BERT-IRT: Accelerating Item Piloting with BERT Embeddings and Explainable IRT Models

Kevin P. Yancey and **Andrew Runge** and **Geoff LaFlair** and **Phoebe Mulcaire**
Duolingo
5900 Penn Ave
Pittsburgh, PA 15206
{kyancey,arunge,geoff,phoebe}@duolingo.com

Abstract

Estimating item parameters (e.g., the difficulty of a question) is an important part of modern high-stakes tests. Conventional methods require lengthy pilots to collect response data from a representative population of test-takers. The need for these pilots limit item bank size and how often those item banks can be refreshed, impacting test security, while increasing costs needed to support the test and taking up the test-taker's valuable time. Our paper presents a novel explanatory item response theory (IRT) model, BERT-IRT, that has been used on the Duolingo English Test (DET), a high-stakes test of English, to reduce the length of pilots by a factor of 10. Our evaluation shows how the model uses BERT embeddings and engineered NLP features to accelerate item piloting without sacrificing criterion validity or reliability.

1 Introduction

The Duolingo English Test (DET) is a test of English language proficiency that is used for admissions decisions in English medium universities. It measures the four skills of speaking, writing, reading, and listening. It is delivered remotely to test-takers' computers via a desktop application, and it can be taken any time and at any appropriate location with a strong enough internet connection. The DET's value proposition to test-takers is that it is affordable, short in duration, and has a short score reporting turn-around time.

The DET accomplishes this, in part, by using computer adaptive test (CAT) administration to more quickly and accurately estimate test-takers' language proficiency (Cardwell et al., 2022). A computer adaptive test (CAT) uses item parameter estimates to adapt to each test-taker by finding items that will yield maximal information about their proficiency based on how well they've done so far. Item banks for CATs must be very large to ensure that test-takers do not have preknowledge of items (LaFlair et al., 2022; Way, 1998), and they also require high-quality item parameter estimates to ensure that items are selected for administration accurately.

Typically, item parameters are estimated from hundreds of responses for each item collected via pilots. However, these pilots take up the test-taker's valuable time and increase the costs for the assessment, thus limiting the rate at which new items can be added to the bank. Explanatory frameworks that estimate item parameters from item features have been around for a long time, starting with Fischer (1973)'s Log Linear Traits Model (LLTM), and have a rich literature (De Boeck, 2004). These frameworks can be used to help reduce or eliminate the need for item piloting by leveraging item features to estimate item parameters more accurately with less response data. This can have positive downstream effects on test security and on test-takers. For security, it allows for test developers to add to, or replace, their item banks at very high rates, which helps to ensure unique administrations of tests and reduce the effects of item preknowledge. For test-takers, it reduces the amount of time they spend responding to unscored test items during pilots and reduces the costs of test development. These cost savings can be passed on to test-takers and even help lower barriers for less economically advantaged test-takers.

It is well known in the NLP literature (Tenney et al., 2019; Jawahar et al., 2019) that pretrained language models such as BERT (Devlin et al., 2019) learn text representations that represent highly general linguistic properties of words that are useful for a wide range of tasks, including estimating the difficulty of text for L2 learners (Yancey et al., 2021). More recent work has explored using these text embeddings in explanatory IRT models to predict parameters for test items. For example, Benedetto et al. (2021) finetuned

BERT to predict difficulty using datasets of educational questions and real student responses, and Byrd and Srivastava (2022) combined contextual embeddings from BERT with additional manually curated features to predict difficulty and discrimination for general knowledge questions. Similar work has used BERT to predict the difficulty of multiple-choice questions (Reyes et al., 2023) and programming problems (Zhou and Tao, 2020).

One example of using explanatory models this way is described in our previous work, McCarthy et al. (2021), which proposed using BERT embeddings in a multi-task explanatory item response theory (IRT) framework, called BERT-LLTM, to estimate the item parameters of c-test tasks, a task typically used to assess L2 language proficiency. This work introduces a new model, BERT-IRT, which makes several improvements to this approach:

- BERT-LLTM estimated passage-level difficulty and discrimination. BERT-IRT estimates these at the word level, which greatly improves criterion validity and reliability.
- The accuracy of BERT-LLTM's parameter estimates is limited by how well the features predict those parameters, even for items that have enough observed responses that non-explanatory IRT models could produce more accurate estimates. BERT-IRT achieves the best of both worlds by using residual weights, which allows it to refine the parameter estimates derived from features based on response data that has been collected for each item in a manner similar to Bayesian updating.
- BERT-IRT incorporates engineered NLP features that substantially increase the accuracy of the model's parameter estimates.

In addition to the offline evaluation on historical data, we present the results of using this model to shorten pilots by a factor of 10 on a real-world high-stakes test of English for L2 learners.

2 Background: Language Assessment

First, we will provide a brief overview of the relevant concepts from language assessment research.

2.1 Item Response Theory (IRT)

Item Response Theory (IRT; (Lord, 2012)) is essential for most modern high-stakes tests, and for Computer Adaptive Tests (CAT; (Weiss, 1982; Van der Linden and Glas, 2010)) in particular. IRT models are statistical models that are used to improve the time-efficiency and accuracy of assessment by modeling item characteristics (called "parameters") that affect the probability of test-takers of different proficiency levels responding to that item correctly. One of the most common IRT models is the 2PL model (Hambleton et al., 1991), which models both the relative difficulty of an item and how well an item discriminates between high and low proficiency test-takers. IRT models are used to quantify how informative an item will be for a given test-taker (i.e., by computing its Fisher information), which is used by CAT algorithms to increase the efficiency of the test. Additionally, IRT models are used to produce scores from CAT algorithms by computing the expected-a-posteriori (EAP) or maximum-a-posteriori (MAP) of the test-taker's latent proficiency based on the test-taker's observed responses to items and the estimated parameters for those items (Van der Linden and Glas, 2010).

2.2 Validity & Reliability

Validity and reliability are two key concepts in assessing the quality of scores (Furr, 2021), which are the main product of an assessment. Validity refers to the degree to which the score measures its intended "construct" (i.e., what it's intended to measure). One common piece of validity evidence is criterion validity, which is the test score's correlation with other known measures of the same or similar construct. Reliability is the consistency of the score. This is often measured by taking the correlation between retests by the same test-taker (i.e., test-retest reliability).

2.3 The C-Test Task Type

This paper focuses on estimating item parameters of c-test tasks for L2 learners of English. C-tests are reading tasks that measure test-takers' general language ability (Norris, 2018). As shown in Figure 1, each c-test task is composed of a paragraph in which some of the words are damaged by removing the second half of the word. Specifically, the first and last sentences of the passage are left intact to provide context, but every other word of the intermediary sentences is damaged. The test-takers' task is to complete all of the damaged words. Research on c-tests has shown that test-taker performance on these tasks correlates with overall language proficiency test scores (Daller et al., 2021), measures of reading ability (Khodadady, 2014; Klein-Braley, 1997), as well as vocabulary, and grammatical knowledge (Eckes and Grotjahn, 2006; Karimi, 2011; Khodadady, 2014).

2.4 Testlets

In our IRT model, we treat each damaged word as a distinct item with its own parameters. This essentially makes each c-test task a testlet (Wainer et al., 2007), where multiple items are administered together and share a common context (i.e., the passage). In our internal evaluation, we found that treating each damaged word as a distinct item dramatically increased criterion validity and reliability, as the IRT model was able to account for the differences in difficulty and discrimination among words within the passage. Specifically, using the Spearman-Brown prophecy formula (Allen and Yen, 2001), we found that we would have to add 25 % more c-test passages to each test session in order to achieve the same increase in test-retest reliability without using testlet scoring.

3 Model

In the following sub-sections, we explain the BERT-IRT model in detail, starting with explaining the standard 2PL IRT model in Section 3.1 and then extending it with an explanatory framework in Section 3.2. We then discuss the BERT-IRT model's features in Section 3.3, before finally explaining the training process in Section 3.4.

3.1 The Standard 2PL IRT Model

We start by formally defining the standard 2PL model, which is extended by our BERT-IRT model. In the 2PL model, the probability that a test-taker with proficiency θ_p will get item i correct depends on two item parameters:

- The intercept, denoted d_i, that models the logit-probability that a test-taker with average ability will answer the item correctly. This measures how easy or difficult the item is.

- The slope, denoted a_i, that defines how much that logit-probability changes depending on a test-taker's proficiency. This measures how discriminative the item is.

With these two item parameters, the 2PL model defines the probability of test-taker p getting item i correct as:

$$P(Y_{p,i} = 1) = f_{\text{logistic}}\left(d_i + a_i\theta_p\right)$$

where $Y_{p,i} \in \{0, 1\}$ is the test-taker's grade on the item.

3.2 Explanatory IRT Framework

In the standard 2PL model, each item parameter would be estimated by finding the values that best predict the observed responses for that item. As in other explanatory IRT frameworks, BERT-IRT extracts features from items and uses those features to predict item parameters as functions of those features. This has two key advantages:

1. This can reduce the amount of response data needed to estimate accurate parameters.

2. This allows one to estimate item parameters for novel items for which no response data has been collected.

However, for an item with many observed responses, explanatory IRT models may produce less accurate item parameter estimates than what could be achieved by non-explanatory IRT models, due to variance in item parameters that are not explained by the features. To overcome this, BERT-IRT uses residual weights to adjust the item parameter estimates of each item based on the observations for that particular item.

BERT-IRT uses a set of K item features to estimate a_i and d_i. Let $X_{i,k} \in \mathbb{R}$ denote the value of the k-th feature for item i where $X_{i,0}$ is a constant such that $X_{i,0} = 1$ for all i.

An item's intercept parameter, d_i, is thus modeled as a linear function of the item's features, X_i, plus the item-specific residual, denoted $\varepsilon_{d,i}$. The equation for d_i thus becomes:

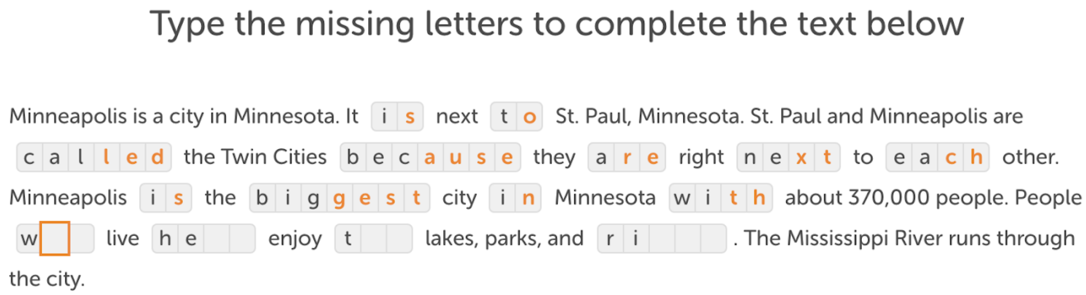
$$d_i = \varepsilon_{d,i} + \sum_{k=0}^{K} \upsilon_k X_{i,k}$$

where $\upsilon \in \mathbb{R}^{K+1}$ is a vector consisting of the bias term, υ_0, and the feature weights.

Slope parameters are defined similarly, but use a log-linear framework. The formula for slope parameters is thus:

$$a_i = \exp\left(\varepsilon_{a,i} + \sum_{k=0}^{K} \beta_k X_{i,k}\right)$$

where $\beta \in \mathbb{R}^{K+1}$ is the vector consisting of the bias term and feature weights, and $\varepsilon_{a,i}$ is the residual weight. The log-linear framework is often



Type the missing letters to complete the text below

Minneapolis is a city in Minnesota. It is next to St. Paul, Minnesota. St. Paul and Minneapolis are called the Twin Cities because they are right next to each other. Minneapolis is the biggest city in Minnesota with about 370,000 people. People w□□ live he□□ enjoy t□□ lakes, parks, and ri□□□. The Mississippi River runs through the city.

Figure 1: Example C-Test Item

closer to the true relationship between the slope parameters and the item features, has nicer convergence properties, and enforces that slope parameters are positive.

3.3 Model Features

Most of the features used by BERT-IRT are extracted from the pretrained BERT model by feeding in the undamaged passage (i.e., the passage without letters omitted from the damaged words). Two embeddings for each item are used as features:

Passage Embedding (n=768) - This is computed as the average of the embeddings extracted for each token in the passage from BERT's 11th layer.

Contextual Word Embedding (n=3,072) - This is computed by concatenating the token's embeddings from the first four layers of BERT. If the damaged word corresponds to multiple BERT tokens, then the embeddings for the applicable tokens are averaged.

Various alternative methods for encoding ctest items were evaluated in preliminary experiments, and this approach was found to be among the best. In particular, we found that using the lowest four layers of BERT to produce contextual word embeddings outperformed using higher layers. We believe this is because lower layers are better able to encode surface-level information, such as word frequency (Jawahar et al., 2019; Li et al., 2021), that are often important to predicting L2 difficulty (François and Fairon, 2012).

In addition, BERT-IRT uses 15 engineered NLP features shown to correlate strongly with c-test item parameters, specifically:

- The log frequency of the damaged word in the Corpus of Contemporary American English (COCA) (Davies, 2008)
- The log frequency of the word in COCA across the 8 sub-corpora (8 features)
- The log document frequency of the damaged word in the COCA corpus
- The length of the answer key (i.e., the number of letters the test-taker must fill in)
- The proportion of vowels in the answer key
- The average log frequency in COCA of each word in the c-test passage
- The position of the damaged word within the passage, normalized by the passage's length
- The conditional probability of the correct word, given the damaged word, derived using COCA frequencies (e.g. if the damaged word is "pass___" and the correct word is "passage", how frequently does that word occur versus alternative solutions such as "passing" vs. "passers" etc.)

3.4 Model Training

To estimate the model weights,[1] we need a training dataset of graded responses from test-takers. This consists of a set of test-taker responses represented as tuples of item, i, test-taker, p, and grade, $g \in \{0, 1\}$. We essentially use gradient descent to perform maximum-a-posteriori (MAP) estimation of the model weights given the observed response data. Details are provided in the subsections below.

3.4.1 Model Weights

The model has four vectors of weights that must be estimated: the intercept bias and feature weights vector, $\upsilon \in \mathbb{R}^{K+1}$, the intercept residuals vector,

[1] Here, we refer to all of the model's learnable parameters as weights to avoid them being conflated item parameters.

$\varepsilon_d \in \mathbb{R}^I$, the slope bias and feature weights vector, $\beta \in \mathbb{R}^{K+1}$, and slope residuals vector, $\varepsilon_a \in \mathbb{R}^I$, where I denotes the number of items in the training dataset.

3.4.2 Theta Estimates

Since our response data is collected as part of a high-stakes test of English, we can compute accurate estimates for test-taker proficiency based on their performance on items whose parameters are not being estimated (i.e., the section scores for item types other than c-test). We use these as fixed estimates for θ_p during model training. In other piloting designs where this is not possible, we could treat these proficiencies as weights to be estimated jointly with the other model weights, but that would require larger quantities of response data to achieve comparable performance results.

3.4.3 Regularization

To avoid the model being underidentified, the residual weights must be regularized. We apply L2 regularization to these parameters. Optimizing the strength of those L2 penalties is important: if the L2 penalties are set too low then the model won't generalize to new items as well as it could, and if they are set too high the model will predict item parameters for items with many observations less accurately than it could. In this context, these L2 penalties are equivalent to using Gaussian priors with zero means. The optimal penalty for intercept residuals would be $0.5/\sigma^2_{\varepsilon_d}$, where $\sigma^2_{\varepsilon_d}$ is the variance in the intercepts that is *not* explained by the features. The optimal penalty for slope residuals is likewise. Thus, we treat $\hat{\sigma}^2_{\varepsilon_d}$ and $\hat{\sigma}^2_{\varepsilon_a}$ as hyperparameters, and set the penalties for intercept residuals and slope residuals to $0.5/\hat{\sigma}^2_{\varepsilon_d}$ and $0.5/\hat{\sigma}^2_{\varepsilon_a}$, respectively.

Since there are many features, we also use L2 regularization on the feature weights. Following the same convention, we set the coefficients of these penalties as $0.5/\hat{\sigma}^2_{\beta}$ and $0.5/\hat{\sigma}^2_{\upsilon}$, respectively, treating $\hat{\sigma}^2_{\beta}$ and $\hat{\sigma}^2_{\upsilon}$ as hyperparameters.

3.4.4 Training Objective

During training, we initialize all weights to zero and use gradient descent to estimate values for the model weights that maximize their log posterior-probability given the test-taker responses in the training dataset, D. The objective function to be maximized is thus specified as follows:

$$\sum_{(i,p,g)\in D} LL(\Phi \mid Y_{p,i} = g) - \frac{0.5}{\hat{\sigma}^2_{\upsilon}} \sum_{k=1}^{K} \upsilon_k^2 \\ - \frac{0.5}{\hat{\sigma}^2_{\beta}} \sum_{k=1}^{K} \beta_k^2 - \frac{0.5}{\hat{\sigma}^2_{\varepsilon_a}} \sum_{i=1}^{I} \varepsilon_{a,i}^2 - \frac{0.5}{\hat{\sigma}^2_{\varepsilon_d}} \sum_{i=1}^{I} \varepsilon_{d,i}^2$$

where Φ denotes the set of weight vectors being estimated (β, υ, ε_a, and ε_d) and LL is the log likelihood function:

$$LL(\Phi \mid Y_{p,s}) = g \cdot \ln P(Y_{p,i} = 1) \\ + (g-1) \cdot \ln(1 - P(Y_{p,s} = 1))$$

3.4.5 Tuning Hyperparameters

The large search space resulting from four hyperparameters and long training times makes tuning hyperparameters difficult. For our experiments, we used a sparse grid search to find acceptable values for hyperparmeters. Since the optimization of the residual hyperparameters requires evaluating how well the model predicts both novel and seen items, we ensured that the training and evaluation datasets were split in such a way that the evaluation dataset included both items that occurred in the training dataset and items that did not.

We found that even a limited search of the hyperparameter space produced good results. However, there are methods that could be applied to the training data to estimate $\sigma^2_{\varepsilon_a}$ and $\sigma^2_{\varepsilon_d}$ directly. These include maximizing the marginal likelihood function, maximizing an approximation to the marginal likelihood function, and fully Bayesian methods implemented via Markov Chain Monte Carlo (Dey et al., 1997; Lindstrom and Bates, 1990; Pinheiro and Bates, 1995; Wolfinger, 1993). Future work could consider the application of these methods.

4 Experiments

Here we present a series of four experiments to evaluate BERT-IRT using data from the Duolingo English Test, a high-stakes test of English for L2 learners. In the first experiment, we use offline evaluation to analyze the model's performance when piloting a new item bank from scratch (i.e., what we refer to as a "fast-start" scenario). In the second experiment, we analyze BERT-IRT's ability to generalize item parameter predictions to unseen items under various conditions. In the third experiment, we investigate how much each feature contributes to the estimation of item parameters. Finally, in the forth experiment, we analyze BERT-IRT's ability to leverage response data from an existing item bank to make predictions for new items with limited piloting data available (i.e., what we refer to as a "jump-start" scenario). As part of this, we discuss the results of using BERT-IRT to add new items to the test's item bank with only a tenth of the normal amount of pilot data.

4.1 Experiment 1. Offline Evaluation in a Fast-Start Scenario

In this experiment, we do an ablation study to evaluate how the model performs when only a limited amount of response data is available for each item. Traditionally, a new item bank would be piloted until 200 observations per item are collected (the minimum needed for reasonably accurate item parameters in an unregularized 2PL model). However, these pilots can be costly and time-consuming, so with BERT-IRT we hope to be able to achieve similar or better performance with much shorter pilots.

For this experiment, we retrieved around a year's worth of historical response data from the test. The dataset included around 3,000 c-test passages with around 50,000 unique items. The unablated dataset had around 600 observations per item, which were split into train and evaluation datasets. The training dataset was sampled to produce ablated training datasets with observation counts of 5, 10, 20, 40, 80, 160, and 200 observations per item.

We compared BERT-IRT to two baselines:

Post-Pilot Operational 2PL - A non-explanatory 2PL model trained on 200 responses per item (i.e., the minimal number of responses per item collected during a standard pilot). This simulates the performance of using the test's operational 2PL model on items that have only recently been created and piloted.

Regularized 2PL - A non-explanatory 2PL model trained on the same ablated datasets as BERT-IRT, where the item parameters are estimated via maximum-a-posteriori (MAP) estimation using a Gaussian prior on each parameter. This regularization is used because unregularized 2PL models will yield very poor results when trained on fewer than 200 responses per item.

We then used those trained models to produce probabilities and scores on the evaluation dataset, which we evaluated using the following metrics:

Cross-Entropy - The cross-entropy between observed binary grades and their probability as predicted by the IRT model. This measures how well the model predicts the probability of the test-taker responding to an item correctly.

Item Mean Grade R - The Pearson correlation between each item's observed mean grade in the response dataset and it's predicted mean-grade according to the IRT model. This mainly measures the IRT model's ability to predict the relative difficulty of each item.

Test-Retest Correlation - The Pearson correlation between c-test scores produced by the IRT model for any two test sessions taken by the same test-taker within 30 days of each-other. This is a well established measure of score reliability in the assessment research literature (Furr, 2021).

Internal Validity Coefficient - The Pearson correlation between the c-test score produced by the IRT model, and the score aggregated from other sections of the test (using their original scoring methods). This is a common measure that is used in the assessment research literature (Furr, 2021) to measure criterion validity.

The results are shown in Figure 2. These plots show that the BERT-IRT model always outperformed the regularized 2PL model regardless of the number of responses available for training. Furthermore, these results show that the BERT-IRT model can achieve similar or better performance than the operational 2PL model with as few as 50 responses per item, representing a *4X increase* in piloting efficiency. The only metric on which BERT-IRT did not outperform the Post-Pilot Operational 2PL baseline was the Internal Validity Coefficient. However, given that this is the case even when BERT-IRT's test-retest reliability is higher, this could indicate the BERT-IRT is finding parameters that better represent aspects of the construct that are specific to c-test items. This could increase test-retest reliability by more accurately measuring the skills needed to answer c-test items, but lower internal validity because the skills measured by c-tests are slightly different than those measured

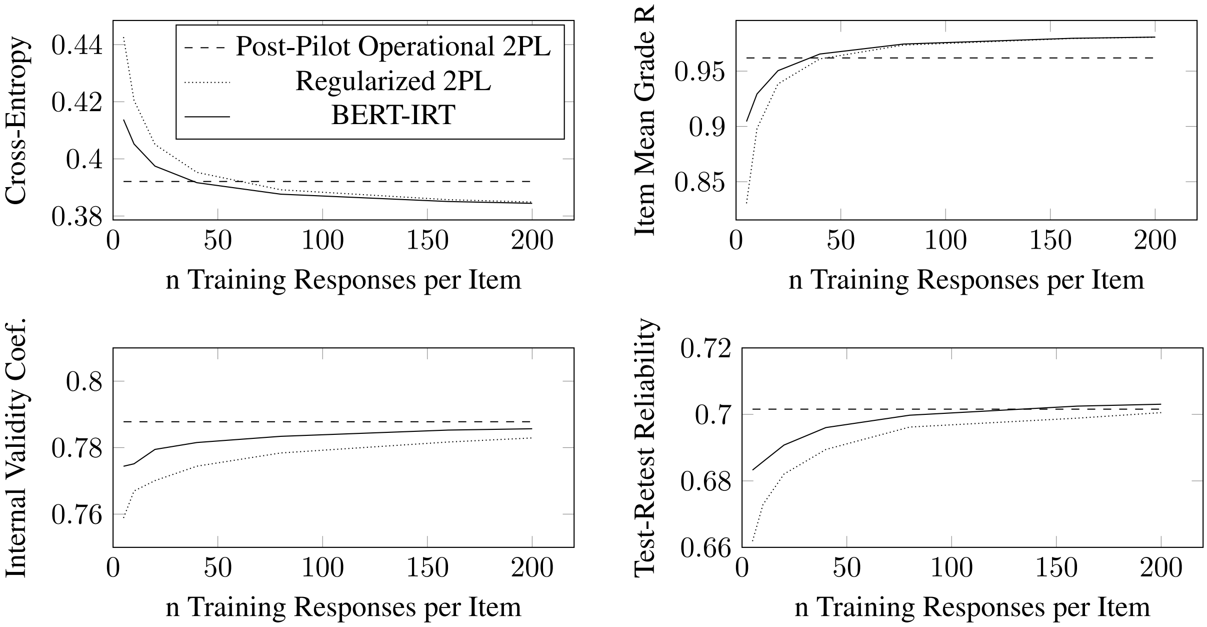


Figure 2: Experiment 1. Evaluation in a Fast-Start Scenario

by other task types. In any case, the difference in internal validity is very small (0.78 vs 0.79).

4.2 Experiment 2. Generalization Experiments

To better understand how the model generalizes parameter estimates across items, we experiment with different splits of the same dataset used in Experiment 1. The splits we used are defined below:

Test-Taker - Response data is split such that all responses for an individual test-taker are assigned to either the training or evaluation datasets. This simulates a fast-start scenario, as in Experiment 1. Since all items occur in training, this is essentially a baseline indicating the ceiling of what should be possible.

Testlet - Response data is split such that all responses to a given c-test passage (i.e., testlet) are assigned to either the training or evaluation datasets. This simulates a jump-start scenario, whereby responses for an existing item bank are used to estimate parameters for new items that have little or no pilot data.

Item - Response data is split such that all responses to a given item are assigned to either training or evaluation datasets. This investigates how well the model can predict item parameters for words in a passage, when there is significant response data for other words in the same passage. This might be useful if one wanted to change which words in a passage are damaged based on its predicted item parameters in order to adjust the c-test passage's difficulty or increase is informativeness.

Split	**Cross-Entropy**	**Item Mean Grade R**
Test-Taker	0.38	0.98
Testlet	0.43	0.88
Item	0.42	0.89
Stem	0.52	0.76

Table 1: Comparison of BERT-IRT item parameter estimates when trained on 20 vs 200 responses.

Stem - Responses data is split such that all responses for items that share a word stem are assigned to either training or evaluation datasets. For example, items for "work", "worked", and "works" would all be put on the same side of the split. For this purpose, we used the Snowball Stemmer from NLTK (Porter, 1980; Bird et al., 2009). This evaluates how well the model generalizes to items assessing previously untested words.

In all cases, we use roughly 80 % of the data for training and 20 % for evaluation. Since under these data splits, individual sessions are split across training and evaluation datasets, its not possible to compute scores for sessions using just evaluation data. Hence, for this experiment we use only metrics that can be computed for individual item responses: Cross-Entropy and Item Mean Grade R.

The results are shown in Table 1. In the baseline split, the model almost perfectly predicts the mean grade of each item over the evaluation dataset, with a correlation of 0.98. The testlet and item splits shows that BERT-IRT generalizes very well to unseen items, predicting the mean grades of unseen items with a correlation of 0.88.

Notably, as shown by the stem split, the model's ability to predict mean grades for a item degrades significantly when that item has a novel stem that the model did not see in training. This shows that the item's word stem explains a significant amount of the variance in the item's parameters. This is a very useful property when jump-starting item parameters using BERT-IRT, because, due to Zipf's law, if the existing item bank is sizeable, most items of newly-created c-test passages will likely share a word-stem with an existing item from the existing bank. However, this means items with novel word stems will likely have less accurate item parameter estimates until sufficient response data for them can be collected.

4.3 Experiment 3. Feature Contributions

To better understand the contributions of various features, we evaluated the importance of each feature using SHAP values (Lundberg and Lee, 2017). In the BERT-IRT model, the features only affect the item parameter estimates through a linear combination defined by the weight vectors υ and β. As such, we compute the SHAP values using the same methods as would be used for linear models using those weight vectors. To account for correlations among features, we compute *observational* SHAP values. From these we compute the feature importance for each feature as the mean absolute SHAP value over all items, and then normalize the resulting feature importances to sum to 1. Since embeddings consist of hundreds of features that would be impractical to list individually, we summarize their importances by summing the embedding feature SHAP values for a given item before taking the absolute value and averaging across items. We also summarize the 8 genre-specific word-frequency features the same way.

The results are shown in Figure 3. The features are presented in the same order as in Section 3.3. For predicting both intercept parameters and log slope parameters, the word embedding is very important, contributing 28 % and 40 % of the prediction, respectively. By comparison, passage embeddings are a relatively weak predictor, contributing only 3 % and 8 % of the prediction, respectively. The word frequency features are also a very important predictor, contributing even more than the word embedding does for predicting intercepts.

Features	**Cross-Entropy**	**Item Mean Grade R**
All Features	0.43	0.88
Embeddings	0.44	0.84
Engineered	0.48	0.69

Table 2: Comparison of BERT-IRT performance on the Testlet split when using different feature sets.

Additionally, we did an ablation study by repeating the Testlet split experiment from Experiment 2, but using only embedding features or only engineered features (see Table 2). These results show that while the embedding features perform quite well on their own, both sets of features complement each other to yield superior results.

4.4 Experiment 4. Online & Offline Evaluation in a Jump-Start Scenario

In this experiment, we evaluate how well BERT-IRT can estimate item parameters for a new pool of c-test items with only a very short pilot, when leveraging large amounts of response data from an existing item bank to learn the relationships between the item features and item parameters.

To test this scenario, we generated 1,039 new c-test passages with GPT-3 (Brown et al., 2020), and piloted them on the test, with each test session being randomly assigned one unscored pilot c-test task in addition to its normal 4 scored c-test tasks. We ran the pilot until we had collected around 20 responses per item. We trained BERT-IRT on both the response data from the existing bank and the pilot, and estimated the parameters for all the new items. In an offline evaluation, we showed that even if we'd used the existing BERT-IRT parameter estimates to score the pilot c-test tasks instead of one of the other 4 operational c-test tasks, criterion validity and reliability would have been negligibly affected. Based on that offline evaluation, we added the new c-test tasks to the operational bank, replacing roughly a third of the existing c-test item bank with only a tenth of the piloting time that would have otherwise been required. Furthermore, analyses of the test following the item bank change

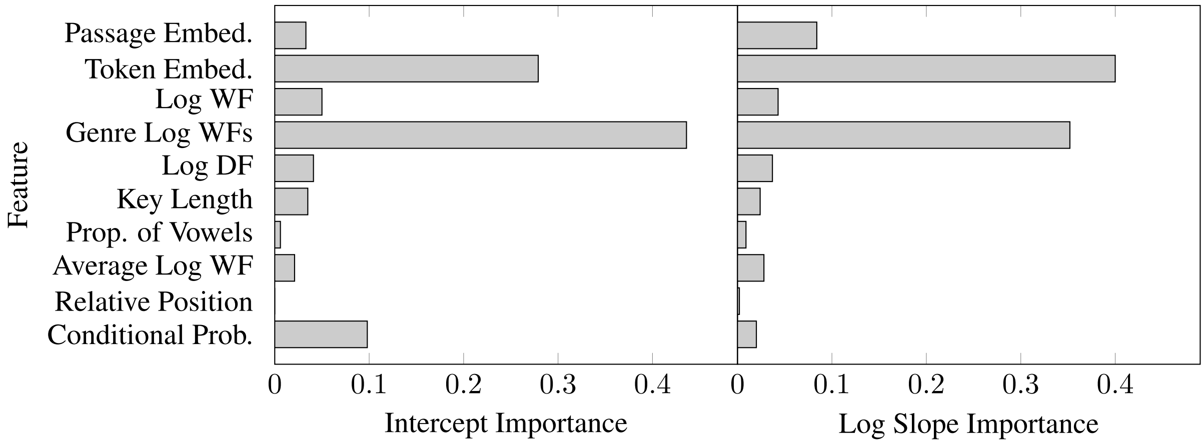


Figure 3: Feature Importances

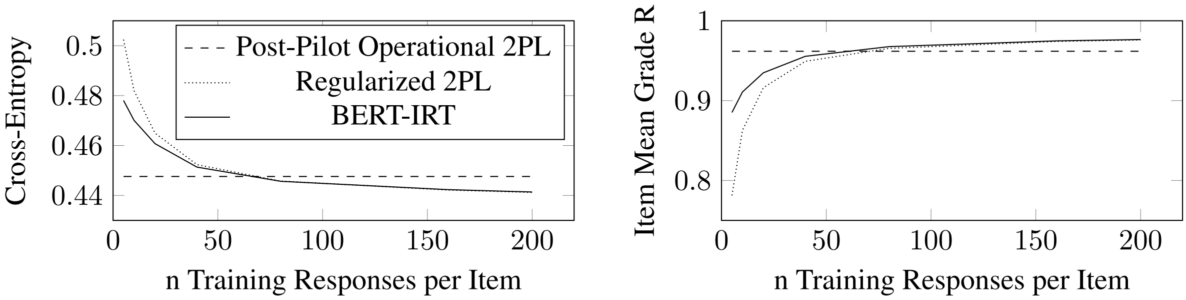


Figure 4: Experiment 4. Evaluation in a Jump-Start Scenario

confirmed that there was no significant impact on criterion validity or reliability.

Since adding the items to the operational item bank, we have collected substantial response data for all the new items, and are able to evaluate the quality of item parameters that would have been obtained had they been estimated with more data. To that end we conducted an ablation experiment similar to Experiment 1, but in a jump-start scenario (i.e., only the response data for the newly added items was ablated).

Figure 4 shows the results for Cross-Entropy and Item Mean Grade R for this ablation study. Similar to the results in Experiment 1, BERT-IRT out-performed the operational 2PL model with only a third of the data. As expected, it also out-performed the regularized 2PL model when trained on the same responses data. Importantly, even though a full third of the c-test item bank was replaced, this ablation study indicates that the impact on criterion validity and reliability would be negligible even if as few as 5 responses per item had been collected (i.e., the maximum difference between BERT-IRT and the Post-Pilot Operational IRT was less than 0.001 for both the Internal Validity Coefficient and Test-Retest Reliability metrics, even when BERT-IRT was trained on as few as 5 responses for each of the new items). This finding stands to dramatically boost the rate at which the item bank can be refreshed.

5 Conclusion & Future Work

In this paper, we demonstrated how an explanatory IRT model with BERT embeddings and other engineered NLP features can be used to accurately estimate item parameters for c-test items with limited piloting data. We showed that the model is able to use these features to generalize item parameter estimates across items, and that both BERT embeddings and engineered features contribute to the performance of the model. Furthermore, we showed how this was used on a high-stakes test of English to replace a third of its item pool with a tenth of the data that would normally have been required. Finally, our ablation study in Experiment 4 showed that we should be able to use BERT-IRT to reduce the pilot even further with negligible impact on criterion validity or reliability.

In a future work, we plan to explore similar applications of NLP and explanatory IRT models to other item types, and ways to reduce or eliminate the need for item piloting even further.

6 Limitations

There are three main limitations to our study:

- As mentioned in Section 3.4.5, this method could be improved if one were to incorporate a method to directly estimate the variance in item parameters that is explained by the features. However, finding a method that is tractable for a large number of features is difficult, and so we leave that to a future work.
- This study only evaluated the model on c-test tasks. Applications to other task types will need to be evaluated, and may require different features or IRT models to achieve good results.
- While Experiment 4 showed we successfully added a large number of c-test items to the bank with as few as 20 pilot responses per item, the ablation study that indicates we may be able to use even fewer pilot responses does not account for the potential impact that less accurate item parameters could have on the efficiency of the CAT algorithm. While we expect that impact would not significantly change our results, more study is needed to ensure that items could safely be added to the test with fewer than 20 responses per item.

Acknowledgements

We would like to thank all of those who helped review this paper and those who provided input on the development of BERT-IRT over the years, especially Alina von Davier, J.R. Lockwood, Burr Settles, and Jill Burstein.

References

Mary J Allen and Wendy M Yen. 2001. *Introduction to measurement theory*. Waveland Press.

Luca Benedetto, Giovanni Aradelli, Paolo Cremonesi, Andrea Cappelli, Andrea Giussani, and Roberto Turrin. 2021. On the application of transformers for estimating the difficulty of multiple-choice questions from text. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 147–157, Online. Association for Computational Linguistics.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Matthew Byrd and Shashank Srivastava. 2022. Predicting difficulty and discrimination of natural language questions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 119–130, Dublin, Ireland. Association for Computational Linguistics.

Ramsey Cardwell, Ben Naismith, Geoffrey T LaFlair, and Steven Nydick. 2022. Duolingo English Test: Technical Manual. Duolingo Research Report, Duolingo.

Michael Daller, Amanda Müller, and Yixin Wang-Taylor. 2021. The C-test as predictor of the academic success of international students. *International Journal of Bilingual Education and Bilingualism*, 24(10):1502–1511.

Mark Davies. 2008. Word frequency data from The Corpus of Contemporary American English (COCA). `https://www.wordfrequency.info`.

Paul De Boeck. 2004. *Explanatory item response models: A generalized linear and nonlinear approach*. Springer Science & Business Media.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dipak K Dey, Ming-Hui Chen, and Hong Chang. 1997. Bayesian approach for nonlinear random effects models. *Biometrics*, pages 1239–1252.

Thomas Eckes and Rüdiger Grotjahn. 2006. A closer look at the construct validity of C-tests. *Language Testing*, 23(3):290–325.

Gerhard H Fischer. 1973. The linear logistic test model as an instrument in educational research. *Acta psychologica*, 37(6):359–374.

Thomas François and Cédrick Fairon. 2012. An "ai readability" formula for french as a foreign language.

In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 466–477.

R Michael Furr. 2021. *Psychometrics: an introduction*. SAGE publications.

Ronald K Hambleton, Hariharan Swaminathan, and H Jane Rogers. 1991. *Fundamentals of item response theory*, volume 2. Sage.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3651–3657. Association for Computational Linguistics.

Neda Karimi. 2011. C-test and vocabulary knowledge. *Language Testing in Asia*, 1(4):7.

E. Khodadady. 2014. Construct validity of C-tests: A factorial approach. *Journal of Language Teaching and Research*, 5.

C. Klein-Braley. 1997. C-Tests in the context of reduced redundancy testing: an appraisal. *Language Testing*, 14(1):47–84.

Geoffrey T. LaFlair, Thomas Langenfeld, Basim Baig, André Kenji Horie, Yigal Attali, and Alina A. Davier. 2022. Digital-First Assessments: A Security Framework. *Journal of Computer Assisted Learning*, page jcal.12665.

Bai Li, Zining Zhu, Guillaume Thomas, Yang Xu, and Frank Rudzicz. 2021. How is BERT surprised? layerwise detection of linguistic anomalies. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4215–4228. Association for Computational Linguistics.

Mary J Lindstrom and Douglas M Bates. 1990. Nonlinear mixed effects models for repeated measures data. *Biometrics*, pages 673–687.

Frederic M Lord. 2012. *Applications of item response theory to practical testing problems*. Routledge.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

Arya D. McCarthy, Kevin P. Yancey, Geoffrey T. LaFlair, Jesse Egbert, Manqian Liao, and Burr Settles. 2021. Jump-starting item parameters for adaptive language tests. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 883–899, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

John Norris. 2018. *Developing C-tests for estimating proficiency in foreign language research*. Peter Lang, Berlin, Germany.

José C Pinheiro and Douglas M Bates. 1995. Approximations to the log-likelihood function in the nonlinear mixed-effects model. *Journal of computational and Graphical Statistics*, 4(1):12–35.

Martin F Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.

Diego Reyes, Abelino Jimenez, Pablo Dartnell, Séverin Lions, and Sebastián Ríos. 2023. Multiple-choice questions difficulty prediction with neural networks. In *International Conference in Methodologies and intelligent Systems for Techhnology Enhanced Learning*, pages 11–22. Springer.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Wim J Van der Linden and Cees AW Glas. 2010. *Elements of adaptive testing*, volume 10. Springer.

Howard Wainer, Eric T Bradlow, and Xiaohui Wang. 2007. *Testlet response theory and its applications*. Cambridge University Press.

Walter D Way. 1998. Protecting the integrity of computerized testing item pools. *Educational Measurement: Issues and Practice*, 17(4):17–27.

David J Weiss. 1982. Improving measurement quality and efficiency with adaptive testing. *Applied psychological measurement*, 6(4):473–492.

Russ Wolfinger. 1993. Laplace's approximation for nonlinear mixed models. *Biometrika*, 80(4):791–795.

Kevin Yancey, Alice Pintard, and Thomas Francois. 2021. Investigating readability of french as a foreign language with deep learning and cognitive and pedagogical features. *Lingue e linguaggio*, 20(2):229–258.

Ya Zhou and Can Tao. 2020. Multi-task bert for problem difficulty prediction. In *2020 International Conference on Communications, Information System and Computer Engineering (CISCE)*, pages 213–216.