Psychological Test and Assessment Modeling, Volume 65, 2023 (1), 55-75

Automated Distractor Generation for Fill-in-the-Blank Items Using a Prompt-Based Learning Approach

Jiyun Zu, Ikkyu Choi, Jiangang Hao

Educational Testing Service, 660 Rosedale Road, Princeton, NJ, 08540

Abstract

There are heavy demands for large and continuous supplies of new items in language testing. Automated item generation (AIG), in which computerized algorithms are used to create test items, can potentially increase the efficiency of new item development to serve this demand. A challenge for multiple-choice items is to write effective distractors, that is, incorrect yet attractive (Haladyna, 2004). We propose a prompt-based learning approach (Liu et al., 2021) for automatically generating distractors for one of the most common language-assessment item types, fill-in-the-blank vocabulary items. The proposed method treats distractor generation as a natural language generation task and utilizes a transformer-based, pretrained language model (Radford et al., 2019) fine-tuned to ensure appropriate and useful output. The fine-tuning process adopted a prompt-based learning approach, which has been found to be particularly effective in small-sample scenarios (Gao et al., 2021). We illustrate this approach on a specific item type from a standardized English language proficiency assessment. Specifically, we study the effects of different prompts and demonstrate the effectiveness of the proposed prompt-based learning approach by comparing features of generated distractors with those from a rule-based approach.

Keywords: Automated distractor generation, automated item generation, natural language processing, deep learning language models, prompt-based learning

Author Note

Correspondence concerning this article should be addressed to Jiyun Zu, Educational Testing Service, 660 Rosedale Road, Princeton, NJ, 08540. Email: jzu@ets.org

Language testing programs, like many other educational and psychological testing programs, face increasing demands for flexible test administrations. Since the COVID-19 pandemic, many language proficiency tests are offered to be taken at home with more available testing dates. Example tests include high-stake tests such as the Duolingo English Test, Pearson PTE Academic, and the TOEFL iBT® test. This move towards flexible administrations has led to heavy demands for large and continuous supplies of new items.

Automated item generation (AIG), in which computerized algorithms are used to create test items, can potentially increase the efficiency of new item development. For cognitive tests, a three-step template-based approach (Gierl & Lai, 2015) has been successfully used to automatically generate a variety of item types including mathematics achievement items (Embretson & Kingston, 2018), medical knowledge items (Gierl et al., 2012), and fluid reasoning items (Kyllonen et al., 2019). This approach involves developing item models that consist of components, creating content variants for each component, and then using computer algorithms to select and combine components into new items. The number of items created from one item model can be large because it is the product of the number of variants for each component. More recently, deep learning language models have been utilized for AIG for personality assessments, where the template-based approach is not suitable. For example, von Davier (2018) trained a long-short-term memory- (LSTM-) based recurrent neural network model and Hommel et al. (2022) fine-tuned a transformer-based deep learning language model to automatically generate personality statements (e.g., "I work hard").

Like personality assessments, AIG for language assessments will be influenced by advancement in natural language processing (NLP) technology. Over the last a few years, there have been several breakthroughs in the field of NLP. A prominent example is the use of large language models consisting of multiple hidden layers and utilizing a transformer architecture (Vaswani et al., 2017). Such transformer-based models achieved state-of-the-art performance on a wide range of NLP benchmark tasks, such as the General Language Understanding Evaluation (GLUE; Wang et al., 2019), the Standard Question Answering Dataset (SQuAD; Rajpurkar et al., 2016), and the Situations with Adversarial Generations (SWAG; Zellers et al., 2018). However, the benchmark tasks tend to have a very large number of examples that not all practical problems can provide. Prompt-based learning is an approach to address this challenge; it is designed to use large language models in an efficient manner in a small sample context. One demonstration of the power of language prompting is Generative Pre-trained Transformer 3 (GPT-3; Brown et al., 2020). Leveraging natural language prompts and its 175-billion parameters, with only a few examples that demonstrate the specific task, GPT-3 has achieved good performance on many different NLP tasks, such as translation and question-answering.

In this paper, inspired by the effects of language prompts, we propose a prompt-based learning approach to address the challenge of small number of training examples in AIG. Specifically, the proposed method involves fine-tuning a large, pre-trained

language model to generate distractors for fill-in-the-blank multiple-choice vocabulary items. A typical fine-tuning process consumes a large number of examples (oftentimes several tens of thousands), yet it is rare for a testing program to have such a large item pool. Thus, we designed language prompts for distractors and leveraged the prompts in fine-tuning to address this small sample challenge.

The remainder of this paper is organized as follows. We first introduce fill-in-the-blank vocabulary items and existing methods for automated distractor generation. We then describe the proposed prompt-based learning approach in detail and illustrate the proposed approach using data from a standardized English language proficiency test. The paper ends with discussion about the implications and limitations.

Distractor Generation for Fill-in-the-Blank Vocabulary Items

A popular item type in the language assessment and learning domain is fill-in-the-blank (cloze) multiple-choice vocabulary items. Different variants of this item type appear in language assessments (e.g., IELTS, TOEFL iBT®, and TOEIC®) and language learning applications (e.g., Kids A-Z). Below is an example item, where * indicates the correct answer. In each item, a sentence with a word or phrase missing and a few options (usually three or four) are presented, and test takers are asked to select the best answer to complete the sentence. This item type assesses test takers' ability to choose an appropriate word based on their understanding of the context and their vocabulary.

All clothing sold in Develyn's Boutique is made from natural materials and contains no _____ dyes.

(A) immediate

(B) synthetic*

(C) reasonable

(D) assumed

Throughout this paper, we refer to this item as the example item. We also use the following terminologies to refer to various parts of this item type: a stem is a "blanked" sentence that is shown to test takers, a key is the correct option, distractors are incorrect options, and a carrier sentence is a full sentence with its corresponding blank filled in with the key.

One major challenge in developing any multiple-choice item is to write effective distractors. An effective distractor needs to be incorrect yet attractive. For example, Haladyna and Downing (1988) suggested that an effective distractor should be able to attract greater than five percent of test takers. It is however often difficult to write such distractors. A distractor that is too attractive may be regarded as a legitimate key by some, and a distractor that is clearly incorrect may not be attractive at all. Finding the

right balance between being attractive and incorrect thus requires experience and expertise from item content developers, those who create and review content of items. As noted by Haladyna (2004, p. 120), "Writing plausible distractors comes from hard work and is the most difficult part of multiple-choice item writing". The heavy reliance on content developer expertise, combined with the need for multiple distractors per item, often leads to an expensive and time-consuming item development process. On the other hand, if effective distractors can be generated automatically, the current development process for multiple-choice items can become much more efficient.

For fill-in-the-blank vocabulary items, one way of characterizing the attractiveness of a distractor is to leverage its relationship with the context provided by the stem and with the key. Pervious research has used these relationships for automated distractor generation. For example, Hill & Simha (2016) defined good distractors as those fit within a narrow context (represented by the key) but not within the broader context of surrounding words (referred to as context words). They thus proposed to use words that satisfy the following two conditions as distractors: (1) belonging to the same part-of-speech (POS) category as the corresponding key in the Google *n*-gram corpus and (2) having smaller co-occurring likelihood with context words than the key does. Susanti et al. (2018) used semantic similarity based on word embeddings as well as collocation information to rank distractor candidates in terms of their attractiveness. The resulting distractors were thus words that frequently appear surrounding the two adjacent words around the blank but were not semantically similar to the key. Another approach toward distractor generation utilized errors. Sakaguchi et al. (2013) used errors made by English as a second language learners as sources. More specifically, they extracted pairs of errors and correct forms from the Lang-8 Corpus of Learner English, made the correct forms into the key, and turned errors with a high confusion probability by the learners into distractors. Panda et al. (2022), on the other hand, leveraged errors from neural machine translation. Specifically, they processed an English sentence through a round-trip machine translation pipeline to multiple different languages and back to English. After many such round-trips, they took back-translated words aligned with the key as distractor candidates and then used semantic similarity between distractor candidates and the key as well as fill-mask scores from BERT (Devlin et al., 2019) to rank candidates.

In summary, these approaches involved AIG researchers constructing a list of words as distractor candidates, choosing measures for the relationship between distractors with context and the key, and making rules to rank-order distractor candidates based on the chosen measures. The generated distractors were the highest-ranking candidates. However, different researchers used different operational definitions and developed algorithms to generate distractors that best fit the corresponding definition. The reliance upon researchers' own definitions may be unavoidable when there are few items that exemplify desirable relationships among stems, keys, and distractors. However, if example items are available (e.g., from a historical item bank), an alternative approach is to learn those relationships directly from the examples.

In this paper, we propose a prompt-based learning approach via learning from existing example items as an alternative to rule-base approaches. The main motivation is to learn the rules for distractors from example items in which these rules had been successfully applied by expert item content developers, thus avoiding the need for devising explicit rules ourselves. The proposed approach considers distractor generation as a natural language generation task and utilizes a transformer-based, pretrained language model (Radford et al., 2019) fine-tuned with existing example items as training samples. To address a practical challenge of a small number of existing example items, we use a prompt-based learning approach that has been found to be particularly effective in small-sample scenarios (Gao et al., 2021). Details of the proposed approach are provided in the next section.

Prompt-based Learning Approach

In a recent survey paper, Liu et al. (2021) called prompt-based learning the newest paradigm in NLP and attributed the rising interest to the popularity of GPT-3 (Brown et al., 2020). A "prompt" in this context consists of an input text, a natural language description of the target task, and a slot for the output text to be generated. For example, Radford et al. (2019) created a prompt for a text summarization task on the Cable News Network (CNN) and Daily Mail datasets by adding "TL; DR:" (an abbreviation for "too long: didn't read") after a CNN or Daily Mail article (i.e., input text). Denote the input text as [X] and the output text to be generated as [Z], this prompt can then be written as "[X] TL; DR: [Z]". Similarly, for a machine translation task, say, from French to English, a prompt such as "French: [X] English: [Z]" can be used. To paraphrase an input sentence, Schick and Schütze (2021) used the following prompt "Write two sentences that mean the same thing. [X][Z]".

Prompt-based learning allows fine-tuning pre-trained language models with a task specific prompt and training data. Like the standard fine-tuning approach, prompt-based learning leverages general language representation from a pre-trained model and updates parameter estimates based on new training data from the specific downstream task. Prompt-based learning is particularly effective when there is not enough training data, because an effective prompt can help update parameter estimates in the right direction (Liu et al., 2021). Gao et al. (2021) demonstrated that when the number of training samples was small, the prompt-based learning approach outperformed standard fine-tuning procedures by 11% on average on a range of NLP tasks. Scao and Rush (2021) also showed that a prompt may be worth 100 conventional training samples.

Recall that our goal is to generate distractors for fill-in-the-blank vocabulary items by learning from a limited number of example items. We treat this as a text generation task and fine-tune a large language model so that it can generate distractors by learning the relationships between distractors and stems and keys from the example items.

Specifically, a prompt-based learning approach is used to alleviate the small training sample problem.

There are two main design considerations for implementing prompt-based learning: the choice of a pre-trained model and the development of an effective prompt. For the pre-trained model, we chose GPT-2 (Radford et al., 2019). At the time of this study, GPT-2 was the largest open-source model from the GPT series and showed good performance on natural language generation (Radford et al., 2019). The goal of prompt development is to devise a prompting function resulting in the most effective performance on the specific downstream task at hand (distractor generation in this context). We developed multiple prompts by reformatting the carrier sentence, key, and each of the distractors.

Specifically, we examined five different prompts, as can be seen in Table 1. In each prompt, the carrier sentence was presented first, and a distractor appeared at the end of the prompt. The carrier sentence contained information on the context and the key and was thus provided as the input text. The distractor was the target output. We varied the ways we described the nature of distractors across the five prompts. These prompts, each using the sample item with one distractor as examples, are presented in Table 1, in the ascending order of specificity. The first prompt used only "::" to prompt the distractor. The second prompt used the words "Key" and "Distractor," which are common terms in assessments. The third prompt prompted a distractor with a more explicit phrase "not that similar to". The fourth prompt also described a distractor as "not that similar to", but also more specifically pointed out "the word" and "in the previous sentence". Lastly, the fifth prompt was similar to the fourth one but emphasized that a distractor should not fit in the context of the carrier sentence by prompting a distractor with "should not be replaced by". We hypothesized that the more specific prompts would yield better performance.

Table 1

Five Proposed Prompts for Distractor Generation for Fill-in-the-Blank Vocabulary Items

Number	Format	Content
1	Prompt	[X] [K]::[Z]
	Example	All clothing sold in Develyn's Boutique is made from natural materials and contains no synthetic dyes. synthetic::immediate
2	Prompt	[X] Key: [K]. Distractor: [Z]
	Example	All clothing sold in Develyn's Boutique is made from natural materials and contains no synthetic dyes. Key: synthetic. Distractor: immediate
3	Prompt	[X] [Key] is not that similar to [Z].
	Example	All clothing sold in Develyn's Boutique is made from natural materials and contains no synthetic dyes. Synthetic is not that similar to immediate.
4	Prompt	[X] The word ["K"] in the previous sentence is not that similar to [Z].

	Example	All clothing sold in Develyn's Boutique is made from natural materials and contains no synthetic dyes. The word "synthetic" in the previous sentence is not that similar to immediate.
5	Prompt	[X] The word ["K"] in the previous sentence should not be replaced by [Z].
	Example	All clothing sold in Develyn's Boutique is made from natural materials and contains no synthetic dyes. The word "synthetic" in the previous sentence should not be replaced by immediate.

Methods

In this section, we present details of implementing the proposed prompt-learning approach to generate distractors for a fill-in-the-blank vocabulary item type from a standardized English language proficiency assessment. We also describe how we evaluated the results from the proposed approach.

Datasets

Our main data were 4,572 fill-in-the-blank vocabulary items from a large-scale high-stake standardized English-language proficiency test. These items were developed across several years. Each item contained a stem, a key, and three distractors. Each item was developed and reviewed by the content development team of this test before being administered to test takers in test forms. Classical item analysis was conducted to each form. These items were determined as suitable as operational items based on classical item analysis statistics and item content. We thus considered these items as representing desirable relationships among stems, keys, and distractors. The entirety of the 4,572 items will be referred to as the total set in the remainder of this paper. We randomly split the total set into a training set consisting of 3,429 items (75%) and a validation set consisting of the remaining 1,143 items (25%). The stem of each item was combined with all three distractors to form three samples per item. Consequently, there were 10,287 samples in the training set and 3,429 samples in the validation set. Each sample was then transformed into all five prompts shown in Table 1.

Prompt-based Fine-tuning

We fine-tuned the largest GPT-2 model (i.e., 48 layers, 1,600 model dimensions, 1.5 billion parameters and a vocabulary size of 50,257) with the training set adapted with each of the five prompts as shown in Table 1. Each of the five fine-tuning was done

with three epochs with a learning rate of 5e-5 and a batch size of 16[1]. We used the cross-entropy loss and perplexity (i.e., exponential of the cross-entropy loss) in the validation set as metrics to evaluate and compare the performance of the fine-tuned language models. A lower perplexity value indicates that the model was less "confused" with the next word; thus a model with a lower perplexity value can be regarded as a better fitting model. Because different models differed only in terms of the prompts, we considered the prompt in the model with lower loss/perplexity as being better at generating the distractors. All fine-tuning was conducted in Python using the Transformers library (Wolf et al., 2020) via the PyTorch framework (Paszke et al., 2019).

Distractor Generation

We generated distractors for 802 items, which are items in the validation set whose keys are adjectives (ADJ), adverbs (ADV), nouns (NOUN), or verbs (VERB). We refer to these items as the generation set. For each item in the generation set, we generated five distractors using two approaches: the proposed prompt-based learning approach and a rule-based approach. For the prompt-based learning approach, only the best prompt based on the fine-tuning results was used. Considering the previous work utilizing rule-based approaches in the literature, we included the rule-based approach for comparison purposes.

Prompt-based learning approach

To generate distractors, for each item, we provided the prompt and used *top-p* and *top-k* sampling (Holtzman et al., 2020) with $p = .97$ and $k = 30$ to generate 10 sets of the next words. This sampling mechanism picks from the minimum number of tokens[2] that together exceed 97% of the probability mass, thus avoiding unlikely tokens and reducing repeats. We then filtered the sampled words using the process described later in this section ("Common filtering process") and took the first five remaining words after the filtering process as the output from the prompt-based learning approach.

[1] These hyperparameters were chosen based on our previous research on a similar topic, in which we compared 4 sets of settings for learning rate and batch size and found little meaningful differences in the end results.

[2] A token can be a word, part of a word, or just characters like punctuation.

Rule-based embedding approach

For comparison purposes, we also generated distractors using a rule-based approach. Specifically, we implemented Susanti et al.'s (2018) approach relying on word-embeddings while incorporating information from existing examples. The specific settings reflected our best attempt at understanding and implementing rules applied by content experts after examination of the total set. To come up with distractor candidate lists, we selected all adjectives, adverbs, nouns, and verbs in the GoogleBooks unigram corpus whose standardized frequency indices (SFIs) were within the SFI range of all options in the total set. The SFI of word w is defined as $\mathrm{SFI_w} = 10(\log_{10} U_w + 4)$, where U_w is that word's estimated frequency per 1 million tokens (Carroll, 1971). For example, an SFI of 80 means that a word can be expected to occur about once in every 100 tokens, and an SFI of 70 means once in every 1,000 tokens. We then created four lists, one for each POS category, which contained lemmas of the resulting words. For each item, the item-specific distractor candidate list was the distractor candidate list of the same POS as the key removing those whose SFIs were more than 14 away from the SFI of the lemma of the key and those described in the "Common filtering process" section below.

To generate distractors for a given item, we calculated the cosine similarity values of word-embeddings among the lemma of the key and all lemmas in the item-specific distractor candidate list. We used the pre-trained word embeddings in the en_core_web_lg model provided by the spaCy library (Honnibal, et. al, 2020), whose vocabulary consists of 514k unique word embedding vectors of 300 dimensions. Cosine similarity is the cosine of the angle between two vectors. It is within the range of -1 to 1. Because the word-embeddings were trained to put words that are similar semantically closer in the space, cosine similarity of two vectors represents semantic similarity of the two words, where a higher value indicate higher semantic similarity. We then sampled five distractors from the list that matches the POS of the key leveraging the cosine similarity values. Specifically, one lemma was sampled from the third quartile and four lemmas were sampled from the fourth quartile of cosine similarity with the key. The same filtering rule (described in the next subsection) as the prompt-based approach was applied to the sampled lemmas. Lastly, for verbs and nouns, the sampled lemmas were changed to the same form as the key.

Common filtering process

We applied the same filtering process to both the prompt-based learning approach and the rule-based approach. The filtering process involved removing distractors that satisfied one or more of the following four conditions: (a) is the same as the key, (b) is a synonym of the key according to the WordNet synonym list (Fellbaum, 1998), (c) is among the 10 highest scored tokens from the fill-mask task by the language model RoBERTa-large (Liu et al., 2019), or (d) is on the do-not-use word list for this test.

More specifically, to obtain (c), we used the fill-mask pipeline in the Transformers library by providing the stem with the blank replaced by a masked token. Outputs were tokens with scores, where a token with a higher score was more likely to fill in the blank given the stem and the RoBERTa-large language model. Since the distribution of scores across the tokens in the vocabulary differs by the stem, we used the rank (top 10) instead of the absolute value of the scores as a criterion for filtering. The goal of the first three conditions was to reduce the chance of generating distractors that can be considered as another key. The last condition was to ensure that the generated distractors are relevant to the target testing program.

Measures for Evaluating Generated Distractors

Our goal with the proposed prompt-based learning approach was to generate distractors that retain the characteristics of distractors that expert item developers wrote. Relevant characteristics include vocabulary of the distractors, relationship between distractors and the key, and relationship between distractors and the carrier sentence. We used SFIs of generated distractors, semantic similarities between distractors and the keys, and RoBERTa fill-mask ranks of distractors as measures of these characteristics and used these measures to evaluate the generated distractors. Definition of these measures are the same as described in the above "Rule-based embedding approach" and "Common filtering process" sections. For ranks from the fill-mask task, we ignored tokens after the first 15,000.

Results

Effects of Prompts

The loss and perplexity values of the five models fine-tuned with different prompts on the validation set are summarized in Table 2. Given the one-to-one, monotonic relationship between the loss and perplexity values (the latter is the exponential of the former), we focus on the loss value in this section. As expected, the language model fine-tuned with Prompt 1, which didn't describe the relationship between the key and a distractor with text at all, resulted in the largest loss value. Prompt 2, which used assessment terms "key" and "distractor," led to a smaller loss value than that of the model based on Prompt 1. Prompt 3, which queued the distractor with "not that similar to" but did not specifically point out the connection with the input carrier sentence, performed comparably to Prompt 2. Models fine-tuned with Prompt 4 and 5, which specifically pointed out the relationship between the key, distractor, and carrier sentence yielded the smallest loss values. The loss values from Prompts 4 and 5 were comparable, even though the two prompts emphasized different aspects of the relationship between the key and a distractor. Given the similar performance, we focus on the results from the model fine-tuned with Prompt 5 in the remainder of this paper.

Table 2

Loss and Perplexity of Fine-tuned Models with Different Prompts

Prompt	Perplexity	Loss
1. [X] [K]::[Z]	43.13	3.76
2. [X] Key: [K]. Distractor: [Z]	23.29	3.15
3. [X] [Key] is not that similar to [Z].	23.55	3.16
4. [X] The word ["K"] in the previous sentence is not that similar to [Z].	12.73	2.54
5. [X] The word ["K"] in the previous sentence should not be replaced by [Z].	12.71	2.54

Evaluating Generated Distractors

For each item in the generation set, we generated five distractors using the prompt-based learning approach with Prompt 5 and another five distractors using the rule-based embedding approach. We then obtained the evaluation measures used for this study including SFIs, cosine similarities, and fill-mask ranks for the original distractors developed by content developers (three distractors per item) and for the automatically generated distractors from the two approaches (five distractors per item per approach). Descriptive statistics and boxplots of these feature values are respectively provided in Table 3 and Figure 1.

Table 3

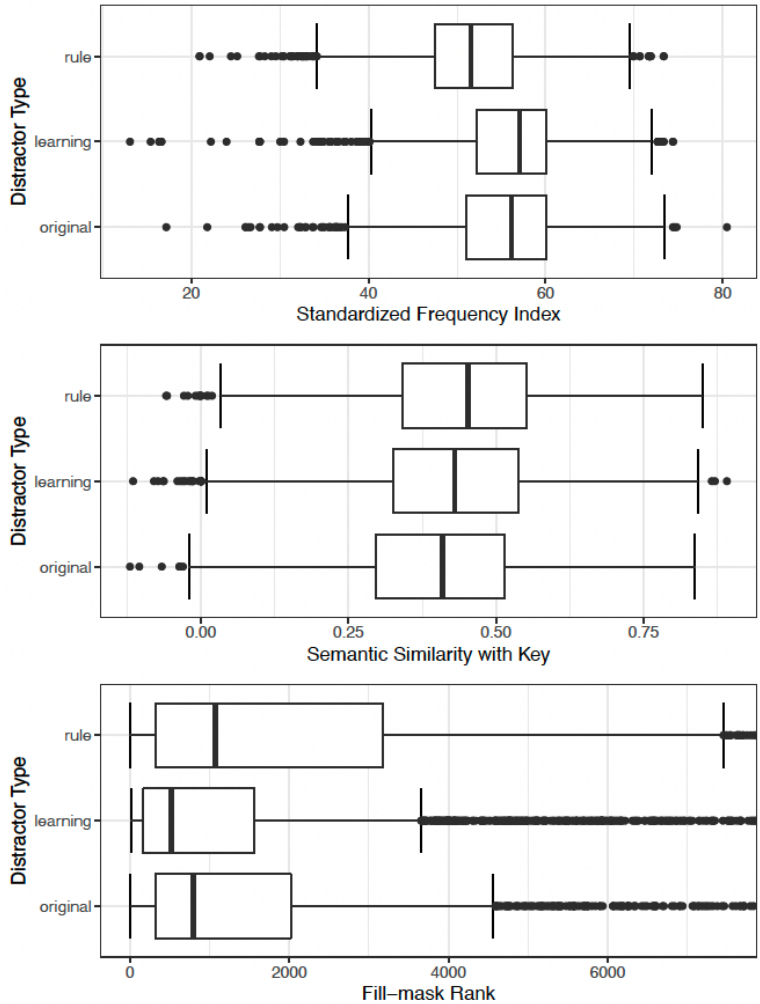
Descriptive Statistics of Features for the Generated Distractors

Measure	Distractor Type	N	2.5th Percentile	25th Percentile	50th Percentile	75th Percentile	97.5th Percentile
SFI	original	2394	39.32	51.04	56.14	60.01	67.25
	learning	3981	43.00	52.20	57.03	60.13	67.58
	rule	3930	38.43	47.46	51.57	56.29	64.67
Similarity with Key	original	2392	0.09	0.30	0.41	0.51	0.70
	learning	3968	0.12	0.33	0.43	0.54	0.72
	rule	3930	0.14	0.34	0.45	0.55	0.72
Fill-mask Rank	original	2164	46.00	318.00	795.50	2023.00	8974.47
	learning	3735	20.00	167.00	520.00	1561.00	8368.30
	rule	3107	32.00	316.50	1073.00	3172.00	12123.65

Note. learning = the prompt-based learning approach. rule = rule-based embedding approach. original = original distractors generated by content developer

Figure 1

Boxplots of Features of Different Types of Distractors



Note. This figure demonstrates the distributions of features, including standardized frequency index, semantic similarity with key, and fill-mask rank of distractors generated by different methods in the generation set. rule = rule-based embedding approach; learning = prompt-based learning approach; original = original distractors generated by content developers.

The top panel in Figure 1 shows that the distribution of SFIs of distractors generated by the prompt-based learning approach largely overlaps with that of the original distractors but slightly leans towards higher frequency words. That is, the prompt-based learning approach yielded fewer rare words than the original distractors written by item developers. The two distributions were particularly close to each other in the middle: their 25th, median, and 75th percentiles differed only by 1.16, 0.89, and 0.12, respectively on the SFI scale. This close overlap was noteworthy because word frequency was not explicitly used as a criterion for the prompt-based learning approach; instead, the language model was able to learn frequency information as part of the overall characteristics of original distractors during the prompt-based fine-tuning process and reproduce that learned information in the generation phase. In comparison, distractors generated by the rule-based method tended to have lower SFIs (i.e., more rare words) than the original distractors. The overall distribution of the rule-based distractor SFIs deviated more from the original distractor SFI distribution than that of the prompt-based learning distractor SFIs did. For example, median SFIs of distractors generated by the rule-based embedding method differed from that of the original distractors by 4.57.

The middle panel of Figure 1 shows that the distribution of semantic similarities with the key (measured by cosine similarity) for distractors generated by the prompt-based learning approach also largely overlapped with that of the original distractors. Even though the distribution from the prompt-based distractors was slightly tilting towards the higher end of the scale, the differences between the two distributions at the 25th, median, and 50th percentiles were all smaller than 0.03 on the cosine similarity scale. The rule-based approach yielded distractors whose semantic similarities with the corresponding keys were even higher than those from the prompt-based learning approach. As a result, the semantic similarity distribution from the rule-based approach deviated further from that of the original distractors.

The bottom panel of Figure 1 shows the distributions of fill-mask ranks of distractors. Although we obtained rank information up to the 15,000th token from the fill-mask pipeline, we only plotted the first 8,000 to facilitate the presentation of the patterns in the high-ranked distractors (the highest rank is 1). The fill-mask rank distribution for the original distractors confirmed that distractors written by expert item developers can be low-ranked words according to the pretrained RoBERTA model. For example, the 75th percentile of the original distractors' fill-mask rank distribution was the 2,023th in the order of most likely words to fill in the blank. This makes it difficult to use fill-mask probabilities or ranks as a measure to rank distractor candidates. We see that, compared to the original distractors' fill-mask rank distribution, the distribution of fill-mask ranks from distractors generated by the prompt-based learning approach was grouped more closely towards the more likely words. That is, the prompt-based learning approach yielded distractors that were overall more likely to fill in the blank than the original distractors (according to the pretrained RoBERTA model). On the other hand, the rule-based distractors were overall less likely to fill in the blank than the original distractors.

The prompt-based distractors were thus likely to be closer to the context provided by the stem than the original distractors and the rule-based distractors. Having distractors that are more relevant to the context (which in turn would make the distractor more attractive) may be desirable, especially if more difficult items are needed. However, there could also be disadvantages. For example, if a distractor is too close to the context, it could be considered as a key. We note however that we did not find any evidence of more key-able distractors in our review (described in the next section) of the generated distractors.

Examples of Generated Distractors

To get a more concrete sense of distractors generated by the prompt-based learning approach and to evaluate them relative to the original distractors and the rule-based distractors, we present a small number of examples in this section. We begin with the example item shown earlier in this paper. Specifically, Table 4 gives the distractors generated by the prompt-based learning and rule-based approaches for that item as well as other relevant information including words with the top 10 fill-mask scores, the key, and the original distractors. The original distractors had mid-to-low similarity values with the key and were quite low on the fill-mask ranking. This again suggests the difficulty of setting rules based on features such as similarity with the key or fill-mask scores, because words with low similarity values and low fill-mask ranking can still be used as distractors by expert item developers.

Table 4

Features for Top-ranked Words, Key, and Distractors for an Example Item

Word	Type	Fill-mask Rank	Fill-mask Score	SFI	Similarity with Key
artificial		1	0.41	54.81	0.71
synthetic	key	2	0.21	51.19	1.00
harmful		3	0.15	50.09	0.43
chemical		4	0.06	58.69	0.68
toxic		5	0.03	51.45	0.49
added		6	0.03	61.42	0.28
unnatural		7	0.02	48.20	0.55
harsh		8	0.02	51.34	0.21
special		9	0.01	63.59	0.41
visible		10	0.00	56.73	0.42

excessive	learning	16	<0.01	54.61	0.49
imported	learning	880	<0.01	52.72	0.44
relative	learning	2174	<0.01	60.06	0.45
uncertain	rule	2752	<0.01	53.42	0.40
consistent	rule	2794	<0.01	56.64	0.54
assumed	original	3344	<0.01	58.38	0.20
immediate	original	3953	<0.01	58.52	0.31
reasonable	original	5383	<0.01	57.76	0.42
digestive	rule	7255	<0.01	47.89	0.46
coherent	learning	12604	<0.01	50.04	0.43
advisable	rule	NA	NA	49.50	0.45
exogenous	rule	NA	NA	45.85	0.59
instilled	learning	NA	NA	41.51	0.35

Note. This table contains *for the item "*All clothing sold in Develyn's Boutique is made from natural materials and contains no _____ dyes." learning = the prompt-based learning approach. rule = rule-based embedding approach. original = original distractors generated by content developers

Words with the 10 highest fill-mask scores include multiple words that are not synonymous with the key but can still properly fill in the blank. For example, if "harmful", "chemical", or "toxic" were presented as options, the designated key "synthetic" may not be the single best answer for this item. However, their SFIs and similarities are indistinguishable from those of the original distractors in that they all have SFIs comparable with the key and mid-range similarity values with the key. This exemplifies the challenge of avoiding "keyable" distractors based only on SFIs and similarity values, as adopted by some of the previous rule-based approaches, and motivated one of our filters to remove the ten highest-ranking words from distractor candidates. We cannot ensure distractors generated after removing the top ten highest-ranking words are all not "keyable", because for certain stems more than ten words can fit in the context. However, removing them reduced the probability of generating keyable distractors.

The five distractors generated by the prompt-based learning approach were "excessive", "imported", "relative", "coherent", and "instilled". They are all adjectives and comparable to the key and to each other in terms of word length or frequency. This homogeneity should help prevent test takers from successfully guessing the key purely based on the appearance. In this particular example, three of the five distractors generated by the prompt-based learning approach had the highest fill-mask ranks among

all original and generated distractors, suggesting that these three distractors fit the context closer according to the pre-trained RoBERTa model.

We reviewed the generated distractors for all items in the generation set using the same criteria as the above. Four examples, each representing a key that belongs to one of the four POS categories (i.e., ADJ, ADV, NOUN, and VERB) are provided in Table 5. As the overall patterns were similar across the examples, we refrain from example-specific comments and present an overall summary interpretation. In general, we observed that the prompt-based learning approach generated distractors that had the same POS category and in the same grammatical form as the key. We also noticed that the prompt-based learning approach generated distractors that fit closer to the context comparing to those generated by the rule-based approach, as previously shown in the fill-mask rank distribution in the bottom panel of Figure 1.

Discussion and Conclusions

In this paper, we proposed a prompt-based learning approach to generate distractors for fill-in-the-blank vocabulary items, which are widely used in language assessments. We also illustrated this approach using data from a large-scale standardized English language proficiency test. We studied five prompts and found that, as expected, prompts describing the nature of distractors in natural language with a specific reference to the carrier sentence yielded the best performance. We also reviewed generated distractors by the prompt-based learning approach and compared them to those generated by a rule-based embedding approach in terms of SFI, semantic similarity, and fill-mask rank. Results suggested that, compared to the distractors from the rule-based approach, the distractors generated by the prompt-based learning approach were closer in terms of SFI and semantic similarity values to the original distractors written by expert item developers. The prompt-based distractors also tended to fit the context closer than the original distractors (measured by the fill-mask rank), while the rule-based distractors were less relevant to the context.

Table 5

Example Items with Different Types of Distractors

Item	Stem	Key	POS	Type	Distractor 1	Distractor 2	Distractor 3	Distractor 4	Distractor 5
1	Tickets for the Anniversary Concert Series will be _____ at the Brewster Hall ticket office from January 9 until January 22.	available	ADJ	original	vacant	revised	finished		
				learning	important	easy	preferable	frequent	useful
				rule	confident	local	prospective	comparable	particular
2	Based on its _____ performance in laboratory tests, the new Conlime cleaning solution was approved for commercial use.	outstanding	ADJ	original	willing	contentious	applicable		
				learning	punctual	favorable	perpetual	credential	absolute
				rule	contagious	experimental	fleeting	consecutive	rotational
3	The committee members were glad to see how _____ Ms. Park presented the benefits of the incentive program.	skillfully	ADV	original	privately	apparently	likely		
				learning	potentially	commonly	variously	highly	finally
				rule	morally	apologetically	sympathetically	expertly	relevantly
4	The Wiltshire Orchestra?is concert was _____ three hours long, ending just after 11 P.M.	approximately	ADV	original	attentively	endlessly	comparatively		
				learning	closely	entirely	energetically	eventually	extremely
				rule	highly	daily	substantially	continuously	jointly
5	Byong-Yoon Sun will retire at the end of the month, and Hye-Kyong Kwon will assume the _____ of company vice president.	role	NOUN	original	portion	use	example		
				learning	method	report	view	name	direction
				rule	branch	admiration	approach	commitment	consequent
6	The success of Friendly Frog toys is primarily due to the clever marketing _____ employed by the manufacturers.	strategy	NOUN	original	layout	status	print		
				learning	order	object	experience	issue	project
				rule	muscle	fallacy	stock	shift	evil
7	Complaints about the telephone service must be _____ to the department manager.	directed	VERB	original	answered	questioned	informed		
				learning	invited	associated	required	participated	connected
				rule	qualify	understood	bothered	ascertained	conformed
8	Fujimori Builders will put up signs redirecting traffic in order to _____ for road construction in the area.	prepare	VERB	original	restore	predict	initiate		
				learning	repair	accept	suggest	compress	display
				rule	install	identify	succeed	carry	acknowledge

One main motivation for the proposed approach was to avoid the difficulty and arbitrariness in devising a set of rules for generating effective distractors in an automated algorithm. In our review of the literature, such attempts resulted in a range of definitions and targets, making it difficult to evaluate the resulting distractors against coherent criteria. Instead, our goal was to learn subtle rules for distractor generation based on example items where such rules had already been applied by human experts. In general, our findings are encouraging in that the resulting distractors from the proposed approach closely resembled the original distractors written by expert item developers in terms of three major characteristics that were consistently included as part of previous rule-based approaches. This provides empirical evidence that the subtle (and often undocumented) rules human experts apply when they write distractors can indeed be learned and reproduced by machines.

We observed differences across the five prompts in terms of representing the original distractors for the validation set. As we hypothesized, there was a clear relationship between the specificity of a prompt and its performance: the more specific and descriptive the prompt was, the better the performance. This finding is in line with the previous findings of Schick and Schütze (2021) and Gao et al. (2021) and thus provides additional empirical evidence for the importance of devising an effective prompt for the target task at hand. For item types other than fill-in-the-blank vocabulary items, depending on the nature of the task (e.g., select the answer that is grammatically correct, select the best answer to a question, select the option that best summarizes a paragraph), prompts for distractors may need to be revised or developed. Operational testing programs often have such resources in item content developers, and we believe that active collaborations between item content developers and AIG researchers can be particularly helpful in devising effective prompts.

Our findings have practical implications for operational testing programs that have a pool of existing multiple-choice items and are interested in AIG. The proposed prompt-based learning approach provides an alternative to going through a challenging and potentially arbitrary process of devising machine-implementable rules for automated distractor generation. It also utilizes readily available, open-source models, libraries, and frameworks. These make its application to another test program easier. By using open-source models, it also helps assuage security and ownership concerns associated with test content being generated by a third-party.

Our proposed prompt-based learning approach, as well as this study itself, is limited on several fronts. The approach cannot be applied to new testing programs that don't have example items. In this study, we implemented the prompt-based learning approach in the context of a large test program with several thousands of existing items. We have yet to apply the approach to much smaller programs. It is thus not clear how generalizable our findings would be to a new test program with only hundreds (or even fewer than hundred) of existing items. Another important limitation is our evaluation of the generated distractors in this study. We relied on evaluation measures that can be computed automatically and our qualitative review of the output. We believe a crucial next step in evaluation is to gather reviews and revisions of the generated

distractors from item developers who are experienced with item development of this item type.

We believe that multiple promising future research strands exist to address the limitations of the proposed approach and the current study. Of particular interest is the smaller-sample (such as with hundreds of existing items) performance of the proposed approach. If that performance is not satisfactory, one option to consider is to utilize larger pretrained language models. There is a trade-off between the size of a pretrained model size and the training sample size needed to achieve a certain downstream task performance. In this paper, we fine-tuned the largest open-source GPT-2 model (with 1.5 billion parameters), but as shown by Brown et al. (2020), GPT-3 (with 175 billion parameters) outperformed smaller models on a variety of tasks with only a few examples. GPT-3 is not open-source, but language models are being developed rapidly, and efforts have been made to release larger open-source models (e.g., Zhang et al., 2022). Another important topic is how the proposed method is affected by the quality example items. Of course, determining whether a given performance is satisfactory is a non-trivial task. We plan to develop a more robust and meaningful evaluation process for automatically generated distractors, and involving expert item developers can be a key part of that process. We are not claiming that the proposed distractor generation method replaces the need for content expert review, especially for high-stake tests. At the current state, we consider it a useful alternative for providing a large number of draft items for human to choose from, thus improving the efficiency of item development. We would like to collect content experts' qualitative and quantitative evaluation, comments, and revisions, and use them for continuous improvement of AIG technology.

References

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020). *Language models are few-shot learners* (arXiv:2005.14165). arXiv. https://doi.org/10.48550/arXiv.2005.14165

Carroll, J. B. (1971). Measurement properties of subjective magnitude estimates of word frequency. *Journal of Verbal Learning and Verbal Behavior*, *10*(6), 722–729. https://doi.org/10.1016/S0022-5371(71)80081-6

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

Embretson, S. E., & Kingston, N. M. (2018). Automatic item generation: A more efficient process for developing mathematics achievement items? *Journal of Educational Measurement*, *55*(1), 112–131. https://doi.org/10.1111/jedm.12166

Fellbaum, C. (1998). *WordNet: An electronic lexical database.* Cambridge, MA: MIT Press.

Gao, T., Fisch, A., & Chen, D. (2021). Making pre-trained language models better few-shot learners. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 3816–3830. https://doi.org/10.18653/v1/2021.acl-long.295

Gierl, M. J., & Lai, H. (2015). Automatic item generation. In S. Lane, M. R. Raymond, & T. M. Haladyna (Eds.), *Handbook of test development* (2nd ed.). Routledge.

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

Haladyna, T. M. (2004). *Developing and validating multiple-choice test items* (3rd ed.). Routledge. https://doi.org/10.4324/9780203825945

Hill, J., & Simha, R. (2016). Automatic generation of context-based fill-in-the-blank exercises using co-occurrence likelihoods and Google n-grams. *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, 23–30. https://doi.org/10.18653/v1/W16-0503

Holtzman, A., Buys, J., Du, L., Forbes, M., & Choi, Y. (2020). *The curious case of neural text degeneration* (arXiv:1904.09751). arXiv. https://doi.org/10.48550/arXiv.1904.09751

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

Honnibal, M., & Montani, I. (2017). *spaCy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing.*

Kyllonen, P., Hartman, R., Sprenger, A., Weeks, J., Bertling, M., McGrew, K., Kriz, S., Bertling, J., Fife, J., & Stankov, L. (2019) General fluid/inductive reasoning battery for a high-ability population. *Behavior Research Methods*, 51, 507–522. https://doi.org/10.3758/s13428-018-1098-4

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2021). *Pre-train, prompt, and predict: a systematic survey of prompting methods in natural language processing* (arXiv:2107.13586). arXiv. https://doi.org/10.48550/arXiv.2107.13586

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:1907.11692). arXiv. http://arxiv.org/abs/1907.11692

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

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). *PyTorch: An imperative style, high-performance deep learning library* (arXiv:1912.01703). arXiv. https://doi.org/10.48550/arXiv.1912.01703

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). *Language models are unsupervised multitask learners*.

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

Sakaguchi, K., Arase, Y., & Komachi, M. (2013). Discriminative approach to fill-in-the-blank quiz generation for language learners. *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 238–242. https://aclanthology.org/P13-2043

Scao, T. L., & Rush, A. M. (2021). *How many data points is a prompt worth?* (arXiv:2103.08493). arXiv. https://doi.org/10.48550/arXiv.2103.08493

Schick, T., & Schütze, H. (2021). *Exploiting cloze questions for few shot text classification and natural language inference* (arXiv:2001.07676). arXiv. https://doi.org/10.48550/arXiv.2001.07676

Susanti, Y., Tokunaga, T., Nishikawa, H., & Obari, H. (2018). Automatic distractor generation for multiple-choice English vocabulary questions. *Research and Practice in Technology Enhanced Learning*, *13*(1), 15. https://doi.org/10.1186/s41039-018-0082-z

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Lukasz, & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*. arXiv:1706.03762

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

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2019). *GLUE: A multi-task benchmark and analysis platform for natural language understanding* (arXiv:1804.07461). arXiv. https://doi.org/10.48550/arXiv.1804.07461

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., … Rush, A. M. (2020). *HuggingFace's Transformers: State-of-the-art natural language processing* (arXiv:1910.03771). arXiv. https://doi.org/10.48550/arXiv.1910.03771

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., Mihaylov, T., Ott, M., Shleifer, S., Shuster, K., Simig, D., Koura, P. S., Sridhar, A., Wang, T., & Zettlemoyer, L. (2022). *OPT: Open pre-trained transformer language models* (arXiv:2205.01068). arXiv. https://doi.org/10.48550/arXiv.2205.01068

Zellers, R., Bisk, Y., Schwartz, R., & Choi, Y. (2018). *SWAG: A large-scale adversarial dataset for grounded commonsense inference* (arXiv:1808.05326). arXiv. https://doi.org/10.48550/arXiv.1808.05326