 1984). Likewise, in online learning, we need to have at least a rough estimation of item difficulties to select items tailored according to the examinees' needs (Wauters et al., 2012).

In the traditional item development process, field testing is a vital step for evaluating item quality. By field testing items, test developers obtain statistical estimates of item analysis (e.g., difficulty or distractor functioning). For evaluating thousands of generated items, field testing is not a feasible and efficient method. That is, it is not realistic to recruit examinees or embed items in an operational test to conduct item analysis in order to estimate item difficulty or discrimination. Current AIG models field test only a subsample of generated items (Gierl et al., 2016; Van Campenhout et al., 2022) underscoring the hardship of field-testing all generated items. Furthermore, item statistics obtained through field testing are conditional upon the quality of data, and the obtained statistical estimates can be context-dependent (i.e., examinee or assessment characteristics may influence item analysis (Gorgun & Bulut, 2021, 2022). For instance, the testing context (e.g., low-stakes test), examinee motivation, target population (e.g., Grade 3 students), test length, item position, and test speededness are a few examples that may influence the data quality, hence statistical item indices. Additionally, Livingston (2013) argued that statistical analysis of items should provide information about items rather than examinees (p. 421). However, the response data generated by examinees taking the assessment cannot be thought to be independent of examinee and test characteristics. Hence, the precision of statistical analysis of items (e.g., item calibration) will depend on how closely they resemble actual examinees and testing conditions (French, 2001). A final consideration about statistical estimates of item quality is that our current conceptualization and operationalization of item quality could be limited in regards to recognizing a reciprocal (i.e., bidirectional) interaction between examinees and test items. That is, examinees are not

passive entities from which we can extract objective information about items; rather, there is an interaction among test settings (e.g., low stakes), items (e.g., difficult), and examinee characteristics (e.g., effortful examinees) (Gorgun & Bulut, 2023).

In addition to field testing, human evaluations have been used in the traditional item development process for rating developed items in terms of difficulty, presentation, and content relevance (Gierl & Lai, 2016). While human ratings have been considered as the ground truth in artificial intelligence (AI)-oriented research, previous research showed that human evaluators could be poor estimators of item quality (Seyler et al., 2017), sometimes introducing bias and subjectivity in item quality evaluations. To overcome these issues, rating scales and training procedures are introduced in item quality evaluations. Nevertheless, human evaluators are expensive and slow, rendering them inefficient evaluators of the quality of generated items. Human evaluators, therefore, are also a bottleneck in item quality evaluations (Lin & Demner-Fushman, 2006).

With thousands of generated items, these traditional evaluation processes (i.e., expert judgment and field testing) are no longer feasible for evaluating each generated item (Gierl & Lai, 2012). While template-based approaches may be more suitable in terms of item quality evaluations due to subject matter experts' involvement in item model development, both ends of the spectrum of AIG approaches are susceptible to the limitations of current item quality evaluation methods. Therefore, we may need to find new evaluation methods that expand the boundaries of traditional item evaluation methods to offer more feasible, scalable, and automatic approaches to item quality evaluation. Developing new evaluation methods may not only facilitate the item evaluation process of automatically generated items but may also contribute to the item generation process by rendering it more feasible and efficient (Lin & Demner-Fushman, 2006).

1.2 Purpose of Dissertation

AIG is an interdisciplinary method combining measurement theory, learning sciences, and computer science. Thus, it requires an interdisciplinary perspective for the design, development, deployment, and evaluation of such systems. For this reason, we reviewed the literature focusing on human-computer interaction and measurement theory. While human-computer interaction refers to the development, evaluation, and deployment of computer systems for human use (Shneiderman et al., 2016; Sinha et al., 2010), measurement theory means evaluating the quality of psycho-educational measurement and assessment tools in terms of accuracy, consistency, usefulness, and predictive value (Allen & Yen, 2001; Brennan, 2006). Although in recent years, the capabilities of AIG systems to generate questions evolved tremendously due to generative AI (Nguyen et al., 2022), evaluation of these items for deployment is still a major issue. Employing an interdisciplinary perspective allowed us to provide a comprehensive overview of current evaluation methods used by researchers in measurement and human-computer interaction disciplines working on AIG, portray the limitations of current evaluation methods, and propose a novel approach leveraging NLP to test the feasibility of evaluating generated items automatically. To facilitate the evaluation process, we propose to test the feasibility of three prominent NLP methods for assessing the quality of generated items and assess the boundaries of automating the item evaluation process for generated cloze questions. As such, this study is a proof-of-concept. Proof-of-concept refers to a pilot project that assesses the feasibility of a design concept or proposal.

We first scan the AIG articles to create a taxonomy of evaluation methods used in AIG research. This allows us to categorize evaluation methods based on their similarities, limitations, and advantages as well as identify quality criteria used by AIG researchers when evaluating the generated items. This process informs our first study where we test our first NLP method for item evaluation.

The first method employs extracting linguistic features from the generated cloze items, and training three machine learning (ML) classifiers. The goal of this study (Study 1) is to assess the feasibility of a simpler yet interpretable method for item evaluation. Study 1 also analyzes the feature importance for each trained classifier to inform AIG researchers regarding which linguistic features might be useful (Paramythis et al., 2010) for evaluating item quality automatically.

The second method assesses the utility of fine-tuning a pre-trained large-language model (LLM) for evaluating the quality of generated cloze items. The goal of this study (Study 2) is to use a more complex language model that is uninterpretable but showed great promise for automating human rating assignments (e.g., essay scoring) (Beseiso & Alzahrani, 2020) by adjusting the weights of the parameters of a pre-trained language model.

The third method employs a generative LLM with 7-billion parameters to assess the efficiency of using a prompt-based approach to item evaluation. The goal of this study (Study 3) is to provide instructions to LLM along with cloze item and item quality labels to assess whether generative models can be trained to follow instructions for item quality evaluation.

1.3 Research Questions

1. What are the common evaluation methods and quality criteria used for evaluating automatically generated items?

   This question is answered in the literature review section where we provide an exhaustive overview of evaluation methods as well as the quality criteria commonly used by AIG researchers.

2. Can we leverage NLP methods to automatically evaluate the quality of generated items?

   2.1. To what extent a classifier trained with linguistic features can be used for

evaluating automatically generated cloze items?

This question is answered in Study 1 where we extracted linguistic features and trained three machine learning classifiers.

2.2. To what extent a fine-tuned pre-trained large-language model can be used for evaluating automatically generated cloze items?

This question is answered in Study 2 where we fine-tuned an LLM, namely, BERT for item quality prediction.

2.3. To what extent an instruction-tuned generative large-language model can be used for evaluating automatically generated cloze items?

This question is answered in Study 3 where we instruction-tuned Llama-2 for item quality prediction.

3. Which linguistic features are more important for predicting item quality?

This question is answered in Study 1 where we analyzed the importance of the linguistic features for each machine learning classifier trained.

Chapter 2

Literature Review

In this chapter, we provide a comprehensive overview of item quality criteria used in both traditional and automated item development processes as well as evaluation methods used in the AIG research. By scanning the AIG literature, we first summarize the quality criteria used in traditional item development and in AIG to highlight similarities and differences between the criteria used in both approaches to item development. By doing so, we intend to bridge the gap between traditional item developers and automated item generators. We then propose a taxonomy of evaluation methods used by AIG researchers to facilitate the discussion around similarities, advantages, and limitations of current item evaluation methods used by AIG researchers. The chapter concludes with a summary of the limitations of current evaluation methods and argues for the need for novel methods for item evaluation in AIG.

2.1 Item Quality Criteria in Traditional Item Development

Item is a technical term that psychometricians and test developers use to refer to an individual question, exercise, prompt, statement, or task in a test or questionnaire that the examinee or respondent reacts through selecting, constructing, or performing a response (American Educational Research Association et al., 2014; Nelson, 2004). Items are building blocks of an assessment or test, hence before talking about the

overall quality of an assessment (e.g., validity), each item should attain a certain level of quality to be included in a test. Attaining information about item quality is typically referred to as *item analysis*. Item analysis provides information about the empirical and structural quality of items (Bandalos, 2018; Lane et al., 2016; Osterlind, 1989). In the literature, one can observe various reasons and objectives attached to performing item analysis. For example, item analysis supplies essential information about several indices of item quality including item difficulty, item discrimination, differential item functioning (Livingston, 2013), distractor functioning (Gierl et al., 2017; Haladyna & Rodriguez, 2013; Thissen et al., 1989), and the dimensionality of the construct being measured (Haladyna & Rodriguez, 2021). It also helps researchers assemble parallel test forms, evaluate the content coverage (Gierl et al., 2021a), and identify items that need to be removed or revised in a given item bank (Haladyna & Rodriguez, 2021). While some researchers consider item analysis as a statistical procedure to extract information about item characteristics (Ashraf, 2020; Clauser & Hambleton, 2011; French, 2001; Rezigalla, 2022), others assert that judgment-based methods can be used for conducting item analysis (Gierl et al., 2021b, 2022; Osterlind, 1989). Below, we first summarize judgment-based approaches used for item analysis. Then, we discuss various statistical procedures used for evaluating psychometric criteria under different measurement theories.

Judgment-based approaches may employ subject-matter experts, examinees, or editorial specialists (Osterlind, 1989) to systematically review and evaluate items according to some criteria. Evaluation criteria may include the degree of alignment between the content and item coverage, congruence between skills and knowledge required from the examinee and item, and grammatical accuracy (Gierl et al., 2016, 2021b). In addition to these criteria, researchers also emphasized the importance of systematically evaluating item quality using measures such as rating scales. For instance, Osterlind (1989) suggested using a 3-point scale for evaluating content alignment with the levels of high congruence, medium congruence, and low congruence (pp.

267-268).

Nonetheless, judgment-based evaluations are not as common as statistical approaches used for item analysis. This might be because item analysis relying on statistical methods is expected to provide more objective measures of item quality. The first index we discuss is item difficulty which is used to quantify the degree of hardness of an item. Under classical test theory (CTT), item difficulty is formulated as the proportion of examinees (i.e., proportion correct score or p-value) who answered the item correctly (Anastasi & Urbina, 2004; Clauser & Hambleton, 2011; Haladyna & Rodriguez, 2021). Thus, as the item difficulty increases, the number of examinees answering the item correctly decreases. As a more advanced method, item response theory (IRT) formulates item difficulty based on a probability function. IRT places item difficulty and examinee ability on the same continuum and expresses item difficulty in terms of the probability of correctly answering an item. If the examinee's ability is higher than an item's difficulty, the examinee is more likely to answer the item correctly and thus the item is easier given the ability level of the examinee. When an item's difficulty and the examinee's ability coincide, the probability of answering the given item is 50% (Osterlind & Wang, 2017).

Another commonly used item quality criterion is item discrimination. Item discrimination is about how well an item can distinguish high-performing examinees (i.e., examinees who know the content well) from low-performing ones (French, 2001). Several indices have been proposed to quantify item discrimination. In CTT, item discrimination is estimated through various approaches: 1) difference in proportion correct scores when high-achievers are compared with low-achievers, 2) point-biserial correlation, 3) biserial correlation, and 4) multi-serial index. The first approach builds on identifying performance groups (i.e., lower group and upper group) based on the total test score. Specifically, examinees' total test scores are ranked in an ascending manner and 25% or 27% of both upper and lower performance groups are determined. After identifying the lower and upper performing groups, the proportion correct scores

for both groups are calculated and the lower group's proportion correct score is subtracted from the upper group's to compute the discrimination index (Cohen et al., 1996; Jenkins & Michael, 1986; Kehoe, 1995). The second and third discrimination indices are based on Pearson's product-moment correlations and while the point-biserial correlation estimates item discrimination by finding the product-moment correlation between the total score and item score, the biserial correlation employs the continuity correction (Ebel & Frisbie, 1986; Kim et al., 2021). The final discrimination index, the multi-serial index, is a multiple correlation considering the response options of an item (Haladyna & Rodriguez, 2021). Unlike the previous three indices discussed, this index also considers distractor discrimination. On the other hand, according to the IRT-based approaches, item discrimination is related to the steepness of the item characteristic curve where the steeper line indicates that the item more effectively differentiates the lower ability examinees below the item location (i.e., difficulty) from the higher ability examinees above the item location (Ashraf, 2020).

Several guidelines have been proposed regarding the interpretation of difficulty and discrimination indices (Cohen et al., 1996; French, 2001). For example, Cohen et al. (1996) suggested that the optimal proportion correct index is approximately .50. While Ebel and Frisbie (1986) suggested that items with a discrimination index of .40 or higher are good items, Clauser and Hambleton (2011) indicated that classroom assessments should have item discrimination at least above 0.0. Yet, these guidelines for item difficulty and discrimination have been criticized because of their sample dependency, especially in the context of CTT-based approaches, as well as examinees' random guessing behavior (Anastasi & Urbina, 2004). Unlike CTT-based approaches, difficulty and discrimination indices are less sample-dependent in an IRT context. However, we need to note that difficulty and discrimination parameters will be impacted by the exam characteristics and the sample of examinees that is used for calibrating the items in IRT-based approaches. For instance, in the presence of high rates of rapid guessing, item parameters can be inflated (e.g., overestimated) and

items may seem more difficult than their actual difficulty levels (Gorgun & Bulut, 2021). Additionally, test speededness, not-reached, omitted, or unanswered items, content stratification, calibration practices (e.g., embedded field items), and item position effects are some of the issues that may drastically influence item difficulty and discrimination parameters. Therefore, the quality and trustworthiness of difficulty and discrimination parameters estimated via either CTT- or IRT-based approaches depend on the sample, the purpose of the assessment, and assessment administration contexts (Hambleton, 1993; Hambleton et al., 1991; Livingston, 2013).

The next criterion we discuss for evaluating item quality is item bias or differential item functioning (DIF). Unlike difficulty and discrimination indices, bias is less frequently emphasized as a criterion of item quality in AIG research. However, the presence of biased items may lead to unfair interpretations and consequences for examinees, hence, items should be inspected for possible item biases across different groups (e.g., race or ethnic background, gender, or region) (Clauser & Hambleton, 2011). The presence of a biased item means that, given the ability levels of examinees, one group is disadvantaged due to group membership. That is, differential item functioning occurs when the probability of correctly answering an item is different due to group membership, albeit we match the ability levels of subgroups. Even though judgmental approaches can be used to evaluate whether items exhibit bias, several statistical approaches have also been developed for empirically detecting bias in items. In these empirical methods, a focal (i.e., disadvantaged group) and a reference group are identified for examining whether an item functions differently for one of the groups (e.g., whether the item difficulty is greater for the focal group) (Clauser & Hambleton, 2011; Livingston, 2013).

The item evaluation criteria discussed so far can be used for both selected-response (e.g., multiple-choice item) and constructed-response (e.g., short-answer) items. In addition to these criteria, additional item quality criteria were specifically developed for selected response items. One such criterion is distractor analysis. There are

three main reasons for examining distractor functioning. These reasons are as follows: 1) whether all distractors have been selected by examinees, 2) whether the keyed response is correctly identified, and 3) whether distractors are more likely to be selected by low-performing students (Gierl et al., 2016; Haladyna et al., 2002). A non-functioning distractor means that none of the examinees have selected the distractor suggesting that the distractor is not plausible or attractive to examinees. Secondly, it is expected that the keyed response is the most frequently selected answer when analyzing examinee responses. Finally, distractors should be more likely to be selected by low-performing examinees because high-quality distractors typically involve common misconceptions that low-performing examinees have. Thus, item discrimination analysis focusing on distractors should indicate that distractors are more attractive to low-performing students (Gierl et al., 2016).

2.2 Item Quality Criteria in AIG

In AIG research, computers are the medium for item generation. Note that in Chapter 1, we made a distinction among different AIG systems with varying levels of automation. While some systems approach item generation as a fully automated task (Ha & Yaneva, 2018), some still heavily rely on humans (e.g., subject matter experts) (Gierl & Haladyna, 2012), and automation comes at the very last stage. Although fully automatic systems minimize human dependency, time, and financial costs, such systems are more susceptible to issues with item generation. This introduces new challenges to item quality evaluation methods. It could be the case that an AIG system generates non-fluent and ungrammatical items that do not necessarily sound like a human-generated natural language (e.g., nonsensical). Therefore, psychometric approaches may lack these considerations related to NLP for providing a comprehensive evaluation framework for AIG. Thus, we need to consolidate psychometric approaches with the challenges that AIG brings to develop a more appropriate framework for AIG evaluations. By reviewing the current quality criteria employed by AIG researchers, we

examine the aspects absent in traditional psychometric evaluation criteria. Furthermore, outlining evaluation criteria in AIG research helps us identify quality criteria informed by psychometrics that are neglected in AIG evaluations.

2.2.1 Linguistic Criteria

The most common linguistic criteria include grammaticality, fluency, relevancy, semantic correctness, syntax clarity, meaning, spelling, naturalness, coherence, and specificity (Amidei et al., 2018; Gatt & Krahmer, 2018; Kurdi et al., 2020; Mulla & Gharpure, 2023). While several research studies include more than one linguistic criterion with a clear description of each criterion (Heilman, 2011; Heilman & Smith, 2010); others lacked a clear description or definition of the criteria included (Becker et al., 2012; Mostow et al., 2017). Here, we need to make a distinction between linguistic criteria and the rating scale used for evaluating generated items. In some studies, linguistic criteria are embedded into the rating scale used. In such cases, researchers start by providing a rating scale (e.g., Good, Okay, or Bad) and they define each category with linguistic criteria. For instance, Becker et al. (2012) and Niraula and Rus (2015) used a 3-point rating scale with categories of Good, Okay, and Bad. They defined a good item as "the questions that ask about the key concepts from the sentence and are reasonable to answer", an okay item as "the questions that target the key concepts but are difficult to answer...", and a bad item as "questions which ask about an unimportant aspect of the sentence, or their answers are easy to guess from the context" (p. 197).

In a similar approach, Heilman and Smith (2010) and Heilman (2011) provided a detailed description of the rating criteria they used for evaluating their rule-based over-generate and rank item generation system. Specifically, they first asked the evaluators to assign one of the five rating scale categories based on the generated sentence quality: good, acceptable, borderline, unacceptable, and bad. If evaluators scored a generated sentence as unacceptable, then evaluators were further asked to enumer-

ate the reasons for unacceptability using the following criteria: (un)grammaticality, incorrect information, vagueness, and awkwardness/other. A strength of their approach is that they defined each quality criterion and reasons for unacceptability. For example, an unacceptable question is defined as "The question definitely has a minor problem" and ungrammaticality is defined as "The question is ungrammatical, does not make sense, or uses the wrong question word (e.g., who, what, which, etc.)" (pp. 183-184). Finally, the researchers also provided examples to the evaluators which can be regarded as an attempt to achieve some level of standardization among evaluators during the evaluation of generated items.

Unlike these rating scale-focused approaches discussed above, several researchers approached the linguistic evaluation of generated items from a criteria-focused perspective. Specifically, they focused on one or several linguistic criteria (e.g., fluency, meaningfulness, plausibility) and evaluated the item's quality dichotomously. In a recent study, Wang, Liu, et al. (2021) used fluency (i.e., coherent and grammatically correct) and relevancy (i.e., whether the item is relevant to the input context-answer pair) as two linguistic criteria (p. 7). Likewise, using grammaticality (i.e., the presence or absence of grammar errors) and semantic correctness (i.e., whether the overall meaning of the generated question is relevant to the context without vagueness), Liu et al. (2017) evaluated generated factual questions. Even though the authors provided the definitions of the criteria used to evaluate question quality, these definitions are vulnerable to different interpretations introducing undesirable noise into the item quality evaluation process. Furthermore, different definitions are used to describe the same linguistic criterion underscoring the lack of uniformity and cohesion in evaluation methods focusing on linguistic aspects of generated questions.

Perhaps a more fine-grained approach for evaluating item quality using linguistic criteria is utilizing a rubric-like rating scale. That is, providing several linguistic criteria and descriptions with a rating scale that allows evaluators to distinguish different degrees of linguistic criteria. For example, Maurya and Desarkar (2020) employed

grammatical correctness and distractability evaluation criteria with a 5-point rating scale (i.e., very poor to very good). This type of approach allows evaluators to consider each criterion separately while enabling them to indicate the degree of presence of each linguistic criterion.

Nonetheless, the linguistic criteria used, and their definitions were not always specified. In some studies, the researchers briefly mentioned that generated items were considered grammatically and syntactically acceptable without providing information on how these criteria were defined and evaluated (Das et al., 2016). For instance, Mostow et al. (2017) and Rodriguez-Torrealba et al. (2022) provided linguistic criteria that evaluators need to consider without a clear definition of each criterion. In their study, Mostow et al. (2017) asked evaluators to rate the quality of items generated in terms of spelling, syntax, clarity, and meaning with a 3-point rating scale (i.e., very poor, moderate, and very well). In the study by Rodriguez-Torrealba et al. (2022), evaluators rated the items based on correctness, plausibility, nonsensicality, and ungrammaticality. Yet, without a clear definition, it is very likely that evaluators may provide their own definitions for each linguistic criterion, introducing bias in the item quality evaluation process.

2.2.2 Psychometric Criteria

The most common psychometric criteria considered in AIG research are item difficulty, distractor analysis, domain relevance, and educational usefulness. In addition to those common indicators of item quality, a few studies took student engagement (Van Campenhout et al., 2022), cognitive models (Gierl et al., 2016, 2022), internal consistency (Hommel et al., 2022; von Davier, 2018), and differential child item functioning (Fu et al., 2022) into account while evaluating the quality of generated items. Similar to linguistic criteria, one can observe diversity in the way researchers defined and used psychometrically informed criteria. Most of this discrepancy also stems from using different approaches to item generation. Based on our review, we

can assert that template-based AIG methods follow more psychometrically informed criteria with little to no emphasis on linguistic criteria due to heavy dependency on subject matter experts during item model development. Additionally, those studies tend to follow a more detailed analysis of the psychometric properties of generated items (e.g., distractor functioning, item discrimination) (Gierl & Haladyna, 2012; Gierl & Lai, 2012; Gierl et al., 2016) which are more aligned with traditional item analysis. On the other hand, a heavy emphasis is given to linguistic criteria for NLP-based AIG approaches while a few loosely-defined psychometric criteria are used for evaluating the quality of items generated with these approaches. This is probably due to the generated items by NLP-based methods being more susceptible to sounding non-natural and non-sensical.

The most frequently used psychometrically informed criterion is item difficulty. Item difficulty, though, was not used invariably across the AIG studies. Several studies approached item difficulty from an empirical perspective by collecting examinee data. For example, by field testing a sample of items generated from medical templates, Gierl et al. (2016) estimated item difficulty as the proportion of examinees who answered the item correctly (i.e., CTT definition of item difficulty). A similar approach to item difficulty estimation was adopted by Van Campenhout et al. (2022) where they compared the difficulty of the human-authored and automatically generated items using students' response data. Specifically, item difficulty was based on the proportion correct-scores of students' first attempt.

Several studies conceptualized difficulty based on cognitive complexity (e.g., Bloom's taxonomy of items) (McCarthy et al., 2021; Settles et al., 2020; Venktesh et al., 2022). Unlike empirical approaches described above, these approaches typically assign item difficulty labels (e.g., easy item, C1 level, moderate item) based on cognitive frameworks or subject-matter expert judgment. In some of these studies, item difficulty was conceptualized as the degree of similarity between item distractors and the keyed response. This conceptualization allowed AIG researchers to generate difficulty-

controllable items (Alsubait, 2015; Alsubait et al., 2014). For instance, using a binary label of item difficulty (i.e., an easy or difficult item), Kurdi et al. (2017), and using a three-level label of item difficulty (i.e., easy, medium, difficult) Lin et al. (2015) conceptualized item difficulty as the similarity between distractors and the keyed response. As the similarity between distractors and keyed response increases, the confusability of distractors increases (Seyler et al., 2017), rendering the generated item more difficult. Similarly, Gao et al. (2019) developed a difficulty-controllable AIG system where the difficulty of an item was determined by employing two reading comprehension systems (i.e., R-Net and BiDAF) (Seo et al., 2018; Wang et al., 2017). Accordingly, if both systems answered a given question correctly under the exact match metric, the question was labeled as easy whereas if none of the systems answered the item correctly, the question was labeled as difficult. Additionally, linguistic features can be extracted from the item stem or alternatives to develop prediction models of item difficulty. Researchers may utilize word embeddings (e.g., BERT), word length, sentence length, or word-level unigram language models to predict item difficulty (McCarthy et al., 2021; Settles et al., 2020).

Analysis of alternatives (i.e., keyed response and distractors) of a multiple-choice item is another psychometric criterion that AIG researchers considered when evaluating automatically generated items. Like item difficulty, one can observe variations in terms of how the examination of alternatives was performed. In line with traditional psychometric analysis, several researchers analyzed distractor functioning in terms of whether each distractor had been mostly selected by lower-performing examinees and whether each distractor had been used (i.e., distractors become non-functional if they are not appealing to lower-performing students and thus have not been selected). For example, Gierl et al. (2016) investigated the distractor functioning of two generated items by examining the discrimination index and biserial correlations. The discrimination index should be zero or negative and biserial correlations should be negative to have properly functioning distractors because it indicates that lower-performing

students are more likely to select distractors.

Another psychometric criterion that a few studies considered is domain relevance. A typical approach for evaluating domain relevance is conducted by employing subject matter experts. This quality concern is aligned with the psychometric criterion of being instructionally relevant (Gierl & Haladyna, 2012; Gierl & Lai, 2012). For instance, Gierl et al. (2021b) suggested using a 4-point rating scale for evaluating the content relevance of generated items. The rating scale included the categories of accept, accept – minor revision, reject – major revision, and reject allowing test developers, subject matter experts, and psychometricians to decide which item models can be used in operational test settings. Although template-based AIG approaches are less susceptible to being domain irrelevant and are easier to evaluate domain relevancy due to subject-matter experts' involvement in the creation and evaluation of item models, several NLP-based item generation systems also considered domain relevance as an evaluation criterion (Afzal & Mitkov, 2014; Alsubait et al., 2016; Chughtai et al., 2022), for example, generated items targeting specific lecture slides of a software engineering course. Generating domain-relevant questions requires the system to identify facts, terminologies, or relationships which are essential components of instructional material. To this end, researchers used named entities (Afzal & Mitkov, 2014); knowledge bases representing relationships among the instructional entities (Rodríguez Rocha & Faron Zucker, 2018; Song & Zhao, 2017); or domain ontologies (Kurdi et al., 2017).

Educational or pedagogical usefulness is another psychometrically informed criterion. Because education usefulness partly depends on the validity argument (Kane, 2013), for example, the purpose of the item use (Mazidi & Nielsen, 2014), some researchers operationalized educational usefulness to assess the AIG system's performance. While some considered usefulness as a support for human learning through content coverage (Jouault et al., 2016; Tamura et al., 2015) and prompting deep thought (Zhang & VanLehn, 2016), others formulated usefulness as a combination

of linguistic criteria composed of, for instance, grammaticality, relevancy in terms of information, and semantic quality (Flor & Riordan, 2018).

Using these linguistic and psychometrically informed criteria, researchers used various evaluation methods and evaluators to assess item quality. Below, we propose a taxonomy of evaluation methods in AIG research for assessing the quality of generated items.

2.3 A Comprehensive Overview of Evaluation Methods Used in AIG

In this section, we provide a taxonomy of evaluation methods used in AIG research. We argue that providing a comprehensive overview of evaluation methods helps us highlight the resources needed for each evaluation method, their advantages, and limitations while emphasizing the need for new evaluation methods to successfully deploy items generated automatically. The proposed taxonomy of evaluation methods is based on resources (e.g., whether reference questions are available), input (e.g., whether response data are used), and quality criteria (e.g., whether psychometric methods are used for evaluation) that researchers have used when evaluating items. Note that many studies are cited across multiple evaluation methods because studies typically combine several evaluation methods (Amidei et al., 2018).

2.3.1 Metric-Based Evaluations

According to the Dictionary of Oxford, the term metric is defined as a system or standard of measurement. Metric-based evaluations in the AIG literature refer to a family of standard measures that automatically evaluate the performance of the system, typically against human performance or ground truth (Gao et al., 2019; Kumar et al., 2018; Marrese-Taylor et al., 2018; Wang et al., 2018). In Table 2.1, we provided examples of AIG studies that used metric-based evaluations.

The most frequently used metrics are BiLingual Evaluation Understudy (BLEU)

Table 2.1: Examples of AIG Systems Evaluated Using Metric-Based Method

Authors	Item Types	Context	AIG Method	Evaluation Method
Becker et al. (2012)	Cloze	Generic	Parse-trees	Logistic regression
Gao et al. (2019)	Constructed response	Reading comprehension	seq2seq	BLEU, METEOR, ROIUGE-L
Ha and Yaneva (2018)	Distractor	Medicine	Concept embeddings, Information retrieval	Embedding similarity
Liang et al. (2018)	Distractor	Biology, Chemistry, Earth science, Physics	Feature-based, Neural net	Logistic regression, Random forest, LambdaMart
Liu et al. (2017)	Constructed response	Factual questions	Sentence simplification	Logistic regression, RankSVM
Marrese-Taylor et al. (2018)	Cloze	Language	Bidirectional LSTM	F1, Recall, Precision
Maurya and Desarkar (2020)	Distractor	Reading comprehension	Hierarchical multi-decoder network	BLEU, ROUGE-L, METEOR, Embedding average
Kumar et al. (2018)	Q&A pairs	Reading comprehension	seq2seq, Answer encoding	METEOR, BLEU, ROUGE-L
Panda et al. (2022)	Distractor generation, Cloze	Language	Neural machine and round-trip machine translation	Specialized metric
Rodriguez-Torrealba et al. (2022)	Multiple-choice, Answer, Distractor	Generic	T-5	BLEU, ROUGE-L, Cosine similarity
Wang et al. (2018)	Constructed response	Biology, Sociology, History	Recurrent neural network (bidirectional LSTM)	BLEU, METEOR, ROUGE-L
Wang, Lan, and Baraniuk (2021)	Constructed response	Math	Pre-trained large-language models	BLEU, METEOR, ROUGE-L, Specialized metric
Wang et al. (2022)	Constructed response	Biology	GPT-3, Prompt engineering	Perplexity, Distinct-3, Toxicity

Note. Adapted from Gorgun and Bulut (2024b).

(Papineni et al., 2002), Metric for Evaluation of Translation with Explicit ORdering (METEOR) (Banerjee & Lavie, 2005), and Recall-Oriented Understudy for Gisting Evaluation (ROUGE) (Lin, 2004). These metrics were originally developed for evaluating machine translation tasks and comparing the performance of the system against human performance. Given that human evaluators are expensive (Hovy, 1999) and costly in terms of time (Papineni et al., 2002), researchers developed various automatic evaluation metrics (e.g., BLEU) that allow system developers to assess the quality of machine translation in terms of closeness to human translations. Additionally, these metrics help researchers and practitioners quantify the magnitude of closeness to human performance to efficiently evaluate the system's performance.

The BLEU metric (Papineni et al., 2002) counts the number of n-gram matches between the candidate translation (i.e., machine translation) and reference translation (i.e., human translation). The matches are not considered with respect to the position of the n-grams and the higher number of matches is associated with a better candidate translation. By counting the number of n-gram matches, BLEU computes a precision score. The precision score can be given as

$$p_n = \frac{\sum_{\mathrm{C}\epsilon\{Candidates\}} \sum_{n-gram\epsilon \mathrm{C}} Count_{clip}(n-gram)}{\sum_{C'\epsilon\{Candidates\}} \sum_{n-gram'\epsilon C'} Count_{clip}(n-gram')}, \tag{2.1}$$

where $Count_{clip}$ is the total count of each candidate word by its maximum reference count. However, BLEU also introduces a penalty function for sentence brevity when estimating the similarity between machine translation and human translation. Specifically, brevity penalty (BP) is defined as

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r, \end{cases} \tag{2.2}$$

where r is the reference corpus length and c is the total length of the candidate translation corpus. Finally, the BLEU score is obtained by

$$BLEU = BP * exp(\sum_{n=1}^{N} w_n log_{p_n}), \tag{2.3}$$

where $w_n = 1/N$ and N is the number of total words. The BLEU metric generates a value between 0 and 1, and higher scores indicate better translation.

Another machine translation metric that AIG researchers use is ROUGE (Lin, 2004) and it has four variations: ROUGE-N, ROUGE-L, ROUGE-W, and ROUGE-S. Yet, AIG systems exclusively used ROUGE-L which focuses on finding the longest common subsequence between a candidate translation and a reference translation. The underlying objective is similar to the BLEU metric—ROUGE-L aims to identify the overlap between two sequences (i.e., a summary sentence). It is conceptualized as an LCS-based F-measure (Lin, 2004) and can be written as

$$F_{LCS} = \frac{(1+\beta^2)R_{LCS}P_{LCS}}{R_{LCS} + \beta^2 P_{LCS}}, \tag{2.4}$$

where β is P_{LCS}/R_{LCS}. Here R_{LCS} LCS-based recall and is defined as

$$R_{LCS} = \frac{LCS(X,Y)}{m}, \tag{2.5}$$

and P_{LCS} LCS-based precision and is defined as

$$P_{LCS} = \frac{LCS(X,Y)}{n}, \tag{2.6}$$

where X and Y are two summaries with a length of m and n, respectively. Similar to BLEU, this equation generates a value between 0 and 1, and higher scores indicate a better candidate summary.

The final member of the automatic machine translation metrics is METEOR (Banerjee & Lavie, 2005). Similar to ROUGE-L, METEOR evaluates the match between two sequences by combining precision and recall values, however, METEOR also considers the order of the matched words. Developers of the METEOR metric tried to overcome the limitations inherent in the BLEU metric by considering recall, word matching between reference and candidate summaries, and the level of grammaticality of the candidate summary (Banerjee & Lavie, 2005). Formally, METEOR is defined as

$$Score = F_{mean} * (1 - Penalty), \tag{2.7}$$

where F_mean is a combination of unigram recall (R) and unigram precision (P)

$$F_{mean} = \frac{10PR}{R + 9P}, \tag{2.8}$$

and $Penalty$ is defined as

$$Penalty = 0.5 * (\frac{\text{the number of chunks}}{\text{the number of unigrams mathced}})^2. \tag{2.9}$$

Recall that METEOR tries to improve the BLEU metric by considering the length of the matches between candidate and reference translation. Hence, the Penalty function achieves that goal by considering chunks (i.e., unigrams in adjacent positions in candidate translation) and unigrams matched. The METEOR metric has a similar interpretation to BLEU and ROUGE-L.

In the AIG literature, these metrics are used when there is a reference question or distractor typically authored by humans. Using the SQuAD data set (Rajpurkar et al., 2016), which includes question and answer pairs, Gao et al. (2019) and (Kumar et al., 2018) evaluated the quality of generated questions by estimating the BLUE, METEOR, and ROUGE-L metrics. Specifically, they considered questions available in the dataset as reference questions while generated questions were considered candidate items. Similarly, Maurya and Desarkar (2020) evaluated the quality of generated distractors with the human-authored ones using the same metrics.

In addition to metrics adapted from machine translation, researchers also used other metrics such as perplexity, F1, grammatical error using Python language tool, toxicity analysis, embedding similarity, and some specialized metrics specifically developed for a given item generation task (Amidei et al., 2018). To provide an example of specialized metrics, we refer to a study by Wang, Lan, and Baraniuk (2021). The researchers developed an AIG system for generating math word problems using math equations representing the math word problem. The researchers developed the *ACC-eq* metric to evaluate the similarity between the input equation and the mathematically represented generated math word problem. Similarity-based metrics such

as GloVe cosine similarity (Rodriguez-Torrealba et al., 2022) or BERT cosine similarity (Maurya & Desarkar, 2020) are employed for measuring the semantic distance between the generated question and reference text. Finally, in a recent study, Wang et al. (2022) evaluated the quality and diversity of generated questions focusing on the coherence of the text, the number of averaged grammatical errors, the average number of distinct tri-grams in questions generated, and the toxicity of generated questions.

While metric-based methods can be easily and quickly applied to evaluate the quality of generated questions, they have a few limitations. First, the data set should include ground truth or human-authored questions or distractors to be able to employ these metrics. Second, because these metrics consider the similarity between the generated and human-authored questions, the other acceptable questions may receive a low score (Kumar et al., 2018), entailing that the perfectly valid questions are considered bad due to differences in the linguistic structure.

2.3.2 Human Evaluators

Perhaps the most popular evaluation method for assessing the quality of questions generated by AIG systems is relying on manual or human evaluators (Kurdi et al., 2020) as they are considered the golden standard for evaluating the fluency and grammaticality of generated questions. Note that human evaluators are used for various purposes (Kurdi et al., 2020) including evaluating the question difficulty (Rodriguez-Torrealba et al., 2022), language fluency (e.g., plausibility or grammaticality) (Mostow et al., 2017), distractibility of items (Maurya & Desarkar, 2020), or domain relevance (Chughtai et al., 2022). In Table 2.2, we provided examples of AIG systems with human evaluators.

We categorize human evaluators into five groups: experts, students, crowdsourcing, researchers, and teachers. Note that there are also gray areas in the literature regarding the identity of human evaluators. For instance, in some studies, researchers only

Table 2.2: Examples of AIG Systems Evaluated Using Human Evaluators

Authors	Item Types	Context	AIG Method	Evaluation Method
Attali et al. (2022)	Multiple-choice	Reading comprehension	GPT-3	Experts
Becker et al. (2012)	Cloze	Generic	Parse-trees	Crowdsource
Chughtai et al. (2022)	Multiple-choice	Engineering	T-5, Sense2vec	Experts
Chung and Hsiao (2022)	Constructed response	Programming	Template-based	Teachers
Dugan et al. (2022)	Constructed response	Generic	T-5	Researchers
Gierl and Lai (2016)	Multiple-choice	Medicine	Template-based	Experts
Liang et al. (2017)	Distractor	Biology, Math, Physics	Generative adversarial neural nets	Experts
Lin et al. (2015)	Multiple-choice	Wildlife	Hybrid semantic similarity	Crowdsource
Mostow et al. (2017)	Multiple-choice	Biology, Sociology, History	Recurrent neural network (bidirectional LSTM)	Students
Wang, Lan, and Baraniuk (2021)	Constructed response	Reading comprehension	Parse-trees, n-grams	Students
Olney (2021)	Cloze	Science	Deep learning summarization	Experts, students
Panda et al. (2022)	Distractor, cloze	Language	Neural machine translation, round-trip machine translation	Students
Rodriguez-Torrealba et al. (2022)	Multiple-choice, answer, distractor	Generic	T-5	Professionals
Song and Zhao (2017)	Constructed response	Generic	Neural machine translation	Not reported
von Davier (2018)	Survey	Personality scale	Recurrent neural network, LSTM	Crowdsource

Note. Adapted from Gorgun and Bulut (2024b).

mentioned that evaluators are native speakers of English without providing any information regarding their educational background or demographic information about these evaluators (Maurya & Desarkar, 2020; Song & Zhao, 2017).

Experts are typically domain or subject matter experts of an assessment or content area (i.e., the target domain of AIG). Experts typically use a rating scale to evaluate the question quality. Quality indicators that experts use to rate generated items include grammaticality (Chughtai et al., 2022; Heilman, 2011), fluency (Song & Zhao, 2017), domain or question relevance (Dugan et al., 2022), complexity (Chung & Hsiao, 2022), acceptability of questions (Gierl et al., 2016; Liang et al., 2017), and question clarity (Rodriguez-Torrealba et al., 2022).

In addition to experts, several studies employed students (Panda et al., 2022), teachers (Chung & Hsiao, 2022), and researchers (Dugan et al., 2022) to evaluate the quality of generated items. Here, we make a distinction between student, teacher, and researcher evaluators and expert evaluators because the former group (i.e., teachers students, and researchers) may lack content or assessment expertise and the lack of thereof can substantially bias the quality assessments of the generated items. Similar to expert evaluators, student, researcher, and teacher evaluators may use a rating scale to judge the generated items' quality (Mostow et al., 2017; Rodriguez-Torrealba et al., 2022).

Expert, teacher, student, and researcher evaluators are limited when it comes to evaluating the large number of items generated with computer algorithms. Because studies need to recruit a large number of experts, students, teachers, or researchers (Maurya & Desarkar, 2020) to assess the quality of the vast number of items generated, crowdsourcing has emerged as a viable alternative to evaluate the quality of generated items quickly and easily. Crowdsourcing allows researchers to recruit a larger number of evaluators which is an economical substitute in terms of time and money. One of the most popular crowdsourcing platforms is Amazon Mechanical Turk (Sorokin & Forsyth, 2008; Strickland & Stoops, 2019). Evaluators recruited through

crowdsourcing typically assess the item quality using a rating scale (Lin et al., 2015), similar to experts, teachers, students, and researchers. However, there are many gray areas in the literature regarding whether crowdsource workers have gone through any training to attain standardization during the evaluation process. Alternatively, the crowdsource workers may be asked to take the assessment in order to obtain empirical data to evaluate the quality of the generated items (Becker et al., 2012; Hommel et al., 2022; von Davier, 2018). However, a potential problem with using the crowdsource worker data to conduct item analysis might be the lack of representativeness of the crowdsource workers as they may not be a good representation of examinees taking the real assessment.

Although human evaluations in question generation are considered the golden standard by providing the ground truth about the item quality, the quality of several human evaluators can be doubtful due to the lack of detailed reporting practices or standardization during the evaluation process. The lack of standardization can lead to biased evaluations regarding the quality of generated questions. For instance, studies include a different number of evaluators ranging from 1 to 364 (Amidei et al., 2018). In AIG research involving more than one evaluator, the inter-rater agreement between the evaluators is seldom reported. In conjunction with this point, in most of the studies, it is unclear whether human evaluators received any training to standardize the rating process of generated questions (Kurdi et al., 2020). Furthermore, the lack of consistency in rating-based evaluations (e.g., using different rating scales or evaluation criteria) makes it extremely hard to compare the performance of various AIG systems with respect to one another. In some studies, the recruitment process of human evaluators and whether any incentives are offered to evaluators are unclear. A detailed description of evaluators, recruitment process, training, rating scale development, and tools used should be reported to be able to gauge the evaluation process, especially considering that idiosyncrasies that human evaluators may introduce during the evaluation process (Lin et al., 2015).

Table 2.3: Examples of AIG Systems Evaluated Using Post-Hoc Methods

Authors	Item Types	Context	AIG Method	Evaluation Method
Attali et al. (2022)	Multiple-choice	Reading comprehension	GPT-3	Psychometric properties
Gierl et al. (2012)	Multiple-choice	Medicine	Template-based	Psychometric properties
Hommel et al. (2022)	Survey	Personality	Recurrent neural network, LSTM, GPT-2	Psychometric properties
Van Campenhout et al. (2022)	Matching, cloze	Psychology	Rule-based	Experimental
Yang et al. (2021)	Cloze	Reading comprehension	BERT	Experimental

Note. Adapted from Gorgun and Bulut (2024b).

2.3.3 Post-Hoc Evaluations

Post-hoc evaluations refer to collecting data to assess the quality of generated items retrospectively (i.e., assessing the item quality after administrating the items) in AIG research. Post-hoc evaluations include experimental studies and psychometric analyses. Table 2.3 lists examples of studies that used post-hoc evaluations to analyze the quality of generated items.

Researchers may evaluate the quality of the AIG system by examining whether the generated questions are associated with higher learner performance. That type of evaluation is referred to as an experimental study. For example, Van Campenhout et al. (2022) compared human-authored items with generated items using student data taking both item types. They found that generated items functioned similarly to human-authored items in terms of item difficulty, student engagement, and persistence. Similarly, in an experimental study comparing students' reading engagement and reading skills in practice quizzes composed of automatically generated items, researchers found a statistically significant result in terms of higher course performance for the experimental group who practiced the content with the generated items (Yang et al., 2021).

Another type of post-hoc evaluation is psychometric analyses of items generated. Using this approach, AIG researchers administer a representative sample of generated items to a group of individuals and then estimate item quality indices (e.g., distractor functioning, item difficulty as discussed in Section 2.1) employing either CTT or IRT. For example, using a representative subsample of generated items, Gierl and Haladyna (2012) field-tested generated items using three medical templates and evaluated the quality of items based on item difficulty, distractor analysis, and keyed response analysis. In a similar study, Attali et al. (2022) also field-tested generated items and evaluated item quality in terms of item difficulty, local independence, and response time.

2.4 Limitations of Current AIG Evaluation Methods

In this section, we summarized the limitations of current evaluation methods to highlight that current methods persist as a bottleneck to transition from item development to item deployment in real-world learning and assessment settings. In Table 2.4, we provided an overview of the limitations of each evaluation method.

Table 2.4: A Summary of Limitations of Current Evaluation Methods

Limitations	Metric-Based	Human Evaluators	Post-Hoc
Resource intensive	✓	✓	✓
Availability of reference items	✓		
Availability of ground truth	✓		
Quality of ratings		✓	
Time consuming		✓	✓
All items cannot be evaluated		✓	✓
Costs		✓	✓
Sample representativeness			✓

Metric-based evaluations necessitate a reference item such as item stem or distrac-

tors (Gao et al., 2019; Ha & Yaneva, 2018) to be able to run metrics such as BLEU or ROUGE-L. In addition, this family of evaluation methods may utilize ground truth to develop prediction models (Becker et al., 2012; Marrese-Taylor et al., 2018). The need for the availability of ground truth or reference items emphasizes the resource intensity of these methods for item evaluation. Finally, evaluating generated items based on the resemblance to reference items can be a major limitation as metrics such as BLEU, ROUGE-L, and METEOR may discard perfectly valid items due to dissimilarity in the structure of the item (Kurdi et al., 2020).

Concerning human evaluators, a major limitation is that humans cannot evaluate all generated questions within a reasonable timeframe. Typically, employing human evaluators is costly and item evaluation is a time-intensive process for humans. While a premise of AIG is that items can be generated instantly and efficiently, human evaluators function contradictory to this promise of quick item generation, violating the fundamental assumption of AIG (Maurya & Desarkar, 2020). Furthermore, the quality of the rating scale as well as expert training are gatekeepers to high-quality item evaluation, emphasizing the resource-intensity of employing human evaluators (Amidei et al., 2018; Kurdi et al., 2020).

Post-hoc evaluations can also violate the fundamental assumption of AIG because all generated items may not be field tested to obtain quality criteria about the item (Gierl et al., 2016; Van Campenhout et al., 2022). Similar to human evaluators, post-hoc evaluations are time-consuming, costly, and require resources to recruit a representative sample of examinees to be able to assess the quality of generated items. While many studies have failed to emphasize the importance of replicating assessment and examinee characteristics while evaluating items using post-hoc evaluations, deviations from actual assessment conditions may jeopardize the quality of indices obtained as well as the inferences made based on post-hoc evaluations.

In the final part of this section, we mapped item quality criteria onto the evaluation methods to stress which quality criteria can be assessed with the current

Table 2.5: Mapping Quality Criteria on Current Evaluation Methods

Quality Criteria	Metric-Based	Human Evaluators	Post-Hoc
Spelling	✓	✓	
Semantic correctness	✓	✓	
Grammaticality		✓	
Fluency		✓	
Syntax clarity		✓	
Naturalness		✓	
Coherence		✓	
Domain relevance		✓	
Educational usefulness		✓	✓
Item difficulty			✓
Item discrimination			✓
Distractor Analysis			✓

evaluation methods (Table 2.5). While the number of quality criteria that metric-based evaluations (e.g., spelling, semantic correctness) and post-hoc evaluations (e.g., item difficulty, discrimination, distractor analysis) can be quite limited, the number of quality criteria that human evaluators can handle is drastically higher. It is not surprising that many AIG research employs human evaluators when it comes to assessing the quality of generated (Amidei et al., 2018; Kurdi et al., 2020; Soni et al., 2019). That being said, human evaluators also have several limitations with respect to quality criteria, that is, prior research indicated that humans are poor evaluators of item difficulty (Bejar, 1983; Chalifour & Powers, 1989; Olson, 2010; Seyler et al., 2017).

2.5 Chapter Summary

This chapter focuses on summarizing evaluation criteria used by traditional item developers and AIG researchers, as well as common evaluators and evaluation methods in AIG. While providing recent examples from AIG systems, we create a taxonomy of evaluation methods used for automated item-generation tasks. As a concluding remark, in most studies, researchers combine multiple evaluations and evaluators to assess the system's performance regarding item generation. By identifying the strengths and limitations of current quality criteria and evaluation methods, this chapter underscores the need for alternative approaches to facilitate the deployment of automatically generated questions in real-world educational settings.

Chapter 3

Methods

In this chapter, we describe three NLP methods that we used for evaluating the quality of questions automatically generated. This is a proof-of-concept for assessing the feasibility of utilizing NLP to automatically evaluate the quality of generated questions and comparing the performance of three prominent NLP methods in terms of model accuracy. To overcome the challenges highlighted in Chapter 2 concerning the commonly employed evaluators and evaluation methods, we propose to develop three prediction models leveraging different NLP and machine learning methods:

1. Classifier training with feature extraction
2. Fine-tuning a pre-trained non-generative large-language model
3. Instruction-tuning a generative large-language model

The methods section is, therefore, divided into three related studies where we compared three different NLP methods for evaluating automatically generated cloze questions. The first study, *Classifier training with feature extraction*, aimed at training three ML classifiers, namely random forest, support vector machine, and logistic regression, by employing linguistic features extracted using cloze item stems and keyed responses. Using these three white-box classifiers allowed us to understand feature importance for quality prediction and to assess the congruence between classifiers in terms of feature importance. We argued that the first study allows us to

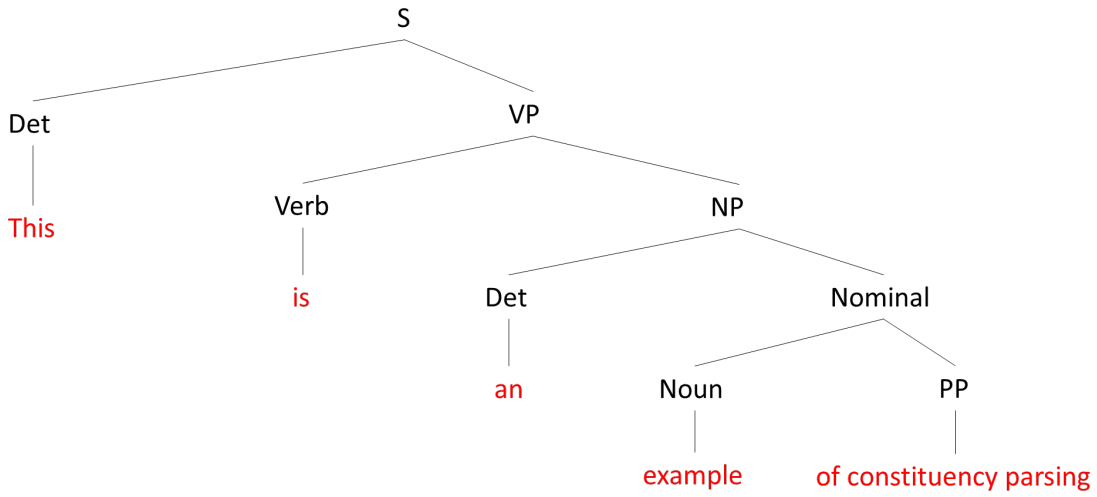
understand the link between item quality and linguistic features (Bejar, 1983; Seyler et al., 2017), shedding light on what linguistic criteria can be included in item evaluations. The second study, *Fine-tuning a pre-trained large-language model*, involves tokenizing the item stems to extract embeddings and adding a linear output layer to re-train the large-language model for adjusting the weights of parameters based on our input data. An advantage of the second approach is that the model re-trained for downstream tasks (e.g., predicting item quality) performs well with small datasets because the parameters already encode a lot of information due to being trained with a vast amount of text with a specific model architecture (Vaswani et al., 2017). The third study, *Instruction-tuning a generative large-language model*, employs a rather new approach to utilizing generative large-language models for downstream tasks. Instruction-tuning involves providing an instruction, example, and output to adjust the weights of a pre-trained large-language model for a specific task. Below, we discussed the dataset employed for training three methods for predicting the quality of automatically generated items.

3.1 Data

A publicly available dataset composed of automatically generated cloze items is used for assessing the feasibility of all three NLP-based methods for automatically evaluating the quality of generated questions. Cloze questions are a type of constructed-response item where a part of the sentence is removed, and examinees are asked to fill in the blank in a given sentence. Typically, no response option is provided with cloze questions, rendering these questions more difficult than selected-response items (Abraham & Chapelle, 1992). Cloze questions also pose an interesting research context because a part of the stem is masked deliberately and extracting features or tuning large-language models is challenging due to less linguistic information being available.

We used the *Mind the Gap* dataset (Becker et al., 2012) involving a total of 2252

automatically generated items. Becker et al. (2012) used 105 popular Wikipedia articles, including topics from history and science. They then selected 10 sentences from each article using a document summarization approach. Specifically, sentences from each article are selected based on whether they involve the most frequently occurring words considering that the sentences with the most frequently occurring words are most important for a given article. Using each selected sentence, Becker et al. (2012) created a parse tree using the constituency parser method (see Figure 3.1) and identified the most important parts of the sentence by employing a semantic role labeler. The candidate gaps (i.e., masked parts) in a sentence involved child nouns and adjectival phrases. Several examples of generated cloze questions and quality labels assigned by crowdsource workers are given in Table 3.1.



Note. S: Sentence. Det: Determiner. VP: Verb pronoun. NP: Det nominal. PP: Preposition pronoun.

Figure 3.1: An example of constituency parsing.

Each generated item was evaluated using crowdsource workers recruited through Amazon's Mechanical Turk (Lin et al., 2015; Maurya & Desarkar, 2020; von Davier, 2018). Becker et al. (2012) asked crowdsource workers to rate the quality of generated items using the quality labels of *Good*, *Okay*, and *Bad*. Specifically, crowdsource workers were presented with the original sentence along with the generated cloze

Table 3.1: Automatically Generated Cloze Item Examples and Quality Labels Assigned by Crowdsource Workers

Generated Cloze Item	Keyed Answer	Quality Label
They are differently named parts of the whole 'church'; Protestants reject the Roman Catholic doctrine that <blank>is the one true church.	it	Bad
Fighting began on 3 November 1839, when <blank>, the Royal Saxon, attempted to sail to Guangdong.	a second British ship	Bad
<blank>, the slotting machine and the shaping machine were developed in the first decades of the 19th century.	the planning machine	Good
On the same ship were several other Dutch travellers, including Elias Hesse, who would be called <blank>nowadays.	a travel writer	Good
His role, like that of many of the Norse gods, is <blank>.	complex	Bad
On 20 March, NISA announced that <blank>had been returned to a condition of cold shutdown.	both reactors	Good
In June 2010, UK aid group Oxfam reported a dramatic increase in the number of rapes occurring in the Democratic Republic of <blank>.	the Congo	Good
<blank>is described separately below.	The driving mechanism behind this movement	Bad
It was <blank>.	a monumental feat for the 'Mongols' (as they became known collectively)	Bad
Also of importance to <blank>was the creation of the Masnavi, a collection of mystical poetry by the 13th-century Persian poet Rumi.	Sufism	Good

items and were asked to assign the label *Good* if the generated item asked a key concept from the sentence and was reasonable to answer; *Okay* if the generated item asked a key concept but was difficult to answer due to length and ambiguity; and *Bad* if the generated item asked an unimportant concept. Becker et al. (2012) employed 85 crowdsource workers, and each generated item was rated by four crowdsource workers. To enhance the quality of ratings assigned, Becker et al. (2012) identified the poor-performing raters by comparing the mean distance of ratings assigned by the workers, and those with two standard deviations above the mean distance were removed. This allowed Becker et al. (2012) control for rater bias (Snow et al., 2008; Wiebe et al., 1999).

A second quality control step was introduced to control for the disagreement among the raters by having each generated question rated by four crowdsource workers. Specifically, only items where at least three raters assigned the same rating were kept for subsequent analysis. Thus, generated items where two raters disagreed with the rest were removed to enhance the quality of ratings assigned to each item. This yielded a total of 1825 generated questions. In addition to these data cleaning steps, we followed a similar approach to Becker et al. (2012) and collapsed the rating categories of *Okay* and *Bad* within a single label of *Bad* showing poorer quality items. This allowed us to differentiate poor-quality items from high-quality items according to crowdsource ratings. The final dataset included 1102 *Bad* items and 723 *Good* items.

In this study, we refer to items where a portion of the sentence is masked as the item stem and the answer that belongs to the masked portion of the item as the keyed response. Below, we describe the analytic plan followed for each study.

3.2 Analytic Plan

3.2.1 Study 1: Classifier Training with Feature Extraction

In Study 1, we trained three machine learning (ML) classifiers to predict item quality labels assigned by crowdsource workers. We first extracted linguistic features from cloze item stems and keyed responses. Note that one major challenge was that cloze items are short and do not contain some other auxiliary information such as the availability of distractors or passages, rendering the feature extraction process highly difficult. We should also highlight that the way items are generated has a huge influence on the quality criteria that need to be considered during item evaluation. In that regard, because items are generated by identifying part of the sentences that can be masked for stem generation, issues of grammaticality, fluency, and sensibleness are less of a concern for the dataset we used in this study. Below, we first explain the feature extraction and ML classifier training processes in detail.

Feature Extraction. Extracting features for cloze items was challenging because cloze items were shorter and items were composed of only a stem and keyed response. We reviewed studies focusing on predicting item statistics and characteristics (Ha et al., 2019; Yaneva & Von Davier, 2023; Yaneva et al., 2020) to identify features that would help us predict quality labels assigned by crowdsource workers. Nonetheless, only a limited number of features could be used with the current dataset as items were structurally different than previous studies focusing on item statistics prediction. By these features, we tried to reflect item quality from a linguistic point of view and extracted features reflection cohesion, connectivity, similarity, and sentence complexity. We provided an overview of features we extracted for ML classifier training in Table 3.2.

We first extracted interpretable textual features such as connectivity, cohesion, and text length using Coh-Metrix (Graesser et al., 2011; McNamara et al., 2014). Coh-

Table 3.2: Features Extracted for Training Machine Learning Models

Features	Resources
Descriptives	Coh-Metrix
Text Easability Principal Component Scores	Coh-Metrix
Lexical Diversity	Coh-Metrix
Connectives	Coh-Metrix
Situation Model	Coh-Metrix
Syntactic Complexity	Coh-Metrix
Syntactic Pattern Density	Coh-Metrix
Word Information	Coh-Metrix
Readability	Coh-Metrix
Cosine similarity between the predicted response and keyed response	BERT & RoBERTa
Cosine similarity between item stem and keyed response	BERT
Constituency parse tree depth	NLTK

Metrix calculates various cohesion and coherence metrics for a given text. To extract Coh-Metrix features, we only used item stems (i.e., generated questions where a part was masked during the item generation process) because using keyed responses with item stems would have yielded human written forms of the items. We extracted 108 linguistic and discourse representations of the stems, yet we did not use all of the Coh-Metrix features because of the lack of variability in the features. Specifically, features related to paragraph count were not usable because of the brevity of the item stems. The list of indices that Coh-Metrix calculates is given in Table 3.2. The final set of the Coh-Metrix indices we used in classifier training was as follows:

- **Descriptives:** The number of words in a stem, the mean and standard deviation of the number of words, the mean and standard deviation of the number of syllables, and the mean and standard deviation of the number of letters.
- **Text Easability Principal Component Scores:** z-scores and percentiles of text easability principal components of narrativity, syntactic simplicity, word

concreteness, referential cohesion, deep cohesion, verb cohesion, connectivity, and temporality.

- **Lexical Diversity:** Lexical diversity, type-token ratio, content word lemmas, lexical diversity, type-token ratio, all words, lexical diversity, MTLD, all words, and lexical diversity, VOCD, all words.

- **Connectives:** All connectives incidence, causal connectives incidence, logical connectives incidence, adversative and contrastive connectives incidence, temporal connectives incidence, expanded temporal connectives incidence, additive connectives incidence, positive connectives incidence, and negative connectives incidence.

- **Situation model:** Causal verb incidence, causal verbs and causal particles incidence, the ratio of intentional particles to intentional verbs, intentional verbs incidence, ratio of casual particles to causal verbs, LSA verb overlap, and WordNet verb overlap.

- **Syntactic Complexity:** The mean of the number of modifiers per noun phrase.

- **Syntactic Pattern Density:** Noun phrase density, verb phrase density, adverbial phrase density, preposition phrase density, agentless passive voice density, negation density, gerund density, and infinitive density.

- **Word Information:** Noun incidence, verb incidence, adjective incidence, adverb incidence, pronoun incidence, the mean of CELEX word frequency for content words, CELEX Log frequency for all words, CELEX Log minimum frequency for content words, age of acquisition for content words, familiarity for content words, concreteness for content words, imagability for content words, meaningfulness, Colorado norms, content words, polysemy for content words, hypernymy for nouns, hypernymy for verbs, and hypernymy for nouns and verbs.

- **Readability:** Flesch reading ease, Flesch-Kincaid grade level, and Coh-Metrix L2 readability.

Next, to quantify the difficulty of answering the masked part of the items, we employed a pre-trained large language model, RoBERTa, to predict the masked portion of the item. RoBERTa intended to replicate Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) pre-training while carefully examining each hyperparameter and training data size to create a robust language model that may match or exceed the performance of BERT (Liu et al., 2019). Using RoBERTa, we predicted the masked portion of each item stems. During masked portion prediction, RoBERTa provides a top k tokens with probability scores for the masked portion of a sentence. We picked the most likely token with the highest probability score and then calculated the cosine similarity between the predicted token and the actual keyed response. Cosine similarity is a frequently used metric to quantify the extent of similarity between two sentences. The cosine similarity is the dot product of the angle between the embedding vectors of each sentence. Typically, cosine similarity ranges between 0 and 1, and values closer to 1 suggest more similarity between two sentences. We used the *BERT-base-uncased* model to obtain sentence embeddings for predicted tokens and keyed responses. The cosine similarity metric we obtained between predicted tokens and keyed responses was another feature we used for training our ML classifiers.

The next feature we extracted was again based on cosine similarity using item stems and keyed responses. We obtained sentence embeddings for item stems and keyed responses using *BERT-base-uncased*. Then, we calculated the cosine similarity metric between the vector of embeddings obtained using item stems and keyed responses.

The final feature we extracted was the parse tree depth using the Natural Language Toolkit (NLTK; Bird et al. (2009)) library. Using the constituency parsing method, we parsed item stems and created parse trees for each item stem. An example of a parse tree is as follows: "((ROOT (S (NP (NP (JJ Contemporary) (JJ transnational)

(NNS interactions)) (PP (IN between) (NP (NP (NP (NNP Hmong)) (PP (IN in) (NP (DT the) (NNP West)))) (CC and) (NP (NP (NNP Miao) (NNS groups)) (PP (IN in) (NP (NNP China))))))) (, ,) (VP (VBG following) (NP (DT the) (CD 1975) (NNP Hmong) (NN diaspora))) (, ,) (VP (VBP have) (PP (IN to) (NP (NP (DT the) (NN development)) (PP (IN of) (NP (NP (DT a) (JJ global) (JJ Hmong) (NN identity)) (SBAR (WHNP (WDT that)) (S (VP (VBZ includes) (NP (NP (NP (ADJP (RB linguistically) (CC and) (RB culturally) (JJ related)) (NNS minorities)) (PP (IN in) (NP (NNP China)))) (SBAR (WHNP (WDT that)) (S (ADVP (RB previously)) (VP (VBD had) (NP (DT no) (JJ ethnic) (NN affiliation))))))))))))) (. .)))". We then calculated the parse tree depth. This process generated a feature vector composed of integer values indicating the complexity of the syntactic structure of each item stem. The code is available for feature extraction in Appendix B.1.

Classifier Training. We used three frequently used ML classifiers for educational datasets (Domladovac, 2021): Random Forest (Breiman, 2001), Support Vector Machine (Hearst et al., 1998), and Logistic Regression (Wright, 1995). These classifiers were selected because they are relatively transparent approaches and performed well with educational datasets with smaller sample sizes (Demmans Epp & Phirangee, 2019; Gorgun et al., 2022; Romero & Ventura, 2007). We used 5-fold nested cross-validation to select the best set of hyperparameters through a randomized search method. The list of hyperparameters for each classifier, as well as the search space for hyperparameter tuning, is given in Table 3.3.

For random forest, the best model performance was achieved with the number of estimators = 94, the maximum number of features = "auto", the maximum depth of trees = 4, the minimum number of samples leaf = 2, and criterion = entropy. For the support vector machine, the best set of hyperparameters was C = 1, and kernel = "sigmoid". Finally, the best set of hyperparameters for the logistic regression model was C = 1, penalty = "none", and solver = "liblinear". We examined model

Table 3.3: Hyperparameters Search Space for Classifier Tuning

ML Classifier	Hyperparameters	Values
Random Forest	Number of estimators	[10:200]
	Max features	['auto', 'sqrt']
	Max depth	[2, 4]
	Min samples split	[2, 3, 4]
	Min samples leaf	[2, 3]
	Criterion	['gini', 'entropy']
Support Vector Machine	Kernel	['linear', 'poly', 'rbf', 'sigmoid']
	C	[100, 10, 1, .01, .001]
Logistic Regression	Solver	['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga']
	Penalty	['none', 'l1', 'l2', 'elasticnet']
	C	[100, 10, 1, .01, .001]

performance both on the training and test sets to diagnose overfitting issues. In all models, the model accuracy was higher on the training set, suggesting the absence of overfitting. On the training set, the model accuracy values for the random forest, support vector machine, and logistic regression were 68%, 62%, and 66%, respectively. The analyses were conducted in Python (Version 3.9.12) (Van Rossum & Drake, 2009). The code is available for feature extraction in Appendix B.2.

3.2.2 Study 2: Fine-Tuning a Pre-Trained Large-Language Model

For fine-tuning a pre-trained large-language model, we employed a frequently used LLM in education tasks for text classification (Shen et al., 2021), automated scoring (Beseiso & Alzahrani, 2020), and process mining (Scarlatos et al., 2022), namely Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019). BERT relies on a multi-layer bidirectional self-attention mechanism (Vaswani et al., 2017) and has two variants: BERT-base with 110 million parameters, 12 attention heads, 768 dimensions, and 12 transformer layers, and BERT-large with 340 million

parameters, 16 attention heads, 1024 dimensions, and 24 transformer layers. Thanks to adopting a *bidirectional* self-attention mechanism, unlike GPT, which only has a one-directional self-attention, BERT can attend to context to both its left and right (Devlin et al., 2019), allowing it to learn contextual information from either direction and capturing the relationships among the tokens from both directions.

BERT has an encoder-only architecture, and fine-tuning a BERT model allows researchers to re-train BERT (i.e., adjust the weights of the parameters of a pre-trained BERT model) for downstream tasks such as text classification, question-answering, sentiment analysis, and text summarization. The first layers of BERT contain generic language representation and during the fine-tuning process, we intend to adjust parameter weights by adding a linear output layer. The process starts by tokenizing the input text to be fed to the BERT model. At this stage, the maximum number of tokens, whether paddings will be used, and whether sentences with longer sequences will be truncated are indicated. BERT accepts a maximum of 512 tokens (Devlin et al., 2019). Padding is used to adjust the length of sentences with the number of tokens smaller than the indicated limit. For instance, if the maximum length of sentences is limited to 250 tokens, then the sentences with a fewer number of tokens will be padded (i.e., [PAD]) to adjust the length of the sentences. On the other hand, when truncation is activated, the sentences with longer sequences will be trimmed to match the number of tokens. During training, a [CLS] token is added at the beginning of every sentence, and a [SEP] token is added at the end of every sentence. The input goes through 12 transformer layers for BERT-base or 24 transformer layers for BERT-large. Each layer includes a list of token embeddings (768 for BERT-base and 1024 for BERT-large) and generates the same number of embeddings on the output. As embeddings go through each layer, the parameter weights are adjusted. The [CLS] token at the last layer includes pooled information about the embeddings and is used by the classifier. Figure 3.2 depicts the process of fine-tuning a BERT model for downstream tasks.

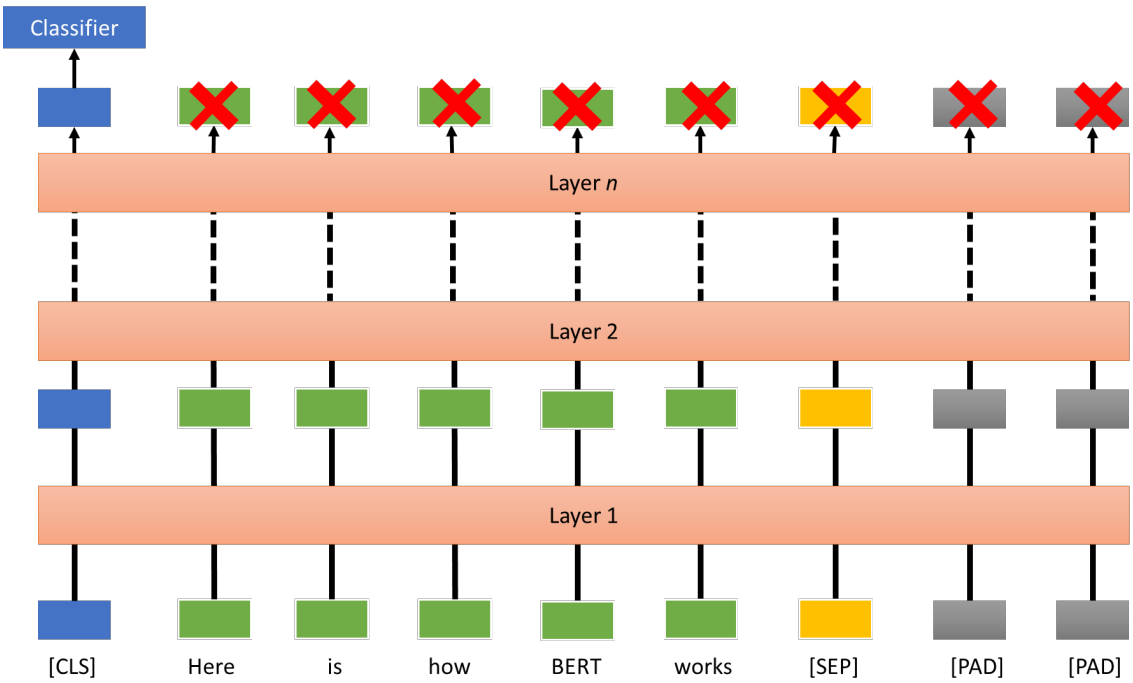


Figure 3.2: Fine-tuning BERT Flowchart

We fine-tuned both BERT-base and BERT-large. During the preprocessing stage, we lower-cased each generated question. We split the data into training and set sets, and 20% of the data ($n = 363$) was used as the holdout test set. We tokenized the generated items using *bert-large-uncased* and *bert-base-uncased* and obtained embeddings, including input IDs and token-type IDs. We set the maximum token length to 250 and used both truncation and padding. An example item stem after tokenization was applied is as follows: "[CLS] the chapter house was originally used in by benedictine monks for daily meeting. [SEP] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD]". Then using *AutoModelForSequenceClassification* and *bert-large-uncased* or *bert-base-uncased*, we fine-tuned the model using the hyperparameters learning rate $= 2e-5$, training batch size $= 8$, evaluation batch size $= 16$, epochs $= 10$, and weight decay $= 0.01$. The process and hyperparameters used for fine-tuning both BERT models were exactly the same to facilitate the comparison between the two models. The analyses were conducted in Google Colab with V100 GPU. Fine-tuning BERT-Base and BERT-Large

took approximately 3 and 10 minutes, respectively, in Google Colab using GPU. The code is available for feature extraction in Appendix B.3.

3.2.3 Study 3: Instruction-Tuning a Generative Large-Language Model

For instruction-tuning a generative LLM, we used an open-source publicly available LLM, Llama-2 (Touvron et al., 2023) with 7-billion parameters (i.e., Llama 2-7B). Figure 3.3 shows an overview of the instruction-tuning process of the Llama 2-7B model. Llama 2 is a generative LLM pre-trained using publicly available data and includes versions with 7-billion, 13-billion, and 70-billion parameters (Touvron et al., 2023). For instruction-tuning Llama 2 we selected the smallest model with 7-billion parameters. Instruction-tuning is similar to fine-tuning an LLM where the weights of parameters are adjusted for a specific task. However, in instruction-tuning, an instruction is provided along with input and output pairs, and models are re-trained for the specific task.

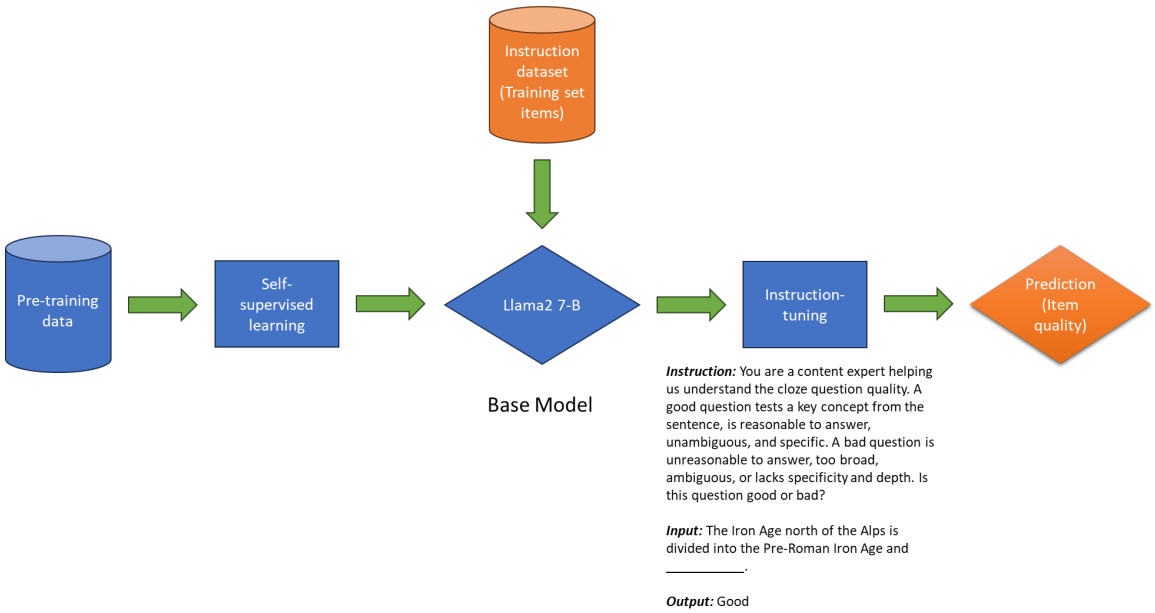


Figure 3.3: Instruction-Tuning Llama 2-7B Flowchart

We started by deploying the cleaned dataset to the Hugging Face platform. In the

subsequent stage, we experimented with several prompts to identify the best prompt that would yield higher model accuracy. To craft our prompts, we relied on the quality criteria that Becker et al. (2012) employed with crowdsource workers. The first prompt we tried was completely based on the definition that Becker et al. (2012) provided to crowdsource workers and was as follows: *Begin your response with yes or no. Does the blank in the next sentence assess key concepts from the sentence and would be reasonable to answer.* However, this yielded suboptimal results. We then provided the definition of a cloze question in the prompt as *We removed a part of the following sentence to construct a cloze question. We are trying to understand whether the following question is a good or bad question. A Good question is one that tests key concepts from the sentence and would be reasonable to answer. A Bad question is one that asks about an unimportant aspect of the sentence or has an uninteresting answer that can be figured out from the context of the sentence. Is this question a good or bad question?.* Similar to the first prompt we tried, this was also not feasible for differentiating a good question from a bad one. A substantial improvement in model accuracy was obtained when we used more descriptor words to specify what a good or bad question is. The final prompt selected was as follows: *You are a content expert helping us understand the cloze question quality. A good question tests a key concept from the sentence, is reasonable to answer, unambiguous, and specific. A bad question is unreasonable to answer, too broad, ambiguous, or lacks specificity and depth. Is this question good or bad?.* During this stage, we ran Llama-2 with an API to tweak the prompt and get an instant response from the model. We also randomly selected a few cloze items from the dataset for prompt tweaking. For a complete list of prompts, please see Appendix A.

We used 20% ($n = 363$) of questions as the holdout test data. Using Llama 2-7B, we supervised fine-tuned (i.e., instruction-tuned) the model with the remaining 80% ($n = 1462$) of the data. Because supervised fine-tuning a model with 7 billion parameters is computationally intensive, we employed parameter-efficient fine-tuning (PEFT)

(Liu et al., 2022) and efficient fine-tuning of quantized LLMs (QLoRA) (Dettmers et al., 2023) to reduce memory usage and computational costs. The former allowed us to identify the most important parameters for tuning an LLM. PEFT updates the weights of the most important parameters for the task that the model is being re-trained on, minimizing the computational costs required for the supervised fine-tuning of an LLM (Liu et al., 2022). On the other hand, QLoRA allowed us to reduce memory use by storing the trained models in a compressed 4-bit format rather than an original 32-bit format (Dettmers et al., 2023). Leveraging these two methods, we supervised fine-tuned (i.e., instruction-tuned) Llama 2-7B for approximately 4 hours.

We loaded the Llama 2-7B model to a Python-based environment for the downstream task of item evaluation using the function *AutoModelCausalLM*. We set the hyperparameters as follows: maximum sequence length = 512, learning rate = $2e-4$, max steps = 1000, training batch size = 4, and optimizer = *adam*. The supervised fine-tuning was conducted in Google Colab with V100 GPU. The code is available for Llama 2-7B instruction tuning in Appendix B.4.

3.3 Model Evaluation

We evaluated the performance of all models developed in Studies 1, 2, and 3 using the same set of performance metrics. We compared the predicted label against the ground truth assigned by the crowdsource workers. Specifically, we considered a 2X2 confusion matrix where predicted labels and ground truth are compared in a pairwise fashion, and true positive (TP), true negative (TN), false positive (FP), and false negative (FN) rates are derived. A better-performing model has higher rates of true negatives and true positives than false positives and false negatives. Thus, a better model has a higher number of diagonal elements than off-diagonal elements, indicating that the model has done a good job in terms of identifying the true labels of cloze questions rather than misclassifying them. In Figure 3.4, we demonstrated a 2X2 confusion matrix and indicated regions where one can find TP, TN, FP, and FN

Table 3.4: Confusion Matrix for Evaluating the Developed NLP Models

		Predicted	
		Good	Bad
Ground Truth	Good	TP	FN
	Bad	FP	TN

rates.

In addition to the confusion matrix, we calculated the number of true predictions over the total number of predictions, i.e., *accuracy* as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{3.1}$$

Another metric we used for quantifying the number of true predictions made correctly by the model, i.e., *precision*, is calculated as follows:

$$Precision = \frac{TP}{TP + FP}. \tag{3.2}$$

Recall or *Sensitivity* allowed us to quantify the number of correctly predicted true cases over the total number of true positives in the dataset and is calculated as

$$Recall = \frac{TP}{TP + FN}. \tag{3.3}$$

We also used *Specificity* to estimate the number of correctly predicted negative cases over the total number of negative cases in the dataset as follows:

$$Specificity = \frac{TN}{TN + FP}. \tag{3.4}$$

A metric combining both recall and precision is the F1-score and is computed as

$$F1 = 2\frac{Precision * Recall}{Precision + Recall}, \tag{3.5}$$

which generates a harmonic mean of precision and recall. It is especially useful when the goal of the prediction model is to establish a balance between precision and recall. The final metric we used to evaluate the performance of the developed models

is the receiver operating characteristic (ROC) curve which plots the true positive rate (TPR) defined as

$$TPR = \frac{TP}{TP + FN}, \tag{3.6}$$

against the false positive rate (FPR), defined as

$$FPR = \frac{FP}{FP + TN}. \tag{3.7}$$

Accuracy, precision, recall, specificity, F1- score, and ROC curve generate a value between 0 and 1, and values closer to 1 indicate better model performance.

3.4 Chapter Summary

In this chapter, we discussed three natural language processing methods we developed for evaluating item quality automatically. We first explained the data used for the study as well as the data cleaning process. We described the item generation steps and quality labeling process developed by Becker et al. (2012). Then, we defined data preparation stages for developing natural language processing methods. Specifically, we discussed the feature extraction process and classifier training steps for Study 1, the fine-tuning process of BERT-Base and BERT-Large for Study 2, and the instruction-tuning process of Llama 2-7B for Study 3. All models are evaluated using accuracy, precision, recall, specificity, and F1-score. We also use confusion matrix and ROC curves to compare and contrast model performance across the three natural language processing methods.

Chapter 4

Results

4.1 Descriptive Statistics of Generated Cloze Items

In this section, we provided an overview of cloze item characteristics to compare surface-level characteristics of cloze items rated as good and bad by the crowdsource workers. Of 1825 generated cloze items retained in the dataset after the data cleaning process described in Section 3.1, 60% of items were rated as bad by crowdsource workers. Although the majority category is composed of bad items and the dataset is not perfectly balanced, we still can assert that class imbalance is less of a concern in this dataset. However, considering this class distribution is important when we evaluate the performance of each NLP method for predicting item quality. We show the distribution of the number of words in good and bad item stems separately in Figure 4.1.

While Figure 4.1 demonstrates more variability for items labeled as bad compared to good items, we can also observe that the distribution of the number of words mostly perfectly overlaps for good and bad items. Thus, we can expect that the number of words or sentence length might be a less powerful feature differentiating good items from bad ones. While the distributions of the number of words look similar for good and bad items, we also see that bad items have some extreme values with sentence lengths of more than 80 words. The average sentence length (i.e., calculated as the number of words in a sentence) for bad items was 20.84 with a standard deviation

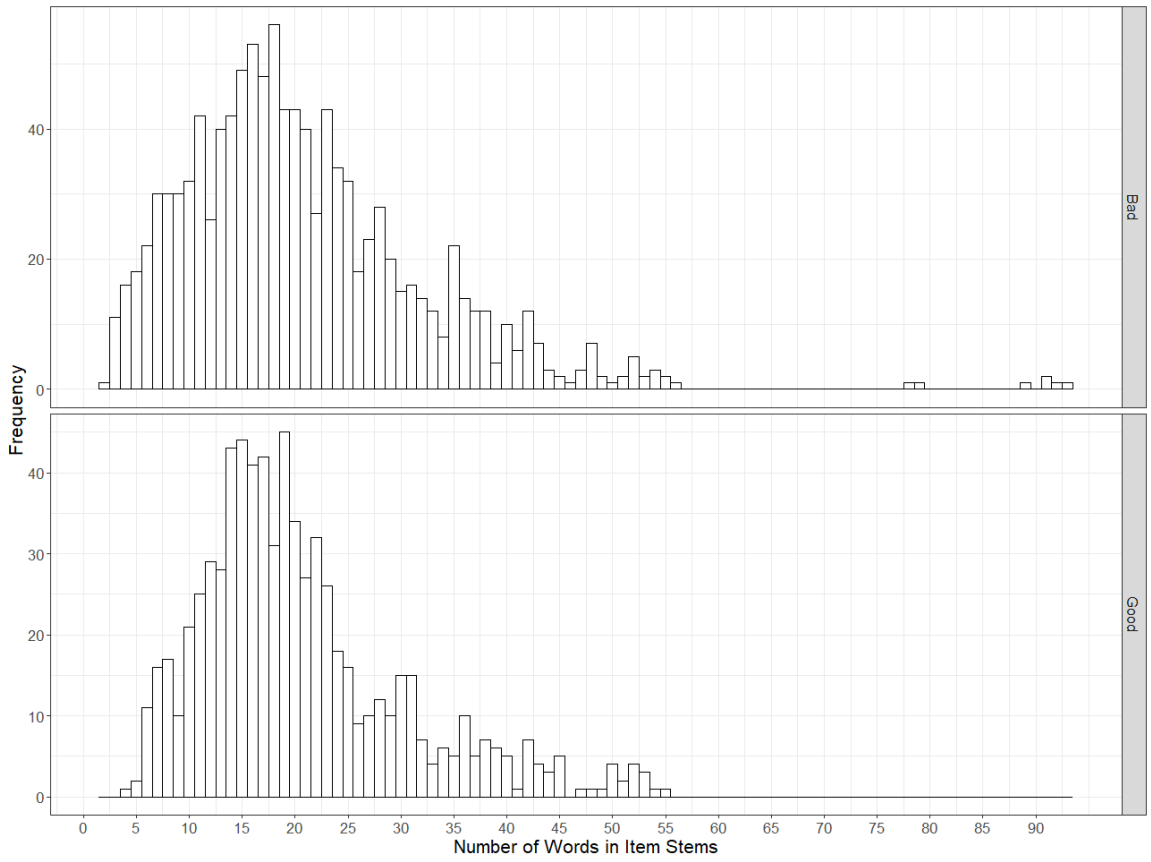


Figure 4.1: The Distribution of the Number of Words in Item Stems

of 11.86. The minimum sentence length was 2, and the maximum sentence length was 93. Whereas, for good items, the average sentence length was 20.54 with a standard deviation of 9.84. The minimum and maximum sentence lengths were 4 and 55, respectively. These results also underscored that bad items included some longer sentences than good items, and the variance was slightly larger for bad items compared to good ones. While mean may not be a feasible feature for differentiating item quality, standard deviation could be useful for distinguishing bad items from good ones.

We also analyzed the distribution of the number of words in keyed responses (see Figure 4.2). A similar trend to item stems has emerged for keyed responses. That is, the distributions of the number of words for bad and good items were similar yet bad items had more extreme values or keyed responses with more words. The average

keyed response length for bad items was 4.36, with a standard deviation of 5.16. The minimum and maximum keyed response lengths were 1 and 44 words, respectively. Whereas, the average keyed response length for good items was 2.17 with a standard deviation of 1.53. The minimum keyed response length was 1, and the maximum keyed response length was 25 for good items. These results indicated that good items had less variability in terms of response length and might suggest that high-quality cloze items should have a shorter masked portion to be reasonable to be answered by the examinees.

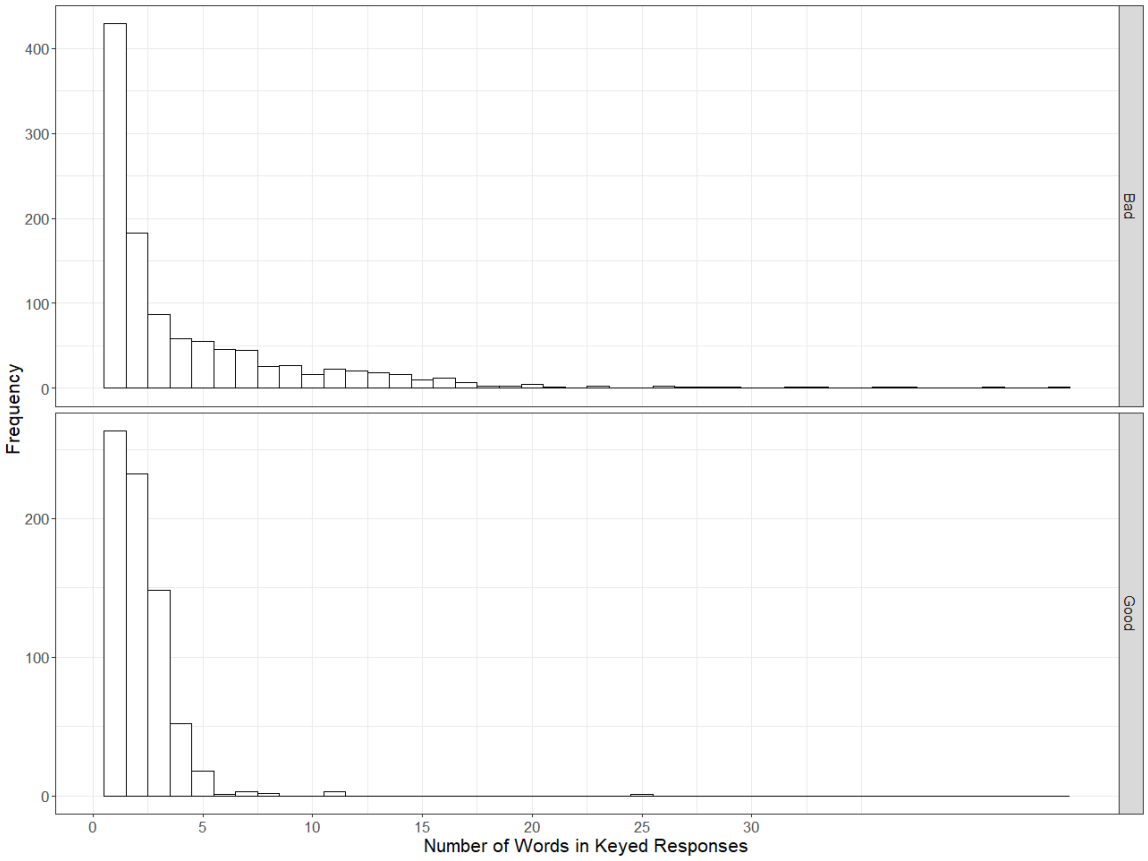


Figure 4.2: The Distribution of the Number of Words in Keyed Response

4.2 Study 1 Results

In Table 4.1, we provided evaluation metrics (i.e., accuracy, precision, recall, and F1-score) for the random forest, support vector machine, and logistic regression on

the holdout test set using the best set of hyperparameters selected with 5-fold nested cross-validation.

Table 4.1: Evaluation Metrics for Machine Learning Classifiers

	Random Forest				Support Vector Machine				Logistic Regression			
	A	P	R	F1	A	P	R	F1	A	P	R	F1
Overall	57%	.70	.54	.44	53%	.44	.50	.36	61%	.62	.59	.56
Good	-	.84	.10	.17	-	.33	.01	.02	-	.65	.32	.43
Bad	-	.56	.98	.56	-	.54	.98	.69	-	.60	.86	.70

Note. A: Accuracy. P: Precision. R: Recall.

The accuracy of the zero-rule classifier for the test set would be 54%. The zero-rule classifier can be considered as the baseline model for comparing classifier performance. Table 4.1 suggests that the support vector machine performs similarly to the zero-rule classifier, whereas random forest and logistic regression slightly outperform the baseline model of the zero-rule classifier. To test whether there are statistically significant differences between the training classifiers, we conducted Cochran's Q test (Raschka, 2018). Cochran's Q test can be used to compare whether there are any statistically significant differences in accuracies of more than two classifiers. We found that there were statistically significant differences among trained classifiers for predicting item quality, $Q = 12.767$, $p < .01$. Cochran's Q analysis is an omnibus test and does not provide pairwise comparisons. To conduct pairwise comparisons between trained classifiers, we used McNemar's test (Raschka, 2018). We found statistically significant differences between random forest and support vector machine, McNemar's $\chi^2 = 9.783$, $p < .01$, and support vector machine and logistic regression, McNemar's $\chi^2 = 8.783$, $p < .01$. However, there was no statistically significant difference between logistic regression and random forest, McNemar's $\chi^2 = 2.182$, $p = .140$.

The random forest classifier achieved 57% accuracy on the test set with precision, recall, specificity, and F1-score of .70, .54, .95, and .44, respectively, on the overall

model. The performance metrics of precision, recall, and F1-score of random forest on good items were .84, .10, and .17. On the other hand, the performance of random forest for bad items was as follows: precision = .56, recall = .98, and F1-score = .56.

The support vector machine classifier achieved 53% on the holdout test set. For the overall model, precision, recall, specificity, and F1-score were .44, .50, .98, and .36, respectively. The performance metrics of precision, recall, and F1-score of the support vector machine on good items were .33, .01, and .02. On the other hand, the performance of the support vector machine for bad items was as follows: precision = .54, recall = .98, and F1-score = .69.

Finally, logistic regression achieved 61% accuracy on the test set. For the overall model, precision, recall, specificity, and F1-score were .62, .59, .86, and .43, respectively. The performance metrics of precision, recall, and F1-score of logistic regression on good items were .33, .01, and .02. On the other hand, the performance of logistic regression for bad items using the performance metrics of precision, recall, and F1-score were .60, .86, and .70, respectively.

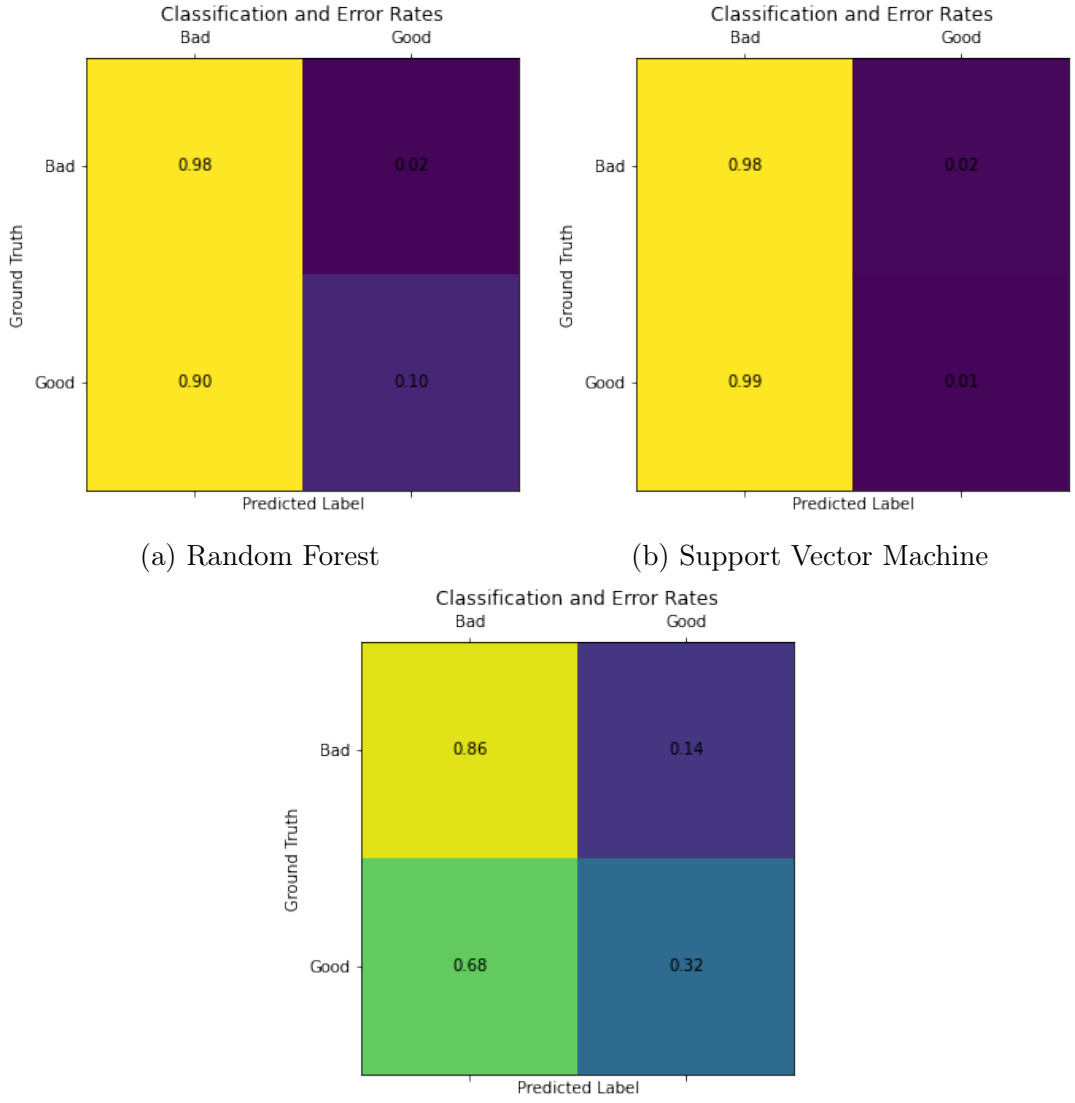
Table 4.2: Confusion Matrix for Machine Learning Classifiers

		Predicted					
		RF		SVM		LR	
		Good	Bad	Good	Bad	Good	Bad
Ground Truth	Good	16	152	2	166	53	115
	Bad	3	194	4	193	28	169

Note. RF: Random Forest. SVM: Support Vector Machine. LR: Logistic Regression.

To better understand classifier performance on the test set, we analyzed the confusion matrix (Table 4.2) and (mis)classification rates (Figure 4.3). On the confusion matrix table, we want diagonal values to be larger than the values on the reverse diagonal. Concerning Figure 4.3 showing (mis)classification rates, the diagonal values with lighter colors (i.e., light green and yellow) indicate higher accuracy whereas reverse diagonal values with darker colors (i.e., dark blue and black) indicate lower

misclassification rates. In other words, diagonal values with lighter colors, and reverse diagonal values with darker colors illustrate better classifier performance.



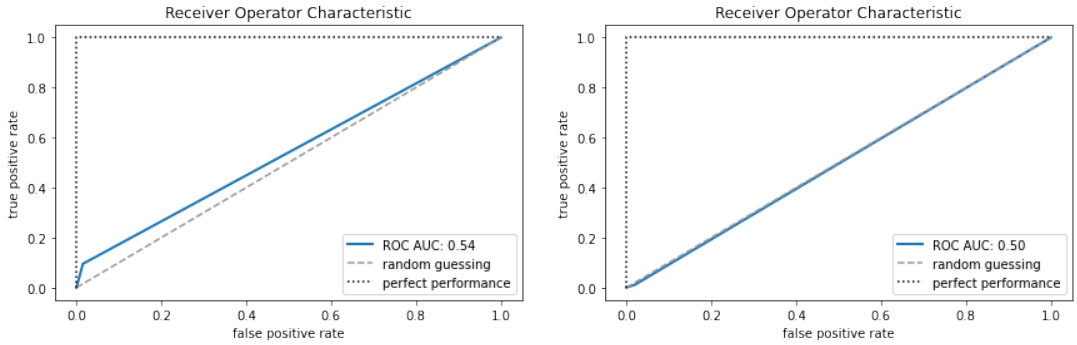
(a) Random Forest

(b) Support Vector Machine

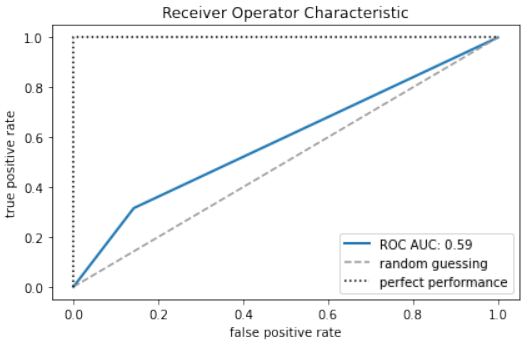
(c) Logistic Regression

Figure 4.3: (Mis)Classification Rates for Trained ML Classifiers

Random forest performed quite well for predicting bad items, however, the majority of good items were misclassified as bad items suggesting that the random forest model actually learned that the majority category is bad (Table 4.2. Only 3 bad items were misclassified as good. However, random forest performed quite poorly in accurately predicting good items. Only 16 items were correctly classified as good. The poor



(a) Random Forest (b) Support Vector Machine



(c) Logistic Regression

Figure 4.4: Receiver Operating Characteristic Curves for Trained ML Classifiers

performance of random forest for classifying good items is evident in Figure 4.3 where we can see that the majority of both bad and good items were classified as bad. While only 2% of bad items were misclassified, 90% of good items were misclassified as bad (Figure 4.3). Thus, this is not good news for the overall performance of random forest predicting item quality.

A similar trend was observed for the support vector machine. In fact, the support vector machine classifier performed worse than the random forest, in which only 1% of good items were correctly classified. While 98% of bad items were correctly classified as bad, this does not show acceptable performance because of the classifier's lack of ability to distinguish bad items from good items. Overall, both confusion matrix (Figure 4.2) and misclassification rates (Figure 4.3) demonstrate that the support vector machine is a useless model for predicting item quality. That is, in practice, all items generated could end up labeled as bad, suggesting the absence of classifier

learning taking place using the set of features we created.

Both Table 4.2 and Figure 4.3 indicate that logistic regression is the best classifier among the three trained classifiers for minimizing the misclassification rate for good items. Of 168 good items, 53 of them were correctly classified as good (32%) by the logistic regression classifier, which is a significant increase from 10% for random forest and 1% for support vector machine. As a matter of fact, in Figure 4.3 for logistic regression, we can see an improvement in model performance as the diagonal values have lighter colors while reverse diagonal values have darker colors compared to random forest and support vector machine where we can see lighter colors allocated only at the left-hand side of the figure. We can also observe that the misclassification rate increased slightly for bad items, suggesting that the logistic regression model learned to map item quality on the item features extracted to a certain degree.

Finally, we also analyzed the receiver operator characteristic (ROC) curves of random forest, support vector machine, and logistic regression (Figure 4.4. The curves closer to the upper left corner indicate better classification accuracy because the false positive rate would be zero, whereas sensitivity and specificity would be 1. Here, the best model, also supported by Table 4.2 and Figure 4.3 is logistic regression which has an ROC curve closer to the upper left corner, followed by the random forest. Support vector machine, on the other hand, had the worst performance because the ROC curve overlaps with the random guessing line, underscoring that the model did not learn to differentiate good items from bad ones using the set of features we extracted from the items.

4.2.1 Error Analysis

To better understand classification errors, we did an error analysis on the misclassified subsample of the test set for all three ML classifiers. While error analysis might not be informative for the support vector machine model as almost all good items were misclassified as bad, thus the model did not predict item quality properly, we reported

trends we observed for the misclassified items for all trained ML classifiers.

Concerning the random forest classifier, some of the misclassified good items included blanks placed at the end of item stems (e.g., "Income inequality in the United States started to rise in the late 1970s, however, the rate of increase rose sharply in the 21st century; it has now reached a level comparable with that found in <blank>."). Additionally, shorter item stems and items that include keyed responses composed of jargon or a specific terminology (e.g., item stem: "Most sedimentary rocks contain either quartz (<blank> rocks) or calcite (especially carbonate rocks).; keyed response: "especially siliciclastic") tended to be misclassified by the random forest classifier. Additionally, when the keyed response included a proper noun, location, or year (e.g., item stem: "In 1461 <blank> established the Armenian Patriarchate of Constantinople.", keyed response: "Sultan Mehmed II"), the items tended to be misclassified. Finally, complex sentences that are hard to parse were misclassified by the random forest classifier.

For the support vector machine, the items with shorter keyed answers, blanks created at the end of item stems, and shorter sentences were misclassified by the support vector machine. Nonetheless, note that the support vector machine essentially learned that the majority category was bad items and classified almost exclusively all items as bad. For that reason, the misclassified batch in the test set was mainly composed of good items, thus it is more difficult to make inferences about the classifier behavior when categorizing items as good and bad.

Finally, for logistic regression, we observed a similar trend to the random forest classifier concerning the misclassified items. Item stems including blank placed at the end of sentences were misclassified. Additionally, extremely long (e.g., "The Empire had reached the end of its ability to effectively conduct an assertive, expansionist policy against its European rivals and <blank> was to be forced from this point to adopt an essentially defensive strategy within this theatre.") or extremely short (e.g., "<blank> were developed in the first decades of the 19th century.") item stems

tended to be misclassified by the logistic regression classifier. Similar to the random forest, blanks that included proper nouns, locations, years, and obvious answers were misclassified by the logistic regression model.

4.2.2 Feature Importance

Although the ML classifiers we trained performed far from being acceptable for identifying item quality, here we report the ten most important features for each classifier which may inform feature studies on predicting item quality (Table 4.3). For random forest, the most important five features were the cosine similarity between stem and keyed response, the cosine similarity between the predicted response and keyed response, causal verbs and causal particles incidence, causal verb incidence, and the mean of hypernymy for words. For the support vector machine, the most important five features were the cosine similarity between the stem and keyed response, the cosine similarity between the predicted response and keyed response, lexical diversity, the type-token ratio of all words, the standard deviation of the number of words, and z-score of text easability of principal scores referential cohesion. Finally, for logistic regression, the most important five features were the cosine similarity between the stem and keyed response, the cosine similarity between the predicted response and keyed response, the standard deviation of the number of words, lexical diversity, the type-token ratio of all words, and z-score of text easability principal component scores referential cohesion.

Across all classifiers, most of the frequently used features belonged to Coh-Metrix categories (see Table 3.2) of word information, text easability principal component scores, situation model, and descriptives. Among the most important ten features, neither Coh-Metrix features from syntactic complexity nor syntactic pattern diversity have been used for training the random forest, support vector machine, or logistic regression classifiers. For the random forest, the most features in the top ten list came from word information and text easability principal component scores. For the sup-

Table 4.3: The Ten Most Important Features for the Trained Classifiers

	Random Forest	Support Vector Machine	Logistic Regression
1	Cosine similarity between stem and keyed response	Cosine similarity between stem and keyed response	Cosine similarity between stem and keyed response
2	Cosine similarity between predicted response and keyed response	Cosine similarity between predicted response and keyed response	Cosine similarity between predicted response and keyed response
3	Causal verbs and causal particles incidence	Lexical diversity, type-token ratio, all words	The standard deviation of the number of words
4	Causal verb incidence	The standard deviation of the number of words	Lexical diversity, type-token ratio, all words
5	The mean of hypernymy for verbs	z-score of text easability principal component scores referential cohesion	z-score of text easability principal component scores referential cohesion
6	z-score of text easability principal component scores narrativity	The mean of hypernymy for nouns and verbs	Flesch-Kincaid grade level
7	z-score of text easability principal component scores deep cohesion	Ratio of intentional particles to intentional verbs	The standard deviation of the number of syllables
8	The mean of meaningfulness, Colorado norms, content words	The mean of hypernymy for verbs	The mean of hypernymy for nouns and verbs
9	The mean of familiarity for content words	z-score of text easability principal component connectivity	Ratio of intentional particles to intentional verbs
10	z-score of text easability principal component scores syntactic simplicity	Ratio of casual particles to causal verbs	The mean of hypernymy for verbs

Note. The features are ordered from the most important to the least important.

port vector machine, the most features in the top ten list belonged to situation model, word information, and text easability principal component scores. Finally, for the logistic regression, the most important features in the top ten list, the most important features belonged to descriptives and word information. Interestingly, readability emerged as one of the ten most important features for only the logistic regression classifier. Given that studies conceptualized readability scores as a potential predictor of item quality and difficulty (Ha & Yaneva, 2019; Štěpánek et al., 2023; Yaneva et al., 2020), in this study, readability scores were among the top ten features only for the logistic regression model.

As hypothesized in Section 4.1, the standard deviation of the number of words was one of the most important predictors for support vector machine and logistic regression. As we discussed, the distributions of the average number of words in good and bad items were very similar (i.e., $M_{Good} = 20.54$ and $M_{Bad} = 20.84$) however there were visible differences in terms of variability of the number of words in good and bad items (i.e., $SD_{Good} = 9.84$ and $SD_{Bad} = 11.86$), which corroborated the findings that features focusing on standard deviation were more meaningful for predicting item quality.

An interesting finding was that for all classifiers, the two most important features were the same, consistent with previous studies focusing on the similarity between distractors and keyed responses or the similarity between the stem and alternatives (Hsu et al., 2018). This suggests that future research may focus on the interplay among the keyed response, distractors, and the stem to gauge the quality of items generated. Furthermore, we analyzed the similarity between the RoBERTa predicted response and keyed response and it emerged as one of the most important features across the trained classifiers. To the best of our knowledge, to date, studies have not investigated the similarity between the LLM-predicted response and keyed response. Thus, this has emerged as a viable feature, implying that future studies may leverage the capacity of LLMs to investigate the relationship between alternatives, keyed

responses, and item stems.

4.3 Study 2 Results

In Study 2, we fine-tuned BERT-Large and BERT-Base models for predicting item quality. Of 365 items in the holdout test set, 143 items (39%) were rated as good by crowdsource workers. While both BERT-Large and BERT-Base performed similarly for predicting item quality, BERT-Base, which has 110 million parameters, seemed to slightly outperform BERT-Large with 340 million parameters.

Table 4.4: Evaluation Metrics for Fine-Tuned BERT

	BERT-Large				BERT-Base			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
Overall	64%	.62	.63	.62	66%	.66	.66	.66
Good	-	.54	.57	.55	-	.58	.58	.58
Bad	-	.71	.69	.70	-	.73	.73	.73

The overall model accuracy was for BERT-Large 64% and BERT-Base 66% (Table 4.4). Nonetheless, we conducted a statistical analysis to test whether the difference between BERT-Large and BERT-Base is significant in terms of model accuracy. We conducted McNemar's test (Raschka, 2018), which compares model accuracies of two trained models in a pairwise fashion. McNemar's test revealed that there was no statistically significant difference between BERT-Large and BERT-Base in terms of model accuracies, McNemar's $\chi^2 = 2.380$, $p = .123$. This finding highlights that the smaller model (i.e., BERT-Base) performed as well as the larger model (i.e., BERT-Large) and the smaller model was computationally more efficient (i.e., it took only 3 minutes to fine-tune as opposed to 10 minutes spent on BERT-Large for fine-tuning) than the larger model. This is also consistent with studies indicating that smaller models can perform as well as models with more parameters (Arase & Tsujii, 2019).

We also evaluated model performance on metrics precision, recall, specificity, and

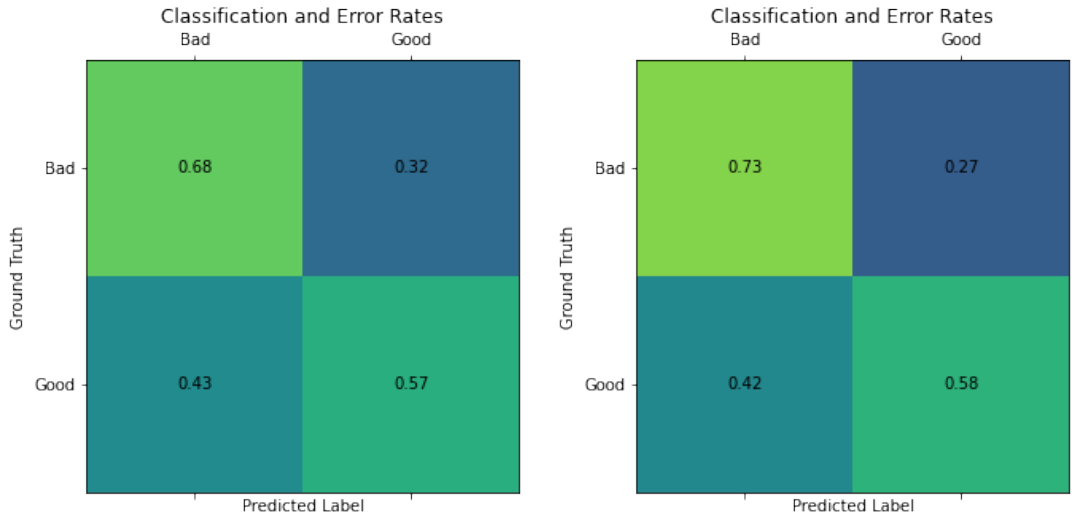
F1-score. The evaluation metrics for the overall model with BERT-Large were as follows: precision was .62, recall was .63, specificity was .68, and F1-score was .62. Whereas the evaluation metrics for the overall model with BERT-Base were as follows: precision was .66, recall was .66, specificity was .73, and F1-Score was .66. Thus, BERT-Base across the evaluation metrics slightly outperformed BERT-Large. We also compared the model performance individually for good and bad items. Both BERT-Base and BERT-Large performed better for bad items. Concerning BERT-Large, precision, recall, and F1 scores were .71, .69, and .70, respectively for bad items. On the other hand, BERT-Base had precision, recall, and F1-score of .73 for all three evaluation metrics. Concerning the good items, BERT-Large had the precision, recall, and F1-score values of .54, .57, and .55, and BERT-Base had the precision, recall, and F1-score values of .58, .58, and .58, respectively.

Table 4.5: Confusion Matrix for Fine-Tuned BERT

		Predicted			
		BERT-Large		BERT-Base	
		Good	Bad	Good	Bad
Ground Truth	Good	81	62	83	60
	Bad	70	152	59	163

To better understand the model performance of BERT-Large and BERT-Base for predicting item quality, we analyzed the confusion matrix (Table 4.5) as well as misclassification rates (Figure 4.5). Of 143 good items, 81 of them were correctly classified (68%) by BERT-Large, and 83 of them were correctly classified (73%) by BERT-Base. Concerning the 225 bad items, 152 of them were correctly classified (57%) by BERT-Large, and 163 of them were correctly classified (58%) by BERT-Base. While we did not find a significant difference between BERT-Large and BERT-Base in terms of model accuracy and both models performed similarly in terms of correctly classifying good items, BERT-Base slightly outperformed BERT-Large in terms of correctly

classifying the bad items (Table 4.5). As a matter of fact, the misclassification rate for bad items was lower for BERT-Base compared to BERT-Large (Figure 4.5).

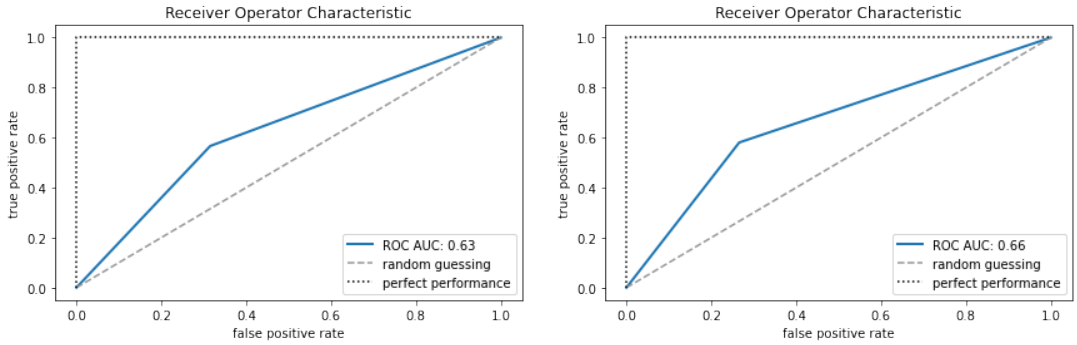


(a) BERT-Large (b) BERT-Base

Figure 4.5: (Mis)Classification Rates for Fine-Tuned BERT

For (mis)classification rates, diagonal values with lighter colors and reverse diagonal values with darker colors show better model performance. Based on the color scheme in Figure 4.5 reveals a similar model performance for both BERT-Large and BERT-Base, we can observe that BERT-Base slightly outperformed BERT-Large. Additionally, when fine-tuned BERT models compared to the ML classifier trained in Study 1, it is clear that the BERT models learned better than ML classifiers for predicting item quality as the BERT models made better predictions for good items. That is, unlike the support vector machine, logistic regression, or random forest, the BERT models predicted almost 60% of good items correctly.

Finally, we analyzed the ROC curves (Figure 4.6) to evaluate the performances of BERT-Large and BERT-Base. We observed a clear improvement in BERT models in terms of ROC curves when compared to the ML classifiers developed in Study 1. The ROC curves indicated better performance for BERT-Base compared to BERT-Large, yet this could be a negligible difference between the two models.



(a) BERT-Large (b) BERT-Base

Figure 4.6: Receiver Operating Characteristic Curves for Fine-Tuned BERT

Overall, the performance of both BERT models highlights a clear improvement in item quality prediction compared to ML classifiers trained in Study 1. While the best model in Study 1 (i.e., logistic regression) achieved a model accuracy of 61%, the best model in Study 2 (i.e., BERT-Base) achieved an accuracy of 66%. Additionally, a meticulous comparison of models, including both ML classifiers and BERT models, using confusion matrix, (mis)classification rates, and ROC curves, suggests clear progress in item quality prediction.

4.3.1 Error Analysis

Similar to Study 1, we analyzed misclassified items in the test dataset when BERT-Base and BERT-Large were used. For both models, trends among misclassified items were similar as both models shared almost the same misclassified items. Items were misclassified when a part of a specific phrase or quotation was used to create a blank (e.g., "The total federal debt is divided into '<blank>' and 'debt held by the public.'). Additionally, items were misclassified when blanks were placed at the end of an item stem or when item stems included proper nouns. Finally, extremely long (e.g., "The operative attitude may have been summed up best by the response Leiber and Stoller received when <blank> brought a serious film project for Presley to Parker and the hill and range owners for their consideration.") or extremely short (e.g., "<blank> did not even want the rays to be named after him.") item stems and complex items

that were hard to parse (e.g., "With the discovery that the disk of the Andromeda galaxy (<blank>) extends much further than previously thought, the possibility of the disk of the Milky Way galaxy extending further is apparent, and this is supported by evidence from the discovery of the outer arm extension of the Cygnus arm.") were misclassified by the BERT models. When crowdsource workers were asked to rate the items, Becker et al. (2012) provided automatically generated items along with original sentences to enhance the quality of labels. While most extremely short items were rated as bad by the crowdsource workers, some of these items were rated as good which might be due to crowdsource workers having access to the original sentences. This may also partly explain why some of these extremely short sentences were misclassified by the BERT models.

4.4 Study 3 Results

In the final study, we instruction-tuned Llama 2–7B. Of 363 cloze items in the holdout test set, 37% of items were rated as good by the crowdsource workers. The overall model accuracy of instruction-tuned Llama 2-7B was 75%. We also evaluated the performance of instruction-tuned Llama 2-7B with precision, recall, specificity, and F1-score. Specifically, we found that precision was .75, recall was .74, specificity was .85, and F1-score was .74 for the overall model. When compared to Study 1 and Study 2 results, the model performance drastically improved from 61% obtained with the ML classifier and 66% obtained with BERT-Base to 75% (Table 4.6). These results suggested the promise of using generative LLM, namely Llama 2-7B for predicting item quality. To better understand model performance, we assessed evaluation metrics separately for good and bad items as well as confusion matrix and misclassification rates.

Concerning the model performance on good items, we obtained the precision, recall, and F1-score values as follows: precision = .69, recall = .57, and F1-score = .63. On the other hand, for the bad items precision, recall, and F1-score were .77, .85, and

Table 4.6: Evaluation Metrics for Instruction-Tuned Llama 2-7B

	Accuracy	Precision	Recall	F1-Score
Overall	75%	.75	.74	.74
Good	-	.69	.57	.63
Bad	-	.77	.85	.81

.81, respectively (see Table 4.6). Our results highlighted that instruction-tuned Llama 2-7B has actually learned to identify bad items from good ones to some degree and it was not randomly guessing the label of the cloze items. If we consider the F1-scores for each level of item quality as our main evaluation method, because the F1-score is calculated by taking a weighted average of recall and precision values, we can conclude that Llama 2-7B outperformed all methods developed for evaluating item quality. Specifically, the best F1-scores were obtained with logistic regression in Study 1 as .43 for good items and .70 for bad items, and with BERT-Base in Study 2 as .58 for good items and .73 for bad items. On the contrary, Llama 2-7B has an F1-score of .63 for good items and .81 for bad items, suggesting better classification rates and fewer misclassification issues compared to Study 1 and Study 2 results.

Table 4.7: Confusion Matrix for Instruction-Tuned Llama 2-7B

		Predicted	
		Good	Bad
Ground Truth	Good	76	57
	Bad	34	196

To scrutinize error rates across quality labels in Llama 2-7B, we analyzed the confusion matrix given in Table 4.7. Of 230 cloze items rated as bad by crowdsource workers, only 15% of items ($n = 34$) were misclassified as good. On the other hand, of those cloze items rated as good by crowdsource workers, 43% ($n = 57$) were misclassified as bad. While these results are not perfect, they suggested that Llama 2-7B

showed great improvement in terms of learning to differentiate good items from bad ones compared to models developed in Study 1 and Study 2.

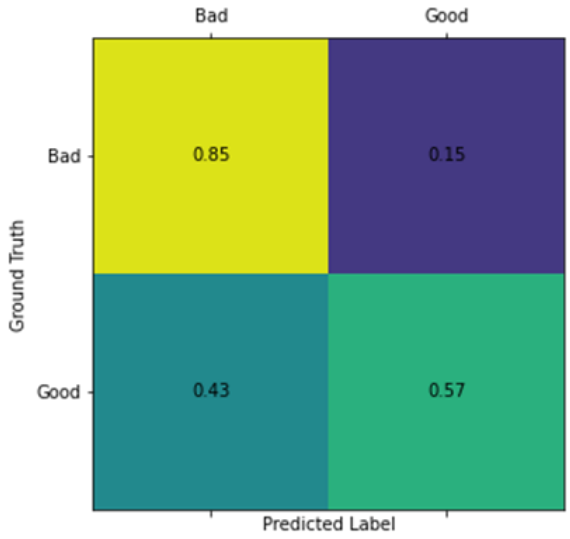


Figure 4.7: (Mis)Classification Rates for Instruction-Tuned Llama 2-7B

We analyzed (mis)classification rates in Figure 4.7. Similar to Study 1 and Study 2, diagonal values with lighter colors and reverse diagonal values with darker colors demonstrate better model performance. It is evident that, for classifying bad items, Llama 2-7B performed quite well yet the results for classifying good items were comparable to the BERT models developed in Study 2. Nonetheless, when misclassification rates and confusion matrix were analyzed holistically, we can conclude that Llama 2-7B outperformed the BERT models in terms of the overall model performance.

Finally, we also analyzed the ROC curve (Figure 4.8) obtained with the instruction-tuned Llama 2-7B. The best-performing model in Study 1 was the logistic regression with the ROC Area Under the Curve (AUC) value of .59. The best-performing model in Study 2 was BERT-Base with the ROC AUC value of .66. For Llama 2-7B, we found the ROC AUC value of .71, suggesting a significant improvement for differentiating good items from the bad ones.

Overall, these results indicated that Llama 2-7B performed well in identifying bad items as only a small portion of questions were misclassified (Figure 4.7). Nonetheless,

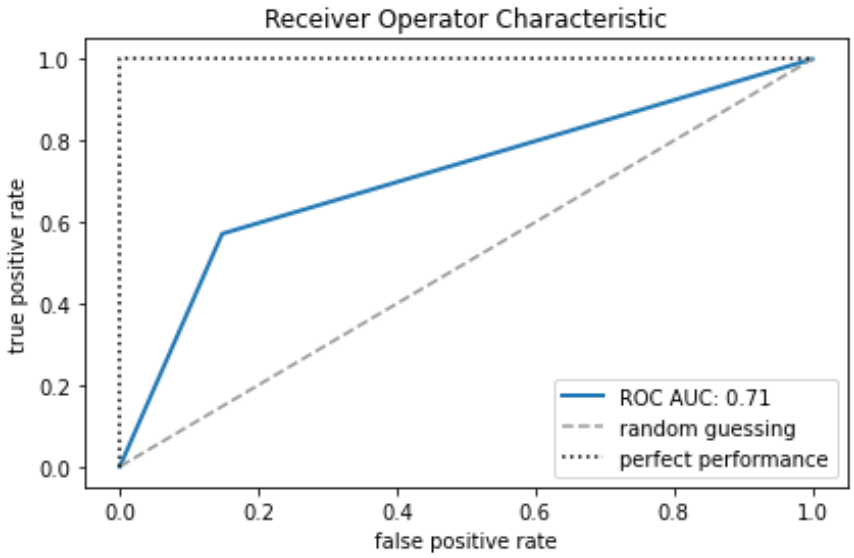


Figure 4.8: Receiver Operating Characteristic Curve for Llama 2-7B

Llama 2-7B filtered out approximately 40% of good items as bad items. However, this is a significant improvement compared to other models developed, especially compared to ML classifiers. These results highlighted a few important conclusions:

1. The instruction-tuned Llama 2-7B can be used for identifying bad items as a filtering process during the item evaluation stage before utilizing subject matter expertise,

2. Almost 40% of acceptable items may be trimmed during this process, suggesting the feasibility of this approach with situations where large item banks are or can be generated, and

3. Instruction-tuning an LLM such as Llama 2-7B can be feasible for identifying a workable size of items for field testing or evaluating items by subject matter experts.

4.4.1 Error Analysis

We conducted an error analysis for the misclassified items when the Llama 2-7B model was used. The trends were very similar to misclassified items by the BERT models

in Study 2. In Study 3, most misclassified items were good items. Those items with shorter stems (e.g., "<blank> supported that policy."), longer keyed responses (e.g., item stem: "However, Xenophon attempts to explain that Socrates purposely welcomed the hemlock <blank>, keyed response: due to his old age using the arguably self-destructive testimony to the jury as evidence), and longer stems (e.g., "There have been relatively few modern attempts to challenge this notion that <blank> is primarily Geoffrey's own work, with scholarly opinion often echoing William of Newburgh's late-12th-century comment that Geoffrey 'made up' his narrative, perhaps through an 'inordinate love of lying") were misclassified by the model. Similar to the BERT models, difficult-to-parse items were also misclassified by Llama 2-7B.

4.5 Chapter Summary

In this chapter, we discussed the results of Studies 1, 2, and 3 in detail. Our goal was to train and compare three NLP methods (i.e., ML classifiers, fine-tuning LLMs, and instruction-tuning generative LLMs) to test the feasibility of automatically evaluating item quality. In Study 1, we extracted linguistic features from item stems and keyed responses and trained random forest, support vector machine, and logistic regression classifiers. Our results highlighted the challenges associated with item quality evaluation using machine learning models. While the support vector machine performed at the chance level, we observed a slight improvement in prediction performance with classifiers trained using random forest and logistic regression, where logistic regression outperformed the other two classifiers. An interesting and useful finding was that in all classifiers, the most important features were the cosine similarity between the stem and keyed response and the cosine similarity between the predicted response and keyed response. In Study 2, we fine-tuned the two BERT models and observed improvement in model performance for predicting item quality. An important conclusion in Study 2 was that the smaller model (i.e., BERT-Base) was computationally more efficient and performed slightly better than the larger BERT model. In Study

3, we instruction-tuned Llama 2 with 7 billion parameters (i.e., Llama 2-7B) and achieved the best performance for evaluating items automatically leveraging generative LLMs. Given the misclassification rates and model performance for each level of item quality, instruction-tuning LLMs appeared to be a feasible approach to facilitate the evaluation process of items generated automatically.

Chapter 5

Discussion

This study aimed at assessing the feasibility of three NLP methods for the often-ignored issue of evaluating the quality of automatically generated items. Chapter 5 starts by recapitulating the purpose of the study. We then summarize the findings for each research question, empirical and methodological implications for practice, and ongoing challenges in the evaluation of items automatically generated. We conclude the chapter by discussing future directions for facilitating the evaluation of automatically generated items and presenting closing remarks.

5.1 Purpose of the Study

With the digital innovation in assessment, the need for large-item banks has increased drastically to supply new and high-quality items to online learning environments (e.g., intelligent tutoring systems) (Corbett et al., 1997) and adaptive assessments (e.g., computerized adaptive tests) (Weiss, 2004). Traditional item development is an iterative process where each item is developed on an individual basis, requiring extensive resources and time (Gierl & Haladyna, 2012; Romberg et al., 1982). AIG has emerged as an innovative approach to item development with the promise that many items can be generated instantly and efficiently (Bejar et al., 2002; Gierl et al., 2023; Kurdi et al., 2020). While AIG researchers have focused on developing better items leveraging newer computational methods (von Davier, 2019), the investigation

of feasible evaluation methods for generated items that allow for efficient and scalable assessment of item quality is often neglected (Gorgun & Bulut, 2024a, 2024b). Evaluation methods for AIG are not only necessary to assess the quality of each generated item but they also help to examine the quality of the AIG system holistically. That is, evaluation methods have a dual role: identifying high-quality items and assessing the overall performance of the system. This study aimed to target this gap in the literature by delineating a comprehensive overview of current evaluation methods and their limitations for AIG research and investigating alternative approaches to item evaluation.

The purpose of this study was to utilize NLP and ML methods to investigate the extent to which items generated can be evaluated automatically. Specifically, we used generated cloze items to examine the model performance of three prominent NLP methods for automatic item evaluation. Cloze items, which are also known as constructed-response or short-answer items, require examinees to formulate their answers. They are typically more difficult than selected-response items (Kuechler & Simkin, 2010), can measure more complex skills, and guessing is minimized due to the absence of response options presented along with the item stem. As such, these items have been fundamental in the classroom and large-scale assessments (Livingston, 2009). Even though cloze items have a wide array of uses, such items are shorter in length and include a masked portion (i.e., a blank in the item stem that examinees are asked to fill in with the best response), rendering them extra challenging to predict item quality with natural language processing methods.

We conducted three studies employing three NLP techniques for predicting item quality. In Study 1, we extracted linguistic features from generated items and trained random forest, support vector machine, and logistic regression. In Study 2, we employed a slightly more complex modeling approach; we fine-tuned two BERT models to predict item quality. In Study 3, we leveraged a generative LLM, Llama 2-7B, and tuned the parameter weights of the model by providing instructions related to item

quality. We compared the performance of these models using the same set of performance metrics. As a proof-of-concept study, we found that especially instruction-tuned LLMs have a promise for evaluating automatically generated cloze items, yet more research is needed to test the reliability, replicability, and generalizability of these findings across different item types, item formats, and item generation models. Below, we describe the findings for each research question and their implications for educational practice.

5.2 Discussion of Findings

5.2.1 A Taxonomy for Current AIG Evaluation Methods

We proposed a taxonomy for current AIG evaluation methods based on the resources, methods, and inputs that AIG researchers use for evaluating automatically generated items. We proposed three categories for AIG evaluation methods: (1) Metric-Based Methods, (2) Human Evaluators, and (3) Post-Hoc Evaluations.

Metric-based methods rely on the existence of reference items or ground truth and compare the generated items against, typically human-authored, reference items (Ha & Yaneva, 2018; Maurya & Desarkar, 2020). Machine translation methods such as BLEU, ROUGE-L, and METEOR (Kumar et al., 2018; Panda et al., 2022) as well as the cosine similarity based on sentence embeddings (Maurya & Desarkar, 2020; Rodriguez-Torrealba et al., 2022) have been used to assess the degree of similarity between generated items and reference items. While all generated questions can be efficiently and instantly evaluated, several limitations render the applicability of metric-based methods quite limited. The limitations of metric-based evaluations include:

1. Reference questions should be available.

2. Item quality is assessed based only on the similarity between the reference items and generated items.

3. Perfectly valid items can be filtered out due to the lack of similarity to the reference questions.

4. Some indices of item quality (i.e., educational usefulness, distractor analysis) may not be obtained with current metrics available.

The second evaluation method of the taxonomy we proposed was human evaluators. This is a rather big family of evaluation methods, utilizing humans with different levels of expertise for item evaluation. Human evaluators included subject matter experts, researchers, teachers, students, and crowdsource workers. An obvious advantage of human evaluators against the other two methods is that human evaluators can assess item quality across many different criteria (e.g., educational usefulness, fluency, or naturalness). While the capacity and expertise of humans may surpass any other method in terms of evaluating item quality across a wide range of criteria, there are some limitations when human evaluators are employed for the evaluation of the automatically generated items. These involve:

1. Evaluating items using human experts is time-consuming.

2. Typically employing humans is expensive.

3. Human experts should be trained to achieve a level of standardization in the item evaluation process.

4. Previous studies have well-documented that humans can be biased.

5. Humans might be limited in evaluating all generated items.

6. Employing human evaluators may violate the fundamental assumption of AIG which asserts that items can be generated efficiently and scalably to supply new items to itembanks.

7. Utilizing human evaluators requires a high-quality rating scale.

The final member of the proposed AIG evaluation taxonomy was post-hoc evaluations. Post-hoc evaluations provide statistical estimates regarding item functioning and item characteristics (Attali, 2018; Van Campenhout et al., 2022). Post-hoc evaluations are the only method that can be used to obtain item analysis including item difficulty, discrimination, and distractor functioning (Gierl et al., 2022). As such, they are regarded as providing objective information regarding item characteristics. We categorized experimental studies and psychometric analysis under post-hoc methods. Nevertheless, several pitfalls should be recognized:

1. Item quality is assessed retrospectively, after the test administration.
2. Sample characteristics and administration conditions may greatly influence the statistics obtained.
3. It is a time-consuming process to recruit representative samples and analyze the items administered.
4. All items generated may not be used during post-hoc evaluations, limiting the evaluation of all generated items.
5. Administrating items to a representative sample can be an expensive process.

We used this taxonomy to argue for the necessity of developing new evaluation methods along with enhancing item generation methods. To supply this demand, in the next phase of the study, we investigated the utility of various natural language processing methods for item evaluation and discussed their potential and limitations for real-world implementations.

5.2.2 Study 1: Performance of Machine Learning Classifiers

Of the three classifiers trained in Study 1, logistic regression outperformed random forest and support vector machine given the F1-score as we sought a balance between precision and recall metrics (i.e., rather than accurately identifying only bad or good

items). Logistic regression also had the highest accuracy, yet the statistical analysis comparing the accuracies of all three classifiers indicated no statistical difference between logistic regression and random forest. The support vector machine performed the worst and evaluation metrics achieved with the support vector machine suggested that the classifier predicted the majority category and no learning took place while training the classifier. The performance of the zero-rule classifier was comparable to that of the support vector machine, highlighting the poor performance of the support vector machine. Of 168 good items, the support vector machine classified only 2 good items correctly (Recall = .01). In fact, ROC curves underscored that the support vector machine predicted item quality at the chance level. Compared to the support vector machine, we observed a slightly better performance for the random forest classifier in terms of correctly classifying good items; of 168 good items, 16 were classified as good (Recall = .10). Still, the problem with classifying the majority of items as bad persisted with the random forest classifier. We observed a spike in the performance of the logistic regression classifier for accurately predicting good items; of 168 good items, 53 of them correctly classified as god (Recall = .32). While this performance is far from being acceptable to be used in real life practice, we can assert that logistic regression learned to present good items better using the linguistic features we extracted.

The underperformance of the support vector machine might be due to the fact that the support vector machine performs well with high-dimensional and multiclass data (Gorgun et al., 2022). A similar interpretation is possible for the random forest classifier where it typically performs well with more complex data (Kirasich et al., 2018). Because quality labels were dichotomized, hence we had a binary classification task, it might not be surprising to find that logistic regression outperformed both the support vector machine and random forest models. In fact, this finding is congruent with that logistic regression is a robust classifier when a classification is a binary task (Carroll & Pederson, 1993).

These results highlighted the difficulty of predicting item quality using ML classifiers, which was a consistent finding with previous studies that also extracted linguistic features and trained ML classifiers for predicting various item characteristics, including item difficulty (Baldwin et al., 2021) and mean response times (Yaneva et al., 2020). Most of these studies have focused on multiple-choice medical test items with longer stems and distractors and tried to predict item difficulty (Ha et al., 2019; Xue et al., 2020; Yaneva & Von Davier, 2023; Yaneva et al., 2019). To the best of our knowledge, this is the first study focusing on cloze items to predict item quality. As we have emphasized, the linguistic features we could extract were limited given shorter item stems and the absence of response options, possibly limiting the performance of ML classifiers trained. In addition, the previous studies have used items created by traditional test development processes (Yaneva & Von Davier, 2023; Yaneva et al., 2021), as such this study differentiates from the previous studies focusing on predicting item characteristics from the item features.

Overall, these results suggested that ML classifiers might be limited for item quality evaluation, and more advanced methods are needed to assess item quality of automatically generated items. Additionally, the selected ML classifiers failed to map item quality on the features extracted, implying either the complexity of quality prediction using solely linguistic features of items or inadequate representation of item quality using the extracted features.

5.2.3 Feature Importance for Trained Classifiers

The second phase of training the ML classifiers has focused on identifying the most important features for predicting item quality. For all classifiers trained, the top 10 features belonged to the Coh-Metrix categories of text easability principal components, situation model, lexical diversity, and word information. Text easability quantifies the difficulty that emerged from the linguistic characteristics of the text. Situation model demonstrates the mental representation of a text based on cognitive

science literature. Lexical diversity is based on the relationship between the unique words and all words in a text. Finally, word information quantifies word frequency scores and psychological ratings based on syntactic categories (Graesser et al., 2004).

Across all ML classifiers trained, the similarity between the item stem and keyed response and the similarity between the predicted response and keyed response emerged as the two most important features for item quality. These findings corroborated previous studies focusing on feature importance for predicting item quality (Hsu et al., 2018). An interesting finding regarding the feature importance was that none of the features related to the syntactic characteristics of items appeared among the top ten most important features. These two categories of Coh-Metrix construct syntactic tree structures to quantify the density of particular syntactic patterns, word types, and phrase types (Graesser et al., 2011; McNamara et al., 2014). Having none of the features from these categories gaming the top 10 list could be due to having shorter item stems or the resemblance between good and bad items in terms of sentence length distributions. This might inform studies focusing on linguistic features to predict item quality (Yaneva et al., 2021).

It is noteworthy to indicate that the features we extracted were based solely on the linguistic characteristics obtained using item stems and keyed responses, and thus we did not include any features related to the response behavior or test-taking experiences of examinees. While studies incorporating features extracted based on examinee test-taking experiences performed much better in terms of predicting item quality (McBroom & Paassen, 2020; Tong et al., 2020), the possibility of extracting these features depends on items being administered to a representative population. The limitations of this type of feature extraction are similar to post-hoc evaluations discussed in Section 2.3.3.

5.2.4 Study 2: Performance of Fine-Tuned BERT

In Study 2, we focused on utilizing more complex models for predicting item quality. Specifically, we used an LLM, BERT, and fine-tuned BERT-Base and BERT-Large to classify items based on quality. Fine-tuning BERT means that the weights of parameters are adjusted based on the pooled embeddings allocated at the [CLS] token of the last layer of the BERT model. Weight adjustment is established by adding a linear layer on top of the model and by re-training the model using the [CLS] tokens (Devlin et al., 2019).

Both fine-tuned BERT models (i.e., BERT-Base and BERT-Large) outperformed ML classifiers trained in Study 1, and BERT-Base slightly had superior performance than BERT-Large. All metrics, including accuracy, precision, recall, and F1-score, were higher for fine-tuned models compared to ML classifiers. In addition, the confusion matrix also supported that fine-tuned BERT has better differentiated good and bad items from one another. For instance, of those 143 good items, 83 were correctly classified (Recall .58). Furthermore, the fine-tuned BERT models made more misclassification errors for bad items compared to ML classifiers, suggesting that, in fact, the fine-tuned BERT models learned better to differentiate the bad items from the good ones, and the prediction was based less on the majority category. These findings were also consistent with previous research focusing on predicting item characteristics using linguistic features. While previous studies have used LLMs such as ELMo, Word2Vec, and BERT to obtain sentence and token embeddings as features to be included during ML classifier training (e.g., random forest, linear regression) (Baldwin et al., 2021), the best model performance was achieved when LLM embeddings were used in the model (Yaneva & Von Davier, 2023).

Nonetheless, to the best of our knowledge, none of the studies have focused on directly fine-tuning an LLM for predicting item quality. This might be due to a few reasons: previous studies have focused on predicting item difficulty rather than binary

item quality necessitating regression models (Štěpánek et al., 2023; Xue et al., 2020), and previous studies combined token and sentence embeddings with other linguistic features (Yaneva et al., 2021). Future research may compare the model performance of fine-tuning BERT with using averaged BERT embeddings as features while training ML classifiers. While a few studies compared the influence of fine-tuning LLMs for specific tasks (Lajkó et al., 2022; Mayfield & Black, 2020; Peters et al., 2019; Wang et al., 2019), the effect of fine-tuning on item quality prediction is unknown and remains a potential future research direction.

An interesting finding is that BERT-Base outperformed BERT-Large when several evaluation metrics were considered (e.g., F1-score). It is important to remind the reader that there was no significant difference in model accuracies of BERT-Base and BERT-Large. Nonetheless, the difference in some evaluation metrics of the fine-tuned BERT models was an interesting and unexpected finding as benchmark studies have often shown that larger models outperform the smaller ones (Rydzewski et al., 2024). This unexpected finding might be due to the small training data size (Ezen-Can, 2020). In addition, it could be the case that the smaller model has adapted to the item evaluation task more effectively by leveraging pre-existing knowledge (Raffel et al., 2020).

5.2.5 Study 3: Performance of Instruction-Tuned Llama-2

As our final natural language processing model, we instruction-tuned an open-source LLM, Llama-2, with 7 billion parameters. As a reminder, this is the smallest Llama model in terms of the number of parameters. We adjusted the parameter weight of Llama 2-7B by providing instructions about the item quality and employing a few innovative approaches for model tuning, This approach included QLoRA (Dettmers et al., 2023) and PEFT (Liu et al., 2022) that systematically reduce computational costs and memory demands by selecting the most important parameters during training and compressing the trained model in a 4-bit format. As such, it was possible to

re-train a generative LLM with 7 billion parameters for the specific task at hand, i.e., item quality evaluation.

During the instruction-tuning phase, we only provided 1462 examples composed of instructions (i.e., prompt), input (i.e., item stem), and output (i.e., quality label). While ML and NLP methods rely on big data to discover (hidden) patterns in the data, this may not be an attainable goal for all scenarios. In some cases, obtaining labeled data might be costly and time-consuming and in other cases, large data points may create concerns about privacy and ethics (Parnami & Lee, 2022). Few-shot learning for these scenarios has emerged as a low-cost solution with the possibility of achieving high-model performance (Wang et al., 2020). This study is one example of few-shot training conducted using Llama 2-7B. This modeling approach is exemplary for future research intending to evaluate generated items with a low-cost solution.

Overall, instruction-tuning a generative LLM was the best-performing model in terms of accuracy, precision, recall, and F1-score. Instruction-tuned Llama 2-7B outperformed both BERT models and ML classifiers, and performance metrics suggested that the model learned to differentiate good items from bad ones better. The performance of instruction-tuned Llama 2-7B was better than that of ML classifiers because it predicted good items remarkably well compared to logistic regression or random forest. Additionally, the performance of instruction-tuned Llama 2-7B was better than the BERT models because the misclassification rates of good and bad items were lower for Llama 2-7B compared to the fine-tuned BERT models. Furthermore, instruction-tuned Llama 2-7B emerged as a viable model for filtering out bad items because only 15% of bad items were misclassified. As AIG promises to generate many items instantly and efficiently, such instruction-tuned LLM models could be introduced as an intermediate new step to filter out bad items before taking generated items to human evaluators or field-testing items. This novel intermediate process may alleviate resource, cost, and time requirements, rendering AIG methods more efficient and scalable for operational test settings. Nonetheless, one crucial caveat

of the instruction-tuned LLM should be recognized. Almost 40% of the good items might be filtered out as bad, and depending on the item bank size, a significant portion of items could be trimmed during this process. In addition, the filtered-out good items may have desirable item characteristics; hence, useful items might be lost during this phase. However, if many items are generated, this evaluation process holds the potential to facilitate item selection for operational use. This finding emphasized the need for new considerations in promoting the pipeline from item generation to item deployment in operational testing.

5.3 Contributions

Before we discuss the limitations and future directions, we summarize the study's main contributions and discuss how the results are informative for future research. First, we provided a comprehensive overview of quality criteria and evaluation methods used by both traditional item developers and AIG researchers. Although AIG is a heavily interdisciplinary field, the literature review highlighted the lack of communication among educators, traditional test developers, computer scientists, and AIG researchers (Cukurova et al., 2019; Luckin & Cukurova, 2019). This study is an attempt to foster interdisciplinary communications around AIG. We hope that the taxonomy proposed for evaluation methods, as well as the quality criteria discussed, provide a thorough portrait of current practices in AIG research.

Second, this study was one of the first attempts to evaluate the quality of automatically generated items using a set of natural language processing methods. While previous studies have mainly focused on predicting item characteristics (e.g., difficulty or average response time) (Baldwin et al., 2021; Xue et al., 2020; Yaneva & Von Davier, 2023), our main focus was items that are generated by computer algorithms and considering item quality as a holistic concept (refer to Sections 3.1 and B.4 to see how item quality was defined). In this study, our primary goal was to gauge the extent to which we could evaluate item quality automatically to assess all

generated items efficiently and scalably. Although as in its current form, this study is a proof-of-concept study, we hope to introduce new methods for item evaluation and discuss how it can be incorporated into the item generation pipeline.

Third, we compared the performance of various natural language processing methods for item evaluation. While some of these methods (e.g., training ML classifiers or extracting sentence and word embeddings) have been used in previous studies for predicting item characteristics (Yaneva et al., 2020), some methods we used in this study (i.e., fine-tuning LLMs and instruction-tuning generative LLMs) are novel approaches to item evaluation.

Finally, we used an open-source generative LLM, Llama 2-7B in this study instead of a more popular LLM of GPT 4. This study sheds light on the performance of an alternative LLM that is pre-trained with open-source data and highlights the possibility of using other LLMs for such intricate tasks.

5.4 Limitations

The results of this study should be carefully interpreted as it is a proof-of-concept study and findings have limited generalizability due to using only one dataset for assessing the feasibility of NLP methods for item evaluation.

5.4.1 Limitations Related to the Dataset

This study utilized a secondary dataset that included items generated before the advent of generative LLMs. Using constituency parsing and semantic role labeler, the parts of the sentences extracted from Wikipedia articles were masked to generate cloze items (Becker et al., 2012). As such the generated items are immune to several quality criteria, including fluency, sensibleness, naturalness, grammaticality, and fluency. Therefore, the evaluation criteria are limited in terms of assessing the quality of items generated. We specifically focused on quality criteria that involved ambiguity issues in items and whether the masked portion was reasonable for examinees to

answer or not.

The second limitation is related to the use of generative LLMs for item generation. The vast majority of current item generation methods make use of generative LLMs (Hommel et al., 2022; Wang et al., 2024). These generative LLMs have the potential to create items that sound more like human-authored ones. As we see a dramatic increase in AIG methods leveraging generative LLMs, it is interesting to investigate how item generation achieved with LLMs influences the performance of evaluation methods relying on these LLMs.

A third limitation is related to the type of items used for assessing the feasibility of NLP and ML models for evaluating item quality. Specifically, in this study, we used only cloze items that were generated in a specific way before the LLM era to assess the feasibility of evaluating items automatically using NLP and ML methods, limiting the generalizability of findings across different item types. Future research should replicate these findings across different item types (e.g., selected-response), assessment contexts (e.g., medicine, math), and item formats (e.g., passage-based items) to investigate the extent to which NLP and ML can be used for item evaluation.

Another limitation is that items in the dataset are relatively short, limiting the quality and variety of features that could be extracted from item stems. While this could demonstrate the rigor of methods we developed for quality prediction, it also limits the generalizability of findings across different item types and formats.

The final limitation is about the nature of the dataset. The dataset involves generic items, not targeting any specific educational content area. While items are a subtype of constructed-response items, there was no purpose or use of assessment associated with item generation, limiting discussion around the validity of items for specific use.

5.4.2 Limitations Related to Quality Labeling

In this study, crowdsource workers have been used for obtaining labels in terms of item quality. We have already pointed out limitations pertaining to utilizing crowdsource

workers for item evaluation in Section 2.3.2. Although there are concerns about the quality of ratings, a few measures have been taken by Becker et al. (2012) such as each generated item being evaluated by four raters and the agreement between raters being considered during the data cleaning phase. Nonetheless, Becker et al. (2012) have not provided demographic information, pedagogical expertise, and the rate of compensation offered to crowdsource workers, limiting researchers or practitioners to evaluate the quality of labels assigned by crowdsource workers (Alelyani & Yang, 2016).

Additionally, a binary classification approach has been utilized, reducing the richness of item quality indicators in a dichotomous decision problem. Following Becker et al. (2012), we binarized the quality labels and collapsed the categories of *Okay* items with *Bad* items, reducing the information available about the item quality. While our goal was to develop a classification model differentiating good items from bad ones, we may have missed important information regarding the degree of item quality. In other words, quality could be formulated as a degree rather than a binary categorization of good vs. bad items.

5.4.3 Limitations Related to Modeling Practices

The data we used for training ML classifiers, fine-tuning BERT, and instruction-tuning Llama are small, composed of only 1825 items after the data cleaning steps. Because of the smaller data size, the performance of natural language methods could be attenuated. While this setting allowed us to utilize few-shot training, the methods should be replicated with larger datasets to assess the impact of data size on model performance fully.

A second limitation of modeling practices is related to using LLM. LLMs are a black-box approach and the pre-training process (e.g., datasets used, hyperparameter selection) is typically unknown to researchers using LLMs. This makes it extremely hard to understand the tuning process as well as interpret biases involved in pre-

trained LLMs. Because of these limitations and the lack of model interpretability, the findings should be carefully examined to alleviate potential biases and prejudices inherent in the models.

Finally, we also employed a single LLM, limiting the generalizability of findings across different LLMs available. For instance, Llama-2 is an open-source LLM trained with open-source data. There are larger and commercial LLMs available (e.g., GPT-4, Falcon) that may outperform the LLama-2 model tuned in this study. While, as a proof-of-concept study, this study provided preliminary results on the utility of LLMs for item evaluation, the feasibility of other LLMs should be examined to establish benchmarking on item evaluation.

5.5 Future Directions

Future research should address the limitations listed above as well as investigate the extent of reproducibility and replicability of results across different assessment and learning settings. Especially, the reliability of LLMs should be further investigated for item quality evaluation. Future research may assess the utility of methods employed in this study for generated items that allow for richer quality criteria. That is, rather than solely focusing on the ambiguity and reasonableness criteria employed in this study, future research may consider quality criteria such as distractor quality, item difficulty, educational usefulness, or grammatically.

An important future direction is to examine the generalizability and applicability of these findings when different item types are used. In this study, we used only automatically generated cloze items to assess the feasibility of NLP techniques for item evaluation. It is imperative to investigate the utility of these models with other item types (e.g., multiple-choice or short-answer items) or formats (e.g., passage-based or image-based items) to gauge the generalizability of these findings. Furthermore, a study comparing different LLMs including open-source and commercial ones should be conducted to compare the performance of various baseline and tuned LLMs for

item quality prediction. Such a study could be conducted utilizing different AIG datasets and LLMs to fully capture the role of LLMs in the future for automating the item quality evaluation process.

In addition, future research may investigate the feasibility of using natural language processing methods when items are generated with generative LLMs. Especially, it is an interesting research direction to investigate how a generative LLM can be used for assessing items generated using generative LLMs. Future research may also focus on using larger datasets to investigate the influence of item bank size on automatic item quality evaluations.

The findings of this study show the performance of three natural language processing methods when the item bank is small. Another possible direction is investigating the relationship between validity arguments and item quality evaluation with NLP methods. Furthermore, future research should investigate using finer-grained quality labels (rather than just dichotomizing the quality labels) and criteria to examine the utility of NLP methods for item evaluation.

In this study, we have exclusively focused on item quality criteria. Although item quality is indispensable for assessment validity, the intricate relationship between (assessment) validity and using NLP methods for item evaluation should be examined. Additionally, predicting item statistics using such automatic methods could facilitate the transition from item generation to deployment in personalized assessment and learning environments. The findings obtained in this study may shed light on future research investigating the utility of linguistic features and NLP techniques for predicting item statistics (e.g., difficulty or discrimination indices of items).

Finally, more systematic item generation and labeling processes should be followed to fully capture the utility of natural language processing methods for item evaluation. For example, items could be generated for a specific assessment purpose, and various quality criteria along with different labeling approaches should be adopted to explore congruence among different labeling methods as well as to scrutinize the feasibility of

item evaluation through natural language processing methods.

5.6 Conclusion

This study assessed the feasibility of three natural language processing methods for automatically evaluating generated cloze items. As model complexity increased, from trained ML classifiers to fine-tuned BERT and instruction-tuned Llama, we observed amelioration in the model performance in terms of accurately predicting item quality. Especially Llama emerged as a viable option for evaluating item quality, as misclassification rates were minimized for the holdout test set compared to the other two methods. While this might not be a surprising finding given that larger models typically perform better for classification tasks (Ramesh & Sanampudi, 2022), it is interesting that the smaller BERT model outperformed the larger BERT model. This underscores the need for a systematic investigation of LLMs for item evaluation. Based on these findings, we suggest incorporating instruction-tuned LLMs as an intermediate step in the item generation pipeline to filter out bad items after generating items automatically. This process may significantly reduce costs, resources, and time requirements associated with evaluating all generated items using human evaluators or field-testing. That is, by running generated items through an instruction-tuned LLM, AIG researchers can get an initial estimate regarding the item quality. Then, bad items can be filtered out and a workable portion of items can be evaluated by experts or field-tested to obtain further item characteristics. It is worth noting that there is a lot of room for improving the item evaluation process through LLMs, yet these findings corroborate the promise of LLMs for automatic item evaluation.

References

Abraham, R. G., & Chapelle, C. A. (1992). The meaning of cloze test scores: An item difficulty perspective. *The Modern Language Journal*, *76*(4), 468–479. https://doi.org/10.2307/330047

Afzal, N., & Mitkov, R. (2014). Automatic generation of multiple choice questions using dependency-based semantic relations. *Soft Computing*, *18*(7), 1269–1281. https://doi.org/10.1007/s00500-013-1141-4

Alelyani, T., & Yang, Y. (2016). Software crowdsourcing reliability: An empirical study on developers behavior. *Proceedings of the 2nd International Workshop on Software Analytics*, 36–42. https://doi.org/10.1145/2989238.2989245

Allen, M. J., & Yen, W. M. (2001). *Introduction to the measurement theory*. Waveland Press.

Alsubait, T., Parsia, B., & Sattler, U. (2016). Ontology-based multiple choice question generation. *KI - Künstliche Intelligenz*, *30*(2), 183–188. https://doi.org/10.1007/s13218-015-0405-9

Alsubait, T. M. (2015). *Ontology-based multiple-choice question generation* [Doctoral dissertation, University of Manchester].

Alsubait, T. M., Parsia, B., & Sattler, U. (2014). Generating multiple choice questions from ontologies: Lessons learnt. *OWLED*, 73–84.

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

Amidei, J., Piwek, P., & Willis, A. (2018). Evaluation methodologies in automatic question generation 2013-2018. *Proceedings of the 11th International Conference on Natural Language Generation*, 307–317. https://doi.org/10.18653/v1/W18-6537

Anastasi, A., & Urbina, S. (2004). *Psychological testing* (7th ed.). Pearson.

Arase, Y., & Tsujii, J. (2019). Transfer fine-tuning: A BERT case study. *arXiv preprint arXiv:1909.00931*. https://doi.org/https://doi.org/10.48550/arXiv.1909.00931

Ashraf, Z. A. (2020). Classical and modern methods in item analysis of test tools. *International Journal of Research and Review*, *7*(5), 397–403.

Attali, Y. (2018). Automatic item generation unleashed: An evaluation of a large-scale deployment of item models. In C. Penstein Rosé, R. Martínez-Maldonado, H. U. Hoppe, R. Luckin, M. Mavrikis, K. Porayska-Pomsta, B. McLaren, &

B. du Boulay (Eds.), *Artificial Intelligence in Education* (pp. 17–29). Springer International Publishing. https://doi.org/10.1007/978-3-319-93843-1_2

Attali, Y., Runge, A., LaFlair, G. T., Yancey, K., Goodwin, S., Park, Y., & von Davier, A. A. (2022). The interactive reading task: Transformer-based automatic item generation. *Frontiers in Artificial Intelligence*, *5*, 903077. https://doi.org/10.3389/frai.2022.903077

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). https://doi.org/10.1111/jedm.12264

Bandalos, D. L. (2018). *Measurement theory and applications for the social sciences*. Guilford Publications.

Banerjee, S., & Lavie, A. (2005). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, 65–72.

Barrada, J., Olea, J., Ponsada, V., Abad, F., Ponsoda, V., & Abad, F. (2009). Test overlap rate and item exposure rate as indicators of test security in CATs. *Proceedings of the 2009 GMAC Conference on Computerized Adaptive Testing*.

Becker, L., Basu, S., & Vanderwende, L. (2012). Mind the gap: Learning to choose gaps for question generation. *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 742–751.

Bejar, I. I. (1983). Subject matter experts' assessment of item statistics. *Applied Psychological Measurement*, *7*(3), 303–310. https://doi.org/10.1177/014662168300700306

Bejar, I. I., Lawless, R. R., Morley, M. E., Wagner, M. E., Bennett, R. E., & Revuelta, J. (2002). A feasibility study of on-the-fly item generation in adaptive testing. *ETS Research Report Series*, 1–44.

Beseiso, M., & Alzahrani, S. (2020). An empirical analysis of bert embedding for automated essay scoring. *International Journal of Advanced Computer Science and Applications*, *11*(10).

Bezirhan, U., & von Davier, M. (2023). Automated reading passage generation with OpenAI's large language model. *arXiv*. https://doi.org/10.48550/arXiv.2304.04616

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python: Analyzing text with the natural language toolkit*. O'Reilly Media, Inc.

Black, P., & Wiliam, D. (1998). Assessment and classroom learning. *Assessment in Education: Principles, Policy & Practice*, *5*(1), 7–74. https://doi.org/10.1080/0969595980050102

Breiman, L. (2001). Random forests. *Machine Learning*, *45*, 5–32.

Breithaupt, K., Ariel, A., & Hare, D. (2010). Assembling an inventory of multistage adaptive testing systems. In W. Van Der Linden & C. Glas (Eds.), *Elements of adaptive testing* (pp. 247–266). Springer.

Brennan, R. (2006). *Educational measurement* (4th edition). Praeger Publishers.

Bulle, N. (2011). Comparing OECD educational models through the prism of PISA. *Comparative Education*, *47*(4), 503–521. https://doi.org/10.1080/03050068.2011.555117

Bulut, O., & Kan, A. (2012). Application of computerized adaptive testing to entrance examination for graduate studies in Turkey. *Eurasian Journal of Educational Research*, (49), 61–80.

Carroll, R. J., & Pederson, S. (1993). On robustness in the logistic regression model. *Journal of the Royal Statistical Society: Series B (Methodological)*, *55*(3), 693–706.

Chalifour, C. L., & Powers, D. E. (1989). The relationship of content characteristics of GRE analytical reasoning items to their difficulties and discriminations. *Journal of Educational Measurement*, *26*(2), 120–132. https://doi.org/https://doi.org/10.1111/j.1745-3984.1989.tb00323.x

Chughtai, R., Azam, F., Anwar, M. W., Haider But, W., & Farooq, M. U. (2022). A lecture centric automated distractor generation for post-graduate software engineering courses. *2022 International Conference on Frontiers of Information Technology (FIT)*, 100–105. https://doi.org/10.1109/FIT57066.2022.00028

Chung, C.-Y., & Hsiao, I.-H. (2022). Programming question generation by a semantic network: A preliminary user study with experienced instructors. In M. M. Rodrigo, N. Matsuda, A. I. Cristea, & V. Dimitrova (Eds.), *Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners' and Doctoral Consortium* (pp. 463–466, Vol. 13356). Springer International Publishing. https://doi.org/10.1007/978-3-031-11647-6_93

Clauser, J. C., & Hambleton, R. K. (2011). Item analysis procedures for classroom assessments in higher education. In C. Secolsky & D. B. Denison (Eds.), *Handbook on Measurement, Assessment, and Evaluation in Higher Education* (pp. 296–309). Routledge.

Cohen, R. J., Swerdlick, M. E., & Phillips, S. M. (1996). Test development. In *Psychological testing and assessment: An introduction to tests and measurement* (3rd ed., pp. 218–254). Mayfield Publishing Co.

Cole, J. S., Bergin, D. A., & Whittaker, T. A. (2008). Predicting student achievement for low stakes tests with effort and task value. *Contemporary Educational Psychology*, *33*(4), 609–624. https://doi.org/10.1016/j.cedpsych.2007.10.002

Corbett, A. T., Koedinger, K. R., & Anderson, J. R. (1997, January). Chapter 37 - Intelligent Tutoring Systems. In M. G. Helander, T. K. Landauer, & P. V. Prabhu (Eds.), *Handbook of Human-Computer Interaction* (2nd ed., pp. 849–874). North-Holland. https://doi.org/10.1016/B978-044481862-1.50103-5

Cukurova, M., Luckin, R., & Clark-Wilson, A. (2019). Creating the golden triangle of evidence-informed education technology with EDUCATE. *British Journal of Educational Technology*, *50*(2), 490–504. https://doi.org/https://doi.org/10.1111/bjet.12727

Das, S., Chatterjee, R., & Mandal, J. K. (2016). An approach for creating framework for automated question generation from instructional objective. In S. C. Satapathy, K. S. Raju, J. K. Mandal, & V. Bhateja (Eds.), *Proceedings of the*

Second International Conference on Computer and Communication Technologies (pp. 527–535). Springer India. https://doi.org/10.1007/978-81-322-2517-1_51

Dębska, B., & Kubacka, A. (2012). A model of formative assessment on the basis of the example of e-student portal. In E. Smyrnova-Trybulska (Ed.), *E-learning for societal needs* (pp. 303–314). Studio Noa.

Demmans Epp, C., & Phirangee, K. (2019). Exploring mobile tool integration: Design activities carefully or students may not learn. *Contemporary Educational Psychology*, *59*, 101791. https://doi.org/https://doi.org/10.1016/j.cedpsych.2019.101791

Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). QLoRA: Efficient finetuning of quantized LLMs. https://doi.org/10.48550/arXiv.2305.14314

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. http://arxiv.org/abs/1810.04805

Domladovac, M. (2021). Comparison of neural network with gradient boosted trees, random forest, logistic regression and svm in predicting student achievement. *2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO)*, 211–216.

Drasgow, F., & Olson-Buchanan, J. B. (Eds.). (1999). *Innovations in computerized assessment*. Psychology Press. https://doi.org/10.4324/9781410602527

Dugan, L., Miltsakaki, E., Upadhyay, S., Ginsberg, E., Gonzalez, H., Choi, D., Yuan, C., & Callison-Burch, C. (2022). A feasibility study of answer-agnostic question generation for education. *Findings of the Association for Computational Linguistics: ACL 2022*, 1919–1926.

Ebel, R. L., & Frisbie, D. A. (1986). Using test and item analysis to evaluate and improve test quality. In *Essentials of educational measurement* (4th ed., pp. 223–242). Prentice-Hall.

Ewell, P. T. (2008). Assessment and accountability in America today: Background and context. *New Directions for Institutional Research*, *2008*, 7–17. https://doi.org/10.1002/ir.258

Ezen-Can, A. (2020). A comparison of LSTM and BERT for small corpus. *arXiv preprint arXiv:2009.05451*. https://doi.org/https://doi.org/10.48550/arXiv.2009.05451

Feng, M., & Heffernan, N. T. (2005). Informing teachers live about student learning: Reporting in the Assistment system. *Technology Instruction Cognition and Learning*, *3*, 1–14.

Feng, M., Heffernan, N. T., & Koedinger, K. R. (2006). Predicting state test scores better with intelligent tutoring systems: Developing metrics to measure assistance required. In M. Ikeda, K. D. Ashley, & T.-W. Chan (Eds.), *Intelligent Tutoring Systems* (pp. 31–40). Springer. https://doi.org/10.1007/11774303_4

Flor, M., & Riordan, B. (2018). A semantic role-based approach to open-domain automatic question generation. *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, 254–263. https://doi.org/10.18653/v1/W18-0530

Flowers, L., Osterlind, S. J., Pascarella, E. T., & Pierson, C. T. (2001). How much do students learn in college? *The Journal of Higher Education*, *72*(5), 565–583. https://doi.org/10.1080/00221546.2001.11777114

French, C. L. (2001). A review of classical methods of item analysis. *Annual Meeting of the Southwest Educational Research Association*, 1–23.

Fu, Y., Choe, E. M., Lim, H., & Choi, J. (2022). An evaluation of automatic item generation: A case study of weak theory approach. *Educational Measurement: Issues and Practice*, *41*(4), 10–22. https://doi.org/10.1111/emip.12529

Gao, Y., Bing, L., Chen, W., Lyu, M. R., & King, I. (2019). Difficulty controllable generation of reading comprehension questions. *arXiv*. http://arxiv.org/abs/1807.03586

Gatt, A., & Krahmer, E. (2018). Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, *61*, 65–170. https://doi.org/10.1613/jair.5477

Gierl, M. J., Bulut, O., Guo, Q., & Zhang, X. (2017). Developing, analyzing, and using distractors for multiple-choice tests in education: A comprehensive review. *Review of Educational Research*, *87*(6), 1082–1116. https://doi.org/10.3102/0034654317726529

Gierl, M. J., & Haladyna, T. M. (2012). Automatic item generation. In M. J. Gierl & T. M. Haladyna (Eds.), *Automatic item generation : Theory and practice* (pp. 3–12). Routledge.

Gierl, M. J., & Lai, H. (2012). The role of item models in automatic item generation. *International Journal of Testing*, *12*(3), 273–298. https://doi.org/10.1080/15305058.2011.635830

Gierl, M. J., & Lai, H. (2013). Using automated processes to generate test items. *Educational Measurement: Issues and Practice*, *32*, 36–50. https://doi.org/https://doi.org/10.1111/emip.12018

Gierl, M. J., & Lai, H. (2016). A process for reviewing and evaluating generated test items. *Educational Measurement: Issues and Practice*, *35*(4), 6–20. https://doi.org/10.1111/emip.12129

Gierl, M. J., Lai, H., Pugh, D., Touchie, C., Boulais, A.-P., & De Champlain, A. (2016). Evaluating the psychometric characteristics of generated multiple-choice test items. *Applied Measurement in Education*, *29*(3), 196–210. https://doi.org/10.1080/08957347.2016.1171768

Gierl, M. J., Lai, H., & Tanygin, V. (2021a). Introduction: The changing context of educational testing. In *Advanced methods in automatic item generation* (1st ed., pp. 1–19). Routledge. https://doi.org/10.4324/9781003025634

Gierl, M. J., Lai, H., & Tanygin, V. (2021b). Methods for validating generated items: A focus on model-level outcomes. In *Advanced methods in automatic item generation* (1st ed., pp. 120–143). Routledge. https://doi.org/10.4324/9781003025634

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. https://doi.org/10.1111/j.1365-2923.2012.04289.x

Gierl, M. J., Shin, J., & Firoozi, T. (2023). Automatic item generation. In *International encyclopedia of education* (4th ed., pp. 193–200). Elsevier. https://doi.org/10.1016/B978-0-12-818630-5.10026-0

Gierl, M. J., Swygert, K., Matovinovic, D., Kulesher, A., & Lai, H. (2022). Three sources of validation evidence needed to evaluate the quality of generated test items for medical licensure. *Teaching and Learning in Medicine*, 1–11. https://doi.org/10.1080/10401334.2022.2119569

Goertz, M., & Duffy, M. (2003). Mapping the landscape of high-stakes testing and accountability programs. *Theory Into Practice*, *42*(1), 4–11. https://doi.org/10.1207/s15430421tip4201_2

Gorgun, G., & Bulut, O. (2021). A polytomous scoring approach to handle not-reached items in low-stakes assessments [Publisher: SAGE Publications Inc]. *Educational and Psychological Measurement*, *81*(5), 847–871. https://doi.org/10.1177/0013164421991211

Gorgun, G., & Bulut, O. (2022). Considering disengaged responses in bayesian and deep knowledge tracing. In M. M. Rodrigo, N. Matsuda, A. I. Cristea, & V. Dimitrova (Eds.), *Artificial intelligence in education. posters and late breaking results, workshops and tutorials, industry and innovation tracks, practitioners' and doctoral consortium* (pp. 591–594). Springer International Publishing.

Gorgun, G., & Bulut, O. (2023). Incorporating test-taking engagement into the item selection algorithm in low-stakes computerized adaptive tests. *Large-scale Assessments in Education*, *11*(1), 27. https://doi.org/https://doi.org/10.1186/s40536-023-00177-5

Gorgun, G., & Bulut, O. (2024a). Current evaluation methods are a bottleneck in automatic question generation. *AI for education: Bridging innovation and responsibility at the 38th AAAI annual conference on AI.*

Gorgun, G., & Bulut, O. (2024b). Exploring quality criteria and evaluation methods in automated question generation: A comprehensive survey. *Education and Information Technologies*. https://doi.org/10.1007/s10639-024-12771-3

Gorgun, G., Yildirim-Erbasli, S. N., & Epp, C. D. (2022, July). Predicting cognitive engagement in online course discussion forums. In A. Mitrovic & N. Bosch (Eds.), *Proceedings of the 15th international conference on educational data mining* (pp. 276–289). International Educational Data Mining Society. https://doi.org/10.5281/zenodo.6853149

Graesser, A. C., Li, H., & Forsyth, C. (2014). Learning by communicating in natural language with conversational agents. *Current Directions in Psychological Science*, *23*(5), 374–380. https://doi.org/10.1177/0963721414540680

Graesser, A. C., McNamara, D. S., & Kulikowich, J. M. (2011). Coh-Metrix: Providing multilevel analyses of text characteristics. *Educational Researcher*, *40*(5), 223–234. https://doi.org/https://doi.org/10.3102/0013189X11413260

Graesser, A. C., McNamara, D. S., Louwerse, M. M., & Cai, Z. (2004). Coh-Metrix: Analysis of text on cohesion and language. *Behavior Research Methods, Instruments, & Computers*, *36*(2), 193–202. https://doi.org/https://doi.org/10.3758/BF03195564

Guo, H., Robin, F., & Dorans, N. (2017). Detecting item drift in large-scale testing. *Journal of Educational Measurement*, *54*(3), 265–284. https://doi.org/10.1111/jedm.12144

Ha, L. A., & Yaneva, V. (2018). Automatic distractor suggestion for multiple-choice tests using concept embeddings and information retrieval. *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, 389–398. https://doi.org/10.18653/v1/W18-0548

Ha, L. A., & Yaneva, V. (2019). Automatic question answering for medical MCQs: Can it go further than information retrieval?

Ha, L. A., Yaneva, V., Baldwin, P., & Mee, J. (2019, August). Predicting the difficulty of multiple choice questions in a high-stakes medical exam. In H. Yannakoudakis, E. Kochmar, C. Leacock, N. Madnani, I. Pilán, & T. Zesch (Eds.), *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications* (pp. 11–20). Association for Computational Linguistics. https://doi.org/10.18653/v1/W19-4402

Haladyna, T. M., Downing, S. M., & Rodriguez, M. C. (2002). A review of multiple-choice item writing guidelines for classroom assessment. *Applied Measurement in Education*, *15*(3), 309–333. https://doi.org/10.1207/S15324818AME1503_5

Haladyna, T. M., & Rodriguez, M. C. (2013). *Developing and validating test items*. Routledge.

Haladyna, T. M., & Rodriguez, M. C. (2021). Using full-information item analysis to improve item quality. *Educational Assessment*, *26*(3), 198–211. https://doi.org/10.1080/10627197.2021.1946390

Hambleton, R. K. (1993). Principles and selected applications of item response theory. In R. L. Linn (Ed.), *Educational measurement* (pp. 147–200). Oryx.

Hambleton, R. K., Swaminathan, H., & Rogers, H. J. (1991). *Fundamentals of item response theory*. Sage.

Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., & Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and their applications*, *13*(4), 18–28.

Heilman, M. (2011). *Automatic factual question generation from text* [Ph.D.]. Carnegie Mellon University.

Heilman, M., & Smith, N. A. (2010). Good question! Statistical ranking for question generation. *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 609–617.

Heubert, J. P., & Hauser, R. M. (Eds.). (1999). *High stakes: Testing for tracking, promotion, and graduation*. National Academy Press.

Hommel, B. E., Wollang, F.-J. M., Kotova, V., Zacher, H., & Schmukle, S. C. (2022). Transformer-based deep neural language modeling for construct-specific automatic item generation. *Psychometrika*, *87*(2), 749–772. https://doi.org/10.1007/s11336-021-09823-9

Hovy, E. (1999). Toward finely differentiated evaluation metrics for machine translation. *Proceedings of the EAGLES Workshop on Standards and Evaluation Pisa, Italy, 1999*.

Hsu, F.-Y., Lee, H.-M., Chang, T.-H., & Sung, Y.-T. (2018). Automated estimation of item difficulty for multiple-choice tests: An application of word embedding techniques. *Information Processing & Management*, *54*(6), 969–984. https://doi.org/10.1016/j.ipm.2018.06.007

Irvine, S. H., & Kyllonen, P. C. (2002). *Item generation for test development*. Erlbaum.

Jenkins, H. M., & Michael, M. M. (1986). Using and interpreting item analysis data. *Nurse Educator*, *11*(1), 10.

Jouault, C., Seta, K., & Hayashi, Y. (2016). Content-dependent question generation using LOD for history learning in open learning space. *New Generation Computing*, *34*(4), 367–394. https://doi.org/10.1007/s00354-016-0404-x

Kane, M. (2013). The argument-based approach to validation. *School Psychology Review*, *42*(4), 448–457. https://doi.org/10.1080/02796015.2013.12087465

Kehoe, J. (1995). Basic item analysis for multiple-choice tests. *Practical Assessment, Research, and Evaluation*, *4*(10), 1–3. https://doi.org/10.7275/07zg-h235

Kim, S.-H., Cohen, A. S., & Eom, H. J. (2021). A note on the three methods of item analysis. *Behaviormetrika*, *48*(2), 345–367. https://doi.org/10.1007/s41237-021-00131-1

Kirasich, K., Smith, T., & Sadler, B. (2018). Random forest vs logistic regression: Binary classification for heterogeneous datasets. *SMU Data Science Review*, *1*(3), 9.

Kober, N., & Rentner, D. S. (2012). Year two of implementing the common core state standards: States' progress and challenges. *Center on Education Policy*.

Kochmar, E., Vu, D. D., Belfer, R., Gupta, V., Serban, I. V., & Pineau, J. (2022). Automated data-driven generation of personalized pedagogical interventions in intelligent tutoring systems. *International Journal of Artificial Intelligence in Education*, *32*(2), 323–349. https://doi.org/10.1007/s40593-021-00267-x

Kosh, A. E., Simpson, M. A., Bickel, L., Kellogg, M., & Sanford-Moore, E. (2019). A cost–benefit analysis of automatic item generation. *Educational Measurement: Issues and Practice*, *38*(1), 48–53. https://doi.org/10.1111/emip.12237

Kuechler, W. L., & Simkin, M. G. (2010). Why is performance on multiple-choice tests and constructed-response tests not more closely related? Theory and an empirical test. *Decision Sciences Journal of Innovative Education*, *8*(1), 55–73. https://doi.org/https://doi.org/10.1111/j.1540-4609.2009.00243.x

Kumar, V., Boorla, K., Meena, Y., Ramakrishnan, G., & Li, Y.-F. (2018). Automating reading comprehension by generating question and answer pairs. *arXiv*. http://arxiv.org/abs/1803.03664

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*(1), 121–204. https://doi.org/10.1007/s40593-019-00186-y

Kurdi, G., Parsia, B., & Sattler, U. (2017). An experimental evaluation of automatically generated multiple choice questions from ontologies. In M. Dragoni, M.

Poveda-Villalón, & E. Jimenez-Ruiz (Eds.), *OWL: Experiences and Directions – Reasoner Evaluation* (pp. 24–39, Vol. 10161). Springer International Publishing. https://doi.org/10.1007/978-3-319-54627-8_3

Laduca, A., Staples, W. I., Templeton, B., & Holzman, G. B. (1986). Item modelling procedure for constructing content-equivalent multiple choice questions. *Medical Education*, *20*(1), 53–56. https://doi.org/10.1111/j.1365-2923.1986.tb01042.x

Lajkó, M., Horváth, D., Csuvik, V., & Vidács, L. (2022). Fine-tuning gpt-2 to patch programs, is it worth it? *International Conference on Computational Science and Its Applications*, 79–91.

Lane, S., Raymond, M. R., & Haladyna, T. M. (Eds.). (2016). *Handbook of test development,* (2nd ed.). Routledge.

Liang, C., Yang, X., Dave, N., Wham, D., Pursel, B., & Giles, C. L. (2018). Distractor generation for multiple choice questions using learning to rank. *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, 284–290. https://doi.org/10.18653/v1/W18-0533

Liang, C., Yang, X., Wham, D., Pursel, B., Passonneaur, R., & Giles, C. L. (2017). Distractor generation with generative adversarial nets for automatically creating fill-in-the-blank questions. *Proceedings of the Knowledge Capture Conference*, 1–4. https://doi.org/10.1145/3148011.3154463

Lin, C., Liu, D., Pang, W., & Apeh, E. (2015). Automatically predicting quiz difficulty level using similarity measures. *Proceedings of the 8th International Conference on Knowledge Capture*, 1–8. https://doi.org/10.1145/2815833.2815842

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, 74–81.

Lin, J., & Demner-Fushman, D. (2006). Methods for automatically evaluating answers to complex questions. *Information Retrieval*, *9*(5), 565–587. https://doi.org/10.1007/s10791-006-9003-7

Linn, R. L. (2003). Accountability: Responsibility and reasonable expectations. *Educational Researcher*, *32*(7), 3–13. https://doi.org/10.3102/0013189X032007003

Liu, H., Tam, D., Muqeeth, M., Mohta, J., Huang, T., Bansal, M., & Raffel, C. (2022, August). Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. https://doi.org/10.48550/arXiv.2205.05638

Liu, M., Rus, V., & Liu, L. (2017). Automatic Chinese factual question generation. *IEEE Transactions on Learning Technologies*, *10*(2), 194–204. https://doi.org/10.1109/TLT.2016.2565477

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 preprint arXiv:1907.11692*.

Livingston, S. A. (2009). Constructed-response test questions: Why we use them; how we score them. R&D Connections. Number 11. *Educational Testing Service*.

Livingston, S. A. (2013). Item analysis. In *Handbook of test development*. Routledge. https://doi.org/10.4324/9780203874776.ch19

Livingston, S. A. (2018). *Test Reliability—Basic Concepts* (Research Memorandum No. ETS RM–18-01).

Lu, C.-Y., & Lu, S.-E. (2021). A survey of approaches to automatic question generation: From 2019 to early 2021. *Proceedings of the 33rd Conference on Computational Linguistics and Speech Processing (ROCLING 2021)*, 151–162.

Luckin, R., & Cukurova, M. (2019). Designing educational technologies in the age of AI: A learning sciences-driven approach. *British Journal of Educational Technology*, *50*(6), 2824–2838. https://doi.org/https://doi.org/10.1111/bjet.12861

Marrese-Taylor, E., Nakajima, A., Matsuo, Y., & Yuichi, O. (2018). Learning to automatically generate fill-in-the-blank quizzes. *arXiv*. http://arxiv.org/abs/1806.04524

Maurya, K. K., & Desarkar, M. S. (2020). Learning to distract: A hierarchical multi-decoder network for automated generation of long distractors for multiple-choice questions for reading comprehension. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 1115–1124. https://doi.org/10.1145/3340531.3411997

Mayfield, E., & Black, A. W. (2020). Should you fine-tune BERT for automated essay scoring? *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, 151–162. https://doi.org/10.18653/v1/2020.bea-1.15

Mazidi, K., & Nielsen, R. D. (2014). Pedagogical evaluation of automatically generated questions. In S. Trausan-Matu, K. E. Boyer, M. Crosby, & K. Panourgia (Eds.), *Intelligent Tutoring Systems* (pp. 294–299). Springer International Publishing. https://doi.org/10.1007/978-3-319-07221-0_36

McBroom, J., & Paassen, B. (2020). *Assessing the quality of mathematics questions using student confidence scores* [unpublished].

McCarthy, A. D., Yancey, K. P., LaFlair, G. T., Egbert, J., Liao, M., & Settles, B. (2021). Jump-starting item parameters for adaptive language tests. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 883–899. https://doi.org/10.18653/v1/2021.emnlp-main.67

McNamara, D. S., Graesser, A. C., McCarthy, P. M., & Cai, Z. (2014). *Automated evaluation of text and discourse with coh-metrix*. Cambridge University Press.

Mills, C. N., Potenza, M. T., Fremer, J. J., & Ward, W. C. (Eds.). (2019). *Computer-based testing: Building the foundation for future assessments*. Routledge. https://doi.org/10.4324/9781410612250

Mostow, J., Huang, Y.-T., Jang, H., Weinstein, A., Valeri, J., & Gates, D. (2017). Developing, evaluating, and refining an automatic generator of diagnostic multiple choice cloze questions to assess children's comprehension while reading. *Natural Language Engineering*, *23*(2), 245–294. https://doi.org/10.1017/S1351324916000024

Mulla, N., & Gharpure, P. (2023). Automatic question generation: A review of methodologies, datasets, evaluation metrics, and applications. *Progress in Artificial Intelligence*, *12*(1), 1–32. https://doi.org/10.1007/s13748-023-00295-9

Nagy, P. (2000). The three roles of assessment: Gatekeeping, accountability, and instructional diagnosis. *Canadian Journal of Education / Revue canadienne de l'éducation*, *25*(4), 262–279. https://doi.org/10.2307/1585850

Nelson, D. (2004). *The penguin dictionary of statistics*. Penguin Books.

Newton, P. E. (2007). Clarifying the purposes of educational assessment. *Assessment in Education: Principles, Policy & Practice*, *14*(2), 149–170. https://doi.org/10.1080/09695940701478321

Nguyen, H. A., Bhat, S., Moore, S., Bier, N., & Stamper, J. (2022). Towards generalized methods for automatic question generation in educational domains. *European conference on technology enhanced learning*, 272–284.

Niraula, N. B., & Rus, V. (2015). Judging the quality of automatically generated gap-fill question using active learning. *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, 196–206. https://doi.org/10.3115/v1/W15-0623

OECD. (2009). Creating effective teaching and learning environments: First results from TALIS. *Teaching and Learning International Survey, OECD*.

Olney, A. M. (2021). Sentence selection for cloze item creation: A standardized task and preliminary results. *Joint Proceedings of the Workshops at the 14th International Conference on Educational Data Mining*.

Olson, K. (2010). An examination of questionnaire evaluation by expert reviewers. *Field Methods*, *22*(4), 295–318. https://doi.org/10.1177/1525822X10379795

Osterlind, S. J. (1989). Judging the quality of test items: Item analysis. In S. J. Osterlind (Ed.), *Constructing Test Items* (pp. 259–310). Springer Netherlands. https://doi.org/10.1007/978-94-009-1071-3_7

Osterlind, S. J., & Wang, Z. (2017). Item response theory in measurement, assessment, and evaluation for higher education. In C. Secolsky & D. B. Denison (Eds.), *Handbook on measurement, assessment, and evaluation in higher education* (pp. 191–200). Routledge.

Palomba, C. A., & Banta, T. W. (1999). *Assessment essentials: Planning, implementing, and improving assessment in higher education*. Jossey-Bass, Inc.

Panda, S., Palma Gomez, F., Flor, M., & Rozovskaya, A. (2022). Automatic generation of distractors for fill-in-the-blank exercises with round-trip neural machine translation. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, 391–401. https://doi.org/10.18653/v1/2022.acl-srw.31

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). BLEU: A method for automatic evaluation of machine translation. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 311–318. https://doi.org/10.3115/1073083.1073135

Paramythis, A., Weibelzahl, S., & Masthoff, J. (2010). Layered evaluation of interactive adaptive systems: Framework and formative methods. *User Modeling and User-Adapted Interaction*, *20*(5), 383–453. https://doi.org/10.1007/s11257-010-9082-4

Parnami, A., & Lee, M. (2022). Learning from few examples: A summary of approaches to few-shot learning. *arXiv preprint arXiv:2203.04291*. https://doi.org/https://doi.org/10.48550/arXiv.2203.04291

Peters, M. E., Ruder, S., & Smith, N. A. (2019). To tune or not to tune? adapting pretrained representations to diverse tasks. *arXiv preprint arXiv:1903.05987*. https://doi.org/https://doi.org/10.48550/arXiv.1903.05987

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, *21*(140), 1–67.

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. *arXiv*. https://doi.org/10.48550/arXiv.1606.05250

Ramesh, D., & Sanampudi, S. K. (2022). An automated essay scoring systems: A systematic literature review. *Artificial Intelligence Review*, *55*(3), 2495–2527. https://doi.org/https://doi.org/10.1007/s10462-021-10068-2

Rampey, B. D., Dion, G. S., & Donahue, P. L. (2009). NAEP 2008: Trends in academic progress. *National Center for Education Statistics*.

Ras, E., & Brinke, D. J.-t. (2015). Computer assisted assessment. Research into e-assessment. *18th International Conference, CAA*.

Raschka, S. (2018). Model evaluation, model selection, and algorithm selection in machine learning. *arXiv preprint arXiv:1811.12808*. https://doi.org/https://doi.org/10.48550/arXiv.1811.12808

Razzaq, L., & Heffernan, N. T. (2006). Scaffolding vs. hints in the assistment system. In M. Ikeda, K. D. Ashley, & T.-W. Chan (Eds.), *Intelligent tutoring systems* (pp. 635–644). Springer. https://doi.org/10.1007/11774303_63

Rezigalla, A. A. (2022). Item analysis: Concept and application. In M. S. Firstenberg & S. P. Stawicki (Eds.), *Medical education for the 21st century*. IntechOpen. https://doi.org/10.5772/intechopen.100138

Rodríguez Rocha, O., & Faron Zucker, C. (2018). Automatic generation of quizzes from DBpedia according to educational standards. *Companion of the The Web Conference 2018 on The Web Conference 2018 - WWW '18*, 1035–1041. https://doi.org/10.1145/3184558.3191534

Rodriguez-Torrealba, R., Garcia-Lopez, E., & Garcia-Cabot, A. (2022). End-to-end generation of multiple-choice questions using text-to-text transfer transformer models. *Expert Systems with Applications*, *208*, 118258. https://doi.org/10.1016/j.eswa.2022.118258

Romberg, T. A., Collis, K. F., Donovan, B. F., Buchanan, A. E., & Romberg, M. N. (1982). *The development of mathematical problem-solving superitems* (A report on the NIE/ECS Item Development Project). Wisconsin Center for Education Research, The University of Wisconsin Madison, Wisconsin.

Romero, C., & Ventura, S. (2007). Educational data mining: A survey from 1995 to 2005. *Expert Systems with Applications*, *33*(1), 135–146. https://doi.org/https://doi.org/10.1016/j.eswa.2006.04.005

Rudner, L. M. (2010). Implementing the graduate management admission test computerized adaptive test. In W. J. van der Linden & C. A. Glas (Eds.), *Elements of adaptive testing* (pp. 151–165). Springer. https://doi.org/10.1007/978-0-387-85461-8_8

Rydzewski, N. R., Dinakaran, D., Zhao, S. G., Ruppin, E., Turkbey, B., Citrin, D. E., & Patel, K. R. (2024). Comparative evaluation of LLMs in clinical oncology. *NEJM AI*, AIoa2300151.

Sadler, D. R. (1989). Formative assessment and the design of instructional systems. *Instructional Science*, *18*(2), 119–144. https://doi.org/10.1007/BF00117714

Sarrouti, M., Ben Abacha, A., & Demner-Fushman, D. (2020). Visual question generation from radiology images. *Proceedings of the First Workshop on Advances in Language and Vision Research*, 12–18. https://doi.org/10.18653/v1/2020.alvr-1.3

Scarlatos, A., Brinton, C., & Lan, A. (2022, April). Process-BERT: A framework for representation learning on educational process data [arXiv:2204.13607 [cs]]. Retrieved August 7, 2023, from http://arxiv.org/abs/2204.13607

Seo, M., Kembhavi, A., Farhadi, A., & Hajishirzi, H. (2018). Bidirectional attention flow for machine comprehension. *arXiv*. https://doi.org/10.48550/arXiv.1611.01603

Settles, B., T. LaFlair, G., & Hagiwara, M. (2020). Machine learning–driven language assessment. *Transactions of the Association for Computational Linguistics*, *8*, 247–263. https://doi.org/10.1162/tacl_a_00310

Seyler, D., Yahya, M., & Berberich, K. (2017). Knowledge questions from knowledge graphs. *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*, 11–18. https://doi.org/10.1145/3121050.3121073

Shen, J. T., Yamashita, M., Prihar, E., Heffernan, N., Wu, X., Graff, B., & Lee, D. (2021). MathBERT: A pre-trained language model for general NLP tasks in mathematics education. http://arxiv.org/abs/2106.07340

Shneiderman, B., Plaisant, C., Cohen, M. S., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). *Designing the user interface: Strategies for effective human-computer interaction*. Pearson.

Sinha, G., Shahi, R., & Shankar, M. (2010). Human computer interaction. *2010 3rd International Conference on Emerging Trends in Engineering and Technology*, 1–4. https://doi.org/10.1109/ICETET.2010.85

Smith, J. K. (2003). Reconsidering reliability in classroom assessment and grading. *Educational Measurement: Issues and Practice*, *22*(4), 26–33. https://doi.org/10.1111/j.1745-3992.2003.tb00141.x

Snow, R., O'connor, B., Jurafsky, D., & Ng, A. Y. (2008). Cheap and fast–but is it good? Evaluating non-expert annotations for natural language tasks. *Proceedings of the 2008 conference on empirical methods in natural language processing*, 254–263.

Song, L., & Zhao, L. (2017). Question generation from a knowledge base with web exploration. *arXiv*. http://arxiv.org/abs/1610.03807

Soni, S., Kumar, P., & Saha, A. (2019). Automatic question generation: A systematic review. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3403926

Sorokin, A., & Forsyth, D. (2008). Utility data annotation with Amazon Mechanical Turk. *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 1–8. https://doi.org/10.1109/CVPRW.2008.4562953

Sosnowski, T., & Yordanova, K. (2020). A probabilistic conversational agent for intelligent tutoring systems. *Proceedings of the 13th ACM International Conference on Pervasive Technologies Related to Assistive Environments*, *43*, 1–7.

Štěpánek, L., Dlouhá, J., & Martinková, P. (2023). Item difficulty prediction using item text features: Comparison of predictive performance across machine-learning algorithms. *Mathematics*, *11*(19), 4104. https://doi.org/10.3390/math11194104

Strickland, J. C., & Stoops, W. W. (2019). The use of crowdsourcing in addiction science research: Amazon Mechanical Turk. *Experimental and Clinical Psychopharmacology*, *27*, 1–18. https://doi.org/10.1037/pha0000235

Su, Y. (2015). Ensuring the continuum of learning: The role of assessment for lifelong learning. *International Review of Education*, *61*(1), 7–20. https://doi.org/10.1007/s11159-015-9465-1

Tamura, Y., Takase, Y., Hayashi, Y., & Nakano, Y. I. (2015). Generating quizzes for history learning based on wikipedia articles. In P. Zaphiris & A. Ioannou (Eds.), *Learning and Collaboration Technologies* (pp. 337–346). Springer International Publishing. https://doi.org/10.1007/978-3-319-20609-7_32

Thissen, D., Steinberg, L., & Fitzpatrick, A. R. (1989). Multiple-choice models: The distractors are also part of the item. *Journal of Educational Measurement*, *26*(2), 161–176. https://doi.org/10.1111/j.1745-3984.1989.tb00326.x

Tong, S., Huang, Y., Lv, R., Tong, W., Liu, Q., Huang, Z., Su, Y., & Chen, E. (2020). *Which questions have high quality: Question quality evaluation metric based on answer records.*

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., & Lample, G. (2023). LLaMA: Open and efficient foundation language models. https://doi.org/http://arxiv.org/abs/2302.13971

Van Campenhout, R., Hubertz, M., & Johnson, B. G. (2022). Evaluating AI-generated questions: A mixed-methods analysis using question data and student perceptions. In M. M. Rodrigo, N. Matsuda, A. I. Cristea, & V. Dimitrova (Eds.), *Artificial Intelligence in Education* (pp. 344–353, Vol. 13355). Springer International Publishing. https://doi.org/10.1007/978-3-031-11644-5_28

Van Rossum, G., & Drake, F. L. (2009). *Python 3 reference manual*. CreateSpace.

VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, *46*(4), 197–221. https://doi.org/10.1080/00461520.2011.611369

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Venktesh, V., Akhtar, M. S., Mohania, M., & Goyal, V. (2022). Auxiliary task guided interactive attention model for question difficulty prediction. In M. M. Rodrigo, N. Matsuda, A. I. Cristea, & V. Dimitrova (Eds.), *Artificial Intelligence in Education* (pp. 477–489, Vol. 13355). Springer International Publishing. https://doi.org/10.1007/978-3-031-11644-5_39

von Davier, M. (2018). Automated item generation with recurrent neural networks. *Psychometrika*, *83*(4), 847–857. https://doi.org/10.1007/s11336-018-9608-y

von Davier, M. (2019). Training optimus prime, M.D.: Generating medical certification items by fine-tuning OpenAI's GPT2 transformer model. https://doi.org/http://arxiv.org/abs/1908.08594

Wang, Q., Rose, R., Orita, N., & Sugawara, A. (2024). Automated generation of multiple-choice cloze questions for assessing English vocabulary using GPT-turbo 3.5. *arXiv preprint arXiv:2403.02078*. https://doi.org/https://doi.org/10.48550/arXiv.2403.02078

Wang, R., Su, H., Wang, C., Ji, K., & Ding, J. (2019). To tune or not to tune? How about the best of both worlds? *arXiv preprint arXiv:1907.05338*. https://doi.org/https://doi.org/10.48550/arXiv.1907.05338

Wang, S., Liu, Y., Xu, Y., Zhu, C., & Zeng, M. (2021). Want To reduce labeling cost? GPT-3 can help. *arXiv*. http://arxiv.org/abs/2108.13487

Wang, W., Yang, N., Wei, F., Chang, B., & Zhou, M. (2017). Gated self-matching networks for reading comprehension and question answering. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 189–198. https://doi.org/10.18653/v1/P17-1018

Wang, Y., Yao, Q., Kwok, J. T., & Ni, L. M. (2020). Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys*, *53*(3), 1–34. https://doi.org/10.1145/3386252

Wang, Y., Wang, C., Li, R., & Lin, H. (2022). On the usse of BERT for automated essay scoring: Joint learning of multi-scale essay representation. *arXiv*. https://doi.org/10.48550/arxiv.2205.03835

Wang, Z., Lan, A. S., & Baraniuk, R. G. (2021). Math word problem generation with mathematical consistency and problem context constraints. *arXiv*. http://arxiv.org/abs/2109.04546

Wang, Z., Lan, A. S., Nie, W., Waters, A. E., Grimaldi, P. J., & Baraniuk, R. G. (2018). QG-net: A data-driven question generation model for educational content. *Proceedings of the Fifth Annual ACM Conference on Learning at Scale*, 1–10. https://doi.org/10.1145/3231644.3231654

Wauters, K., Desmet, P., & Van Den Noortgate, W. (2012). Item difficulty estimation: An auspicious collaboration between data and judgment. *Computers & Education*, *58*(4), 1183–1193. https://doi.org/10.1016/j.compedu.2011.11.020

Weiss, D. J. (1982). Improving measurement quality and efficiency with adaptive testing. *Applied Psychological Measurement*, *6*(4), 473–492. https://doi.org/10.1177/014662168200600408

Weiss, D. J. (2004). Computerized adaptive testing for effective and efficient measurement in counseling and education. *Measurement and Evaluation in Counseling and Development*, *37*(2), 70–84. https://doi.org/10.1080/07481756.2004.11909751

Weiss, D. J., & Kingsbury, G. G. (1984). Application of computerized adaptive testing to educational problems. *Journal of Educational Measurement*, *21*(4), 361–375. https://doi.org/10.1111/j.1745-3984.1984.tb01040.x

Wiebe, J., Bruce, R., & O'Hara, T. P. (1999). Development and use of a gold-standard data set for subjectivity classifications. *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*, 246–253.

Wright, B. D., & Bell, S. R. (1984). Item banks: What, why, how. *Journal of Educational Measurement*, *21*(4), 331–345.

Wright, R. E. (1995). *Reading and understanding multivariate statistics* (L. G. Grimm & P. R. Yarnold, Eds.). American Psychological Association.

Xue, K., Yaneva, V., Runyon, C., & Baldwin, P. (2020). Predicting the difficulty and response time of multiple choice questions using transfer learning. *Proceedings of the fifteenth workshop on innovative use of NLP for building educational applications*, 193–197.

Yaneva, V., Baldwin, P., Mee, J., et al. (2019). Predicting the difficulty of multiple choice questions in a high-stakes medical exam. *Proceedings of the fourteenth workshop on innovative use of NLP for building educational applications*, 11–20.

Yaneva, V., Ha, L. A., Baldwin, P., & Mee, J. (2020). Predicting item survival for multiple choice questions in a high-stakes medical exam. In N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, & S. Piperidis (Eds.), *Proceedings of the Twelfth Language Resources and Evaluation Conference* (pp. 6812–6818). European Language Resources Association.

Yaneva, V., Jurich, D., Ha, L. A., & Baldwin, P. (2021). Using linguistic features to predict the response process complexity associated with answering clinical MCQs. *Association for Computational Linguistics*.

Yaneva, V., & Von Davier, M. (2023). *Advancing natural language processing in educational assessment* (1st ed.). Routledge. https://doi.org/10.4324/9781003278658

Yang, A. C., Chen, I. Y., Flanagan, B., & Ogata, H. (2021). Automatic generation of cloze items for repeated testing to improve reading comprehension. *Educational Technology & Society*, *24*(3), 147–158.

Zhang, L., & VanLehn, K. (2016). How do machine-generated questions compare to human-generated questions? *Research and Practice in Technology Enhanced Learning*, *11*(1), 7. https://doi.org/10.1186/s41039-016-0031-7

Zilberberg, A., Anderson, R. D., Finney, S. J., & Marsh, K. R. (2013). American college students' attitudes toward institutional accountability testing: Developing measures. *Educational Assessment*, *18*(3). https://doi.org/10.1080/10627197.2013.817153

Appendix A: Llama 2-7B Prompts

1. Begin your response with yes or no. Does the blank in the next sentence assess key concepts from the sentence and would be reasonable to answer

2. Is this a good or bad cloze question based on the following criterion? A good question tests a key concept from the sentence, is reasonable to answer, unambiguous, or specific. A bad question is unreasonable to answer, too broad, ambiguous, or lacks specificity and depth

3. We removed a part of the following sentence to construct a cloze question. We are trying to understand whether the following question is a good or bad question. A Good question is one that tests key concepts from the sentence and would be reasonable to answer. A Bad question is one that asks about an unimportant aspect of the sentence or has an uninteresting answer that can be figured out from the context of the sentence. Is this question a good or bad question?

4. You are a content expert helping us understand the cloze question quality. A good question tests a key concept from the sentence, is reasonable to answer, unambiguous, and specific. A bad question is unreasonable to answer, too broad, ambiguous, or lacks specificity and depth. Is this question good or bad?

Appendix B: Python Code

B.1 Feature Extraction

```
# Libraries Needed

!pip install sentence_transformers
from sentence_transformers import SentenceTransformer, util
import stanza
import nltk

text = data['text_masked'].tolist()

from transformers import pipeline

mask_filler = pipeline("fill-mask", "distilroberta-base")
masked_pred = mask_filler(text, top_k=1)

model = SentenceTransformer('bert-base-uncased')

# Similarity between predicted response and keyed response
embeddings_pred = model.encode(masked_pred_df['token_str'].tolist())
embeddings_answer = model.encode(data['Answer'].tolist())
cosine_pred_answer = util.cos_sim(embeddings_pred, embeddings_answer
    )
pairs_pred_answer = []
for i in range(len(cosine_pred_answer)):
    pairs_pred_answer.append(util.cos_sim(embeddings_pred[i],
    embeddings_answer[i]))

# Similarity between item stem and keyed response
embeddings_stem = model.encode(data['text_new'].tolist())
cosine_stem_answer = util.cos_sim(embeddings_stem, embeddings_answer
    )
pairs_stem_answer = []
for i in range(len(cosine_stem_answer)):
    pairs_stem_answer.append(util.cos_sim(embeddings_stem[i],
    embeddings_answer[i]))

# Parse tree depth
nlp = stanza.Pipeline(lang='en', processors='tokenize,pos,
    constituency')
data['pos'] = data['item_stem'].apply(lambda x: nlp(x))

```

```
for index, row in data.iterrows():
    doc = data['pos'][index]
    for sentence in doc.sentences:
        data['tree'][index] = (sentence.constituency)

for index, row in data.iterrows():
    data['tree_depth'][index] = Tree.fromstring(data['tree'][index])
   .height()
```

Listing B.1: Feature Extraction for ML Classifier Training

B.2 Machine Learning Classifier Training

```
# Libraries Needed

from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import RandomizedSearchCV,
    train_test_split

# Random Forest

n_estimators = [int(x) for x in np.linspace(start = 10, stop = 200,
    num = 10)]
max_features = ['auto', 'sqrt']
max_depth = [int(x) for x in np.linspace(2, 4, num = 2)]
min_samples_split = [2, 3, 4]
min_samples_leaf = [2, 3]
criterion = ['gini', 'entropy']
random_grid = {'n_estimators': n_estimators,
               'max_features':max_features,
               'max_depth': max_depth,
               'min_samples_split': min_samples_split,
               'min_samples_leaf': min_samples_leaf,
               'criterion': criterion,


         }

rf = RandomForestClassifier(random_state = 42)
rf_random = RandomizedSearchCV(estimator = rf,
                               param_distributions = random_grid,
                               cv = 5, verbose=2, n_jobs = -1)
rf_random.fit(X_train, y_train)

# Support Vector Machine

kernel = ['linear','poly', 'rbf', 'sigmoid']
C = [100, 10, 1, .01, .001]
random_grid = {'C':C,
               'kernel':kernel }

svm = SVC()
svm_random = RandomizedSearchCV(estimator = svm,
                                param_distributions = random_grid,
                                cv = 5, verbose=2, n_jobs = -1)
svm_random.fit(X_train, y_train)

# Logistic Regression

solver  = ['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga']
penalty = ['none', 'l1', 'l2', 'elasticnet']
C = [100, 10, 1, .01, .001]
random_grid = {'solver': solver,
```

```
                'C':C,
                'penalty':penalty }

lr = LogisticRegression(random_state = 42)
lr_random = RandomizedSearchCV(estimator = lr,
                               param_distributions = random_grid,
                               cv = 5, verbose=2, n_jobs = -1)
lr_random.fit(X_train, y_train)
```

Listing B.2: ML Classifier Training

B.3 Fine-Tuning BERT Models

```
# Libraries needed

import datasets
from datasets import Dataset, DatasetDict
from transformers import AutoTokenizer
from transformers import AutoModelForSequenceClassification
from transformers import TrainingArguments, Trainer
import evaluate
import torch

# BERT-Large

dataset_train = datasets.Dataset.from_pandas(train)
dataset_test = datasets.Dataset.from_pandas(test)

dataset_train = dataset_train.remove_columns('__index_level_0__')
dataset_test = dataset_test.remove_columns('__index_level_0__')
data_dict = datasets.DatasetDict({"train":dataset_train,"test":
    dataset_test})

device = torch.device("cuda") if torch.cuda.is_available() else
    torch.device("cpu")

tokenizer = AutoTokenizer.from_pretrained("bert-large-uncased")

def tokenize_function(examples):
    return tokenizer(examples['sentence1'],
    padding="max_length", truncation=True, max_length=250)

tokenized_datasets = data_dict.map(tokenize_function, batched=True)

model = AutoModelForSequenceClassification.from_pretrained
                        ("bert-large-uncased", num_labels=2).to(
    device)

training_args = TrainingArguments(
    output_dir="my_awesome_model",
    learning_rate=2e-5,
    per_device_train_batch_size=8,
    per_device_eval_batch_size=16,
    num_train_epochs=10,
    weight_decay=0.01,
    evaluation_strategy="epoch",
    save_strategy="epoch")

metric = evaluate.load("accuracy")

def compute_metrics(eval_pred):
    logits, labels = eval_pred
    predictions = np.argmax(logits, axis=-1)
    return metric.compute(predictions=predictions, references=labels
```

```
    )

trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=tokenized_datasets["train"],
    eval_dataset=tokenized_datasets["test"],
    compute_metrics=compute_metrics
)

trainer.train()

# BERT-Base

dataset_train = datasets.Dataset.from_pandas(train)
dataset_test = datasets.Dataset.from_pandas(test)

dataset_train = dataset_train.remove_columns('__index_level_0__')
dataset_test = dataset_test.remove_columns('__index_level_0__')
data_dict = datasets.DatasetDict({"train":dataset_train,"test":
    dataset_test})

device = torch.device("cuda") if torch.cuda.is_available() else
    torch.device("cpu")

tokenizer = AutoTokenizer.from_pretrained("bert-base-uncased")

def tokenize_function(examples):
    return tokenizer(examples['sentence1'],
    padding="max_length", truncation=True, max_length=250)

tokenized_datasets = data_dict.map(tokenize_function, batched=True)

model = AutoModelForSequenceClassification.from_pretrained
                        ("bert-base-uncased", num_labels=2).to(
    device)

training_args = TrainingArguments(
    output_dir="my_awesome_model",
    learning_rate=2e-5,
    per_device_train_batch_size=8,
    per_device_eval_batch_size=16,
    num_train_epochs=10,
    weight_decay=0.01,
    evaluation_strategy="epoch",
    save_strategy="epoch")

metric = evaluate.load("accuracy")

def compute_metrics(eval_pred):
    logits, labels = eval_pred
    predictions = np.argmax(logits, axis=-1)
    return metric.compute(predictions=predictions, references=labels
    )
```

```
98
99  trainer = Trainer(
100     model=model,
101     args=training_args,
102     train_dataset=tokenized_datasets["train"],
103     eval_dataset=tokenized_datasets["test"],
104     compute_metrics=compute_metrics
105 )
106
107 trainer.train()
```

Listing B.3: Fine-tuning BERT Models

B.4 Instruction-Tuning Llama 2-7B

```
# Libraries needed

!pip install -q huggingface_hub
!pip install -q -U trl transformers accelerate peft
!pip install -q -U datasets bitsandbytes einops wandb
from datasets import load_dataset
import torch
from transformers import AutoModelForCausalLM,
        BitsAndBytesConfig, AutoTokenizer, TrainingArguments
from peft import LoraConfig
from trl import SFTTrainer
import torch
import transformers




from huggingface_hub import notebook_login
notebook_login()

dataset_name = "GGorgun/mind_the_gap"
dataset = load_dataset(dataset_name, split = 'train')

def label_to_name(score: float):
    if score == 0:
        return "Bad"
    return "Good"

dataset_data = [
    {
        "instruction": "You are a content expert helping us
   understand
        the cloze question quality.
        A good question tests a key concept from the sentence,
        is reasonable to answer, unambiguous, and specific.
        A bad question is unreasonable to answer, too broad,
   ambiguous,
        or lacks specificity and depth. Is this question a good or
   bad question?",
        "input": row_dict["text"],
        "output": label_to_name(row_dict["label"])
    }
    for row_dict in df.to_dict(orient="records")
]

def formatting_func(example):
    input_prompt = (f"Below is an instruction that describes a task,
    paired with an input that provides further context. "
    "Write a response that appropriately completes the request.\n\n"
    "### Instruction:\n"
    f"{example['instruction']}\n\n"
```

```
    f"### Input: \n"
    f"{example['input']}\n\n"
    f"### Response: \n"
    f"{example['output']}")
    return {"text" : input_prompt}

model_id = "meta-llama/Llama-2-7b-hf"

qlora_config = LoraConfig(
    r=16,
    lora_alpha=32,
    lora_dropout=0.05,
    bias="none",
    task_type="CAUSAL_LM"
)

bnb_config = BitsAndBytesConfig(
    load_in_4bit=True,
    bnb_4bit_use_double_quant=True,
    bnb_4bit_quant_type="nf4",
    bnb_4bit_compute_dtype=torch.bfloat16
)

base_model = AutoModelForCausalLM.from_pretrained(
    model_id,
    quantization_config=bnb_config,
)

tokenizer = AutoTokenizer.from_pretrained(model_id,
   trust_remote_code=True)
tokenizer.pad_token = tokenizer.eos_token


output_dir = "./results"

training_args = TrainingArguments(
    output_dir=output_dir,
    per_device_train_batch_size=4,
    gradient_accumulation_steps=4,
    learning_rate=2e-4,
    logging_steps=10,
    max_steps=1000,
    optim="paged_adamw_8bit",
    fp16=True,
)

max_seq_length = 250

supervised_finetuning_trainer = SFTTrainer(
    model=base_model,
    train_dataset=formatted_dataset['train'],
    peft_config=qlora_config,
    dataset_text_field="text",
    #formatting_func=formatting_prompts_func,
```

```
    max_seq_length=max_seq_length,
    tokenizer=tokenizer,
    args=training_args,

)

supervised_finetuning_trainer.train()
```

Listing B.4: Instuction-Tuning Llama 2-7B