UNIVERSITY POLITEHNICA OF BUCHAREST
Faculty of Automatic Control and Computers
Computer Science and Engineering Department

SCSS Report

Generating Multiple-Choice Questions with Large Language Models

Andreea Duțulescu

Thesis advisors:

Conf.dr.ing. Ștefan Rușeți
Prof.dr.ing. Mihai Dascălu

BUCHAREST
2024

ABSTRACT

Question generation (QG) is an increasingly crucial field in NLP, aiming to automatically formulate questions from diverse information sources. This paper investigates the utilization of large language models (LLMs) for QG tasks. Also, the process of generating challenging and appropriate distractors for multiple-choice questions is a complex and time-consuming task. Existing methods for an automated generation have limitations in proposing challenging distractors, or they fail to effectively filter out incorrect choices that closely resemble the correct answer, share synonymous meanings, or imply the same information. To overcome these challenges, we propose a comprehensive toolkit that integrates various approaches for generating distractors, including leveraging a general knowledge base and employing a T5 language model. Additionally, we introduce a novel strategy that utilizes natural language inference to increase the accuracy of the generated distractors by removing confusing options. Our models demonstrate zero-shot capabilities and achieve good results on the DGen dataset. Furthermore, our models can be fine-tuned to outperform state-of-the-art methods on the considered dataset. To further extend the analysis, we also introduce a human-annotated dataset with scores for 100 test questions with 1085 distractors in total. The evaluations indicated that our generated options have a high quality, surpass all previous automated methods, and are on par with the ground truth of human-defined alternatives.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

Question generation (QG) is a rapidly developing field in natural language processing with applications in education, assessment, and dialogue systems. The emergence of large language models (LLMs) has opened new paths for this task. LLMs, trained on massive amounts of text data, have shown a remarkable ability to understand and generate human language. This has led to significant advancements in automatic QG, with LLMs demonstrating the potential to create high-quality, diverse questions.

Moreover, multiple-choice questions are widely used in classroom quizzes to test students' general knowledge. However, manually designing such tests and choosing the most appropriate distractors to serve as possible choices along with the correct answer is a tedious task. The problem of automatically generating foils has been studied before but is far from solved. A good set of distractors in a multiple-choice setting must fulfill two main conditions in the literature regarding educational tests (Evans, 1984; Towns, 2014), namely: (1) *Validity*: they must not be synonyms or imply the same information as the correct answer; (2) *Difficulty*: they must test a deep understanding of the subject, they have to appear as valid possible answers and not be easily discarded as incorrect ones.

This paper explores the application of LLMs for question generation. We experiment with the various techniques employed to leverage LLMs for QG, including fine-tuning on question-answering datasets and prompting with specific question formats. Also, we propose a toolkit that aims to automatically generate distractors for general knowledge science questions. We use general knowledge bases and Pre-trained Language Models to generate appropriate distractors in accordance with the correct answer. Our approach pays special attention to distractor validity, as the overall correctness of a question, along with its multiple choices, is of utmost importance in a classroom quiz. We leverage a novel approach of using natural language inference to filter distractors that would be an implication, synonym, or the correct answer. Moreover, we employ a ranking mechanism to propose the most appropriate distractors as the final set for a cloze.

The main contributions of this work are as follows:

- Introduce and open-source a comprehensive toolkit that surpasses current state-of-the-art models for distractor generation on both automatic and human evaluation;
- Propose a novel filtering method based on natural language inference that ensures the relevance and validity of the proposed distractors;
- Leverage human annotations in a qualitative evaluation that argues for comparable quality between our generated and human-curated distractors.

2 STATE OF THE ART

2.1 Question Generation

Wang et al. (2020) introduced an advanced question generation model that leverages multiple knowledge sources including answer details, entity graphs, and reinforcement learning incentives to enhance question quality.

This model employs a bidirectional LSTM network with pre-trained GloVE embeddings for encoding both the document and the answer, alongside a graph network using a Graph Attention Network to map entity relationships within the context. The decoding process begins with answer embeddings and features a fusion attention mechanism rich in semantics to integrate answer-related data with the semantic graph for generating multi-hop questions. It utilizes a copying mechanism to bypass words not in its vocabulary and employs reinforcement learning to evaluate sequence quality based on BLEU-4 and Word Movers Distance scores, aiming to craft questions that closely match the reference answers.

Testing on the HotpotQA dataset (Yang et al., 2018) demonstrated marked improvements in question quality, as reflected in high metric scores and positive feedback from human reviewers regarding fluency, complexity, and relevance.

Though using entity graphs for in-depth question generation has been explored before, the complexity and capabilities of such questions have not been fully examined. There is potential for educational systems to benefit from this technology by adjusting the difficulty and inference level of the questions generated. To advance this, Fei et al. (2022) have developed a method to select a logical deduction chain that answers the question while ensuring context entities are referenced appropriately in the question.

Like previous models, this framework uses entity graphs to capture concept relationships from the context document. The graph is initially built from key entities identified in the paragraph title, with entity representations derived from context and answer data processed through a BERT model. A Graph Attention Network then aggregates this information. Relevant entities are selected to guide the answer along a specified reasoning path, with the network trained on node classification. Nodes in the reasoning chain are flagged to control question difficulty, ensuring all model constraints are met for generating the question.

This architecture set new benchmarks on the HotpotQA dataset (Yang et al., 2018), surpassing transformer-based baseline models considerably (49.71 BLEU-1, 25.09 BLEU-4, 27.45 METEOR, 41.83 ROUGE-L). It also received high marks from human evaluators across various contexts, closely matching ground truth scores.

Most question-generation research has focused on generating questions answerable from specific text spans. While this method benefits from large training datasets, it does not fully replicate human questioning behavior. Chakrabarty et al. (2022) pursued a different strategy, creating open-ended questions that necessitate summarizing a document or providing extended explanations. The responses required go beyond mere text spans and involve natural language reasoning over the context.

For this, Chakrabarty et al. (2022) introduced two datasets. The first, the Explain Like I'm Five (ELI5) subreddit, is used for training the model as it demands summarization of complex topics in simple terms. The second is a collection of New York Times articles annotated with high-quality questions for model evaluation.

Here, question generation is independent of the answer, focusing solely on the context and key factual signals. During training, the goal is to generate each question token to maximize its probability given the prior tokens and context. The authors use a two-fold approach for identifying key concepts: an unsupervised keyword mining model for training and a BERT-based model during inference. The CONSISTENT model ensures questions are relevant and answerable based on the context through a two-step filtering process using ALBERT and T0pp models.

The effectiveness of this approach is validated on a scraped New York Times dataset, showing competitive results against existing methods and receiving high human evaluation scores.

2.2 Distractor Generation

Multiple approaches and directions of study have been taken for distractor generation. They differ in scope, with some of them, like the one that we advance in the current work, focusing on distractor generation for science-related and educational questions, wherein the answer is short, often as an entity format; in contrast, other methods are generating distractors for language tests, wherein an entire document context is given at input, and the possible answers tend to have longer forms. However, the majority of these approaches have the same two main components: (1) candidate generation and (2) candidate selection.

For the purpose of generating distractors for general knowledge questions, Ren & Q. Zhu (2021) experimented with two semantic networks (i.e., Wordnet and Probase) and used a probabilistic topic model to choose words related to a certain concept. In order to further take into account the question and its semantics, Chiang et al. (2022) used pre-trained language models (e.g., BERT (Devlin et al., 2019), RoBERTa (Devlin et al., 2018), SciBERT (Beltagy et al., 2019)) to generate possible distractors for fill-in-the-blanks questions. The models were fine-tuned on the CLOTH (**?**) and DGen (Ren & Q. Zhu, 2021) datasets to generate the specific distractors while being answer-aware, since the correct answer was appended after the [SEP] token at the end of the cloze. However, as these methods generate only one-word distractors, they cannot be easily transferred to a wider variety of general-knowledge questions.

Ensuring the validity of the distractors, meaning filtering out generated samples that resemble the correct answer, has yet to be thoroughly approached. This is important because there is a high risk in the attempt to create the most challenging distractors of selecting a foil that would be the synonym or implication of the correct answer, making the question invalid for a student. Ren & Q. Zhu (2021) and Chiang et al. (2022) did not specifically filter out the invalid distractors and relied on the fine-tuning task to generate and select good candidates. H.-L. Chung et al. (2020) generated distractors for language tests and employed a regularization for copying tokens existing in the correct answer, with negative answer training. Panda et al. (2022) only filtered out WordNet synonyms (Miller, 1994) of the answer. A more focused approach was taken by Zesch & Melamud (2014) for language tests, with inference rules for filtering out invalid verbs. However, these methods are limited to concrete repetitions, context-unaware synonyms of the answer, or syntax inconsistencies. A wide area for improvement exists to ensure that the generated distractors are valid and do not confuse students.

In terms of distractor selection and ranking, multiple variants have been proposed, all considering the similarity with the correct answer. Ren & Q. Zhu (2021) learned to rank the distractors by leveraging the fine-tuning of handcrafted features on the dataset, while Chiang et al. (2022) calculated an empirically developed score for the textual characteristics. Both approaches considered multiple features, including contextual embedding similarity, morphological similarity, POS similarity, contextual likelihood, or web-search scores. Approaches that generate distractors for language tests usually rank their foils based on the log-likelihood computed by a fine-tuned language model, usually similar to the one that generates the distractors (e.g., H.-L. Chung et al., 2020, Gao et al., 2020, Qiu et al., 2020).

3 ANSWER SELECTION AND QUESTION GENERATION

3.1 Datasets

While question answering is a well-established field in NLP, question generation remains relatively under-explored. This task lacks dedicated resources and corpora and researchers have had to adapt existing question-answering datasets. This section details the datasets used in our study, which were originally designed for question-answering but were repurposed for our question-generation experiments.

SQuAD (Rajpurkar et al., 2016) is one of the most widely used resources for training and evaluating question generation models, despite being originally designed for question answering. SQuAD v1.1 consists of over 100K question-answer pairs derived from a pool of 5K Wikipedia articles. This provides a diverse range of factual topics and contexts for both questions and answers. The passages cover a broad spectrum of human knowledge. The questions are relatively easy to answer and the dataset has been saturated with QA language models. The reasoning required to answer the questions lies in the majority for syntactic and lexical variation, while about 10% requires more complex reasoning such as world knowledge and multiple-sentence reasoning.

HotpotQA (Yang et al., 2018) is a dataset designed to test a machine's ability to answer questions that require reasoning across multiple sources. Unlike SQuAD, which focuses on finding the answer within a single passage, HotpotQA presents questions that necessitate finding relevant information from two different Wikipedia articles. These questions are more complex and require not only finding the supporting facts but also understanding how those facts from different sources relate to each other to arrive at the answer. Over 40% of the questions require finding a bridge entity to connect the two contexts and provide an answer. Another key feature of HotpotQA is that it includes information about the specific sentences in the source articles that support the answer.

NarrativeQA dataset (Kočiský et al., 2018) is designed to assess reading comprehension, particularly for lengthy texts. It consists of stories, along with corresponding questions and answers. The dataset includes a variety of documents, along with summaries, links to the full stories, and the question and answer sets for each story. This dataset goes beyond simple comprehension tasks that can be solved by just focusing on nearby sentences or the overall frequency of words. Instead, it necessitates a deeper understanding of the narrative

as a whole. It addresses the limitations of existing reading comprehension datasets. Many existing datasets focus on questions that can be answered by looking at a small portion of the text, such as a single sentence, or by simply counting how many times a specific word appears throughout the document.

FairytaleQA (Xu et al., 2022) is a specialized dataset focused on narrative comprehension for kindergarten to eighth-grade students. It addresses the scarcity of high-quality question-answering datasets catering to diverse reading skills, particularly in distinguishing fine-grained narrative understanding. Constructed by educational experts, FairytaleQA encompasses both explicit text-based answers and implicit high-level summarization, reflecting varying difficulty levels. It is curated from children-friendly stories, covering seven narrative elements or relations. FairytaleQA is used for benchmarking state-of-the-art QA models, revealing challenging areas for these models and enabling a finer-grained analysis of comprehension sub-skills. Furthermore, it supports question generation, displaying the capability of generating diverse and higher-quality questions. The questions are annotated on three dimensions that can be referenced: whether the answer is local or can be found through the summary, the explicit or implicit (requesting inference) nature of the reasoning, and the attribute of the task (causal relation, action, prediction, etc).

3.2 Method

Question generation for reading comprehension comprises two parts: (1) selecting the piece of information that will serve as the answer/knowledge evaluated and (2) generating a question whose correct answer should be the previous selection.

We investigated the effectiveness of fine-tuning large language models for question generation. We explore a multi-task learning approach using three question-answering datasets: SQuAD, HotpotQA, and NarrativeQA. The goal is to leverage the inherent relationship between questions, answers, and context to enhance the model's ability to generate novel and relevant questions. Two LLM architectures were employed for fine-tuning: Flan-T5 (H. W. Chung et al., 2022) with 3B parameters and Qwen (Bai et al., 2023) with 7B parameters. Flan-T5 is an encoder-decoder model, while Qwen is a decoder-only architecture. To improve training efficiency for Qwen's larger size, we employed Low-Rank Adaptation (LoRA) (Hu et al., 2021) with half-precision training. LoRA is a novel technique for fine-tuning large language models LLMs on specific tasks. It achieves this by freezing the model weights and introducing a small number of additional parameters in the form of low-rank matrices. This reduces the number of trainable parameters by a large margin while preserving the performance of classic supervised fine-tuning. In our setup, the remaining trainable weights amount to 0.86% of the total number of parameters.

The fine-tuning process involved three distinct tasks, all of them using the `<context>`,

<question>, and <answer> annotations of the datasets:

- Answer Selection: The model is trained to identify an answer within a given context that is most suitable for generating a question. This step helps the model focus on informative elements within the passage. The prompt that we use is: *"Select an answer from the context that can be used to generate a question. Context: <context>"* and the desired generation is *"<answer>"*.
- Question Generation: Here, the model learns to formulate a question based on both the provided context and the selected answer. The prompt is *"Generate a question based on the context and the answer. Context: <context>. Answer: <answer>"* and the generation is *"<question>"*.
- Question Answering: The final task reinforces the model's comprehension of the relationship between questions, answers, and context. By training on existing QA pairs, the model strengthens its understanding of these elements. The prompt is *"Answer the following question based on the context. Context: <context>. Question: <question>"* and the desired generation is *"<answer>"*.

This multi-task fine-tuning approach aims to achieve several benefits. Firstly, it allows the model to exploit the complementary nature of these tasks. By jointly learning answer selection, question generation, and question answering, the model can develop a more holistic understanding of how questions and answers relate to the underlying context.

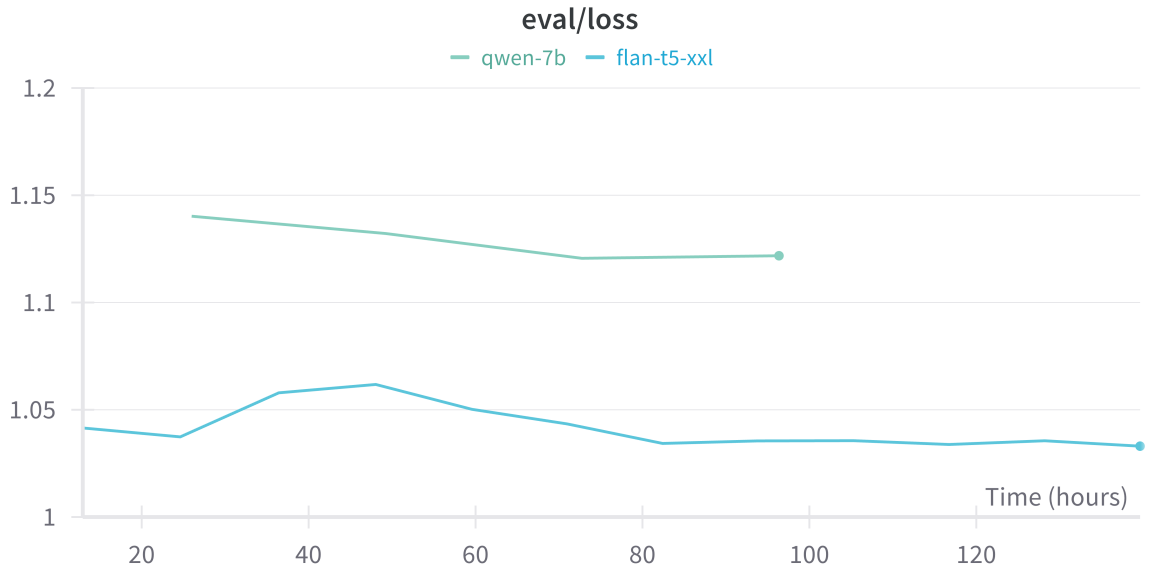

Figure 1: Loss evolution on the evaluation partition during training.

3.3 Experimental Setup

Both language models are obtained from the Huggingface transformers library (Wolf et al., 2020). The finetuning was done, for both models in 1 epoch. The learning rate for the Flan-T5 model [1] was 1e-4 with the AdamW optimizer. For the Qwen model [2], the learning rate was 5e-5, AdaFactor optimizer, trained in half-precision with LoRA, training only the attention layers.

3.4 Evaluation Metrics

We evaluated the models for all three tasks and compared the generation with the dataset ground truth, computing the BERTScore metric (Zhang et al., 2020).

3.5 Results

Table 1 shows the scores obtained on a sample of the test partition from all datasets for both models with the greedy decoding strategy.

Model	Answer Selection	Question Generation	Question Answering
Qwen	**0.85**	**0.90**	**0.94**
Flan-T5	0.81	**0.90**	**0.94**

Table 1: Comparison results (greedy decoding).

Tables 2 and 3 showcase the results for each generation strategy in order to highlight the best approach.

Task	Greedy	Top-k	Top-p	Beam Search
Answer Selection	0.81	0.81	0.81	**0.82**
Question Generation	**0.90**	0.88	0.89	**0.90**
Question Answering	**0.94**	0.93	**0.94**	0.93

Table 2: BERTScore for Flan-T5.

3.6 Discussion

The data presented in Table 1 indicate that the Qwen model demonstrates superior performance, particularly in the domain of answer selection tasks, where it achieves a higher

[1] https://huggingface.co/google/flan-t5-xl

[2] https://huggingface.co/Qwen/Qwen1.5-7B

Task	Greedy	Top-k	Top-p	Beam Search
Answer Selection	**0.85**	0.84	**0.85**	**0.85**
Question Generation	**0.90**	0.88	0.89	0.89
Question Answering	**0.93**	0.92	**0.93**	0.92

Table 3: BERTScore for Qwen.

BERTScore compared to the Flan-T5 model. Both models, however, perform equally well in question generation and question-answering tasks. Flan-T5 offers advantages in terms of computational efficiency, with faster inference and training times, rendering it a more lightweight alternative.

The convergence of performance across both models on larger datasets suggests a plateau in enhancement, as evidenced by similar scores in question generation and question-answering tasks for both models. This implies that, beyond a certain data volume, the incremental gains from additional model complexity diminish.

Furthermore, the results from Tables 2 and 3, which detail performance across different decoding strategies, suggest that greedy decoding is generally the most effective, consistently yielding the best or second-best results across all tasks. This strategy is especially advantageous for its simplicity and direct approach to producing high-quality outputs. Nevertheless, for when diversity in generated content is desired over optimal correctness, alternative decoding strategies such as top-k, top-p, and beam search can be used for generating a wider array of possible outputs.

4 DISTRACTOR GENERATION

4.1 Datasets

DGen (Ren & Q. Zhu, 2021) is a collection of multiple-choice, fill-in-the-blank science questions. It covers various domains like science, common sense, vocabulary, and trivia. The authors compiled the dataset from several open-source MCQ sources. Each item contains the sentence, the correct answer, and three distractors. The length of the foils is one word only.

4.2 Method

Figure 2 showcases an overview of the pipeline employed for this task, with each component detailed in its corresponding sub-section.

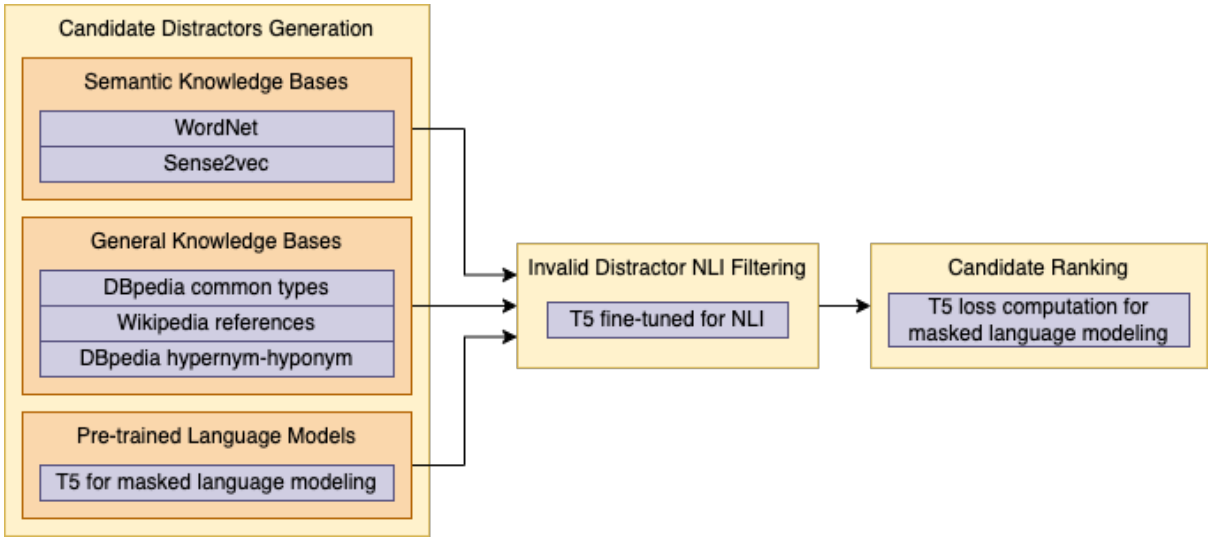

Figure 2: Overall architecture of the distractor generation method.

4.2.1 Candidate Generation

We generate multiple-choice candidates with various methods, as we want to provide a comprehensive toolkit with multiple complementary methods (i.e., semantic and general knowledge bases and LLMs) readily integrated that cover and generalize well for diverse school subjects. These methods generate distractors in a similar manner, followed by a semantic filtering and

candidate ranking procedure. In the case of DGen, the generated candidates are split into individual words to fulfill the restriction of single-word answers.

Semantic Knowledge Bases Semantic databases are used to compute possible distractors for a given word by employing existing links between concepts. Our considered alternatives are WordNet (Miller, 1994) and Sense2Vec (Trask et al., 2015). From WordNet, we use the hypernym-hyponym relation in which different items of the same class as the correct answer (i.e., siblings in the hypernym taxonomy with the answer) are selected to participate in the candidate set. Sense2Vec ensures a broader and more permissive search to select the most similar concepts regarding the embedding representation of the correct answer.

General Knowledge Bases As the main focus of this task is to generate foils for science, general knowledge, and school subjects, the search space for possible distractors must encompass a broad area of topics, knowledge, and information. For this purpose, a large multi-purpose semantic repository is used, namely DBpedia (Lehmann et al., 2015). As a rich knowledge base, DBpedia is an underexplored but valuable resource for this task, as it contains loads of open-source, structured information that can be easily queried to find complex relations between different concepts.

In order to start searching for appropriate distractors, the DBpedia resource corresponding to the answer must be located. In this regard, the lookup API service is used to gather the URIs of the entities related to the answer. The proposed entities are ranked with a pre-trained language model that computes the likelihood of that entity label being generated by a language model instead of the actual answer stem. The entity with the highest probability is used as the corresponding answer resource.

Three approaches for generating candidates with DBpedia are employed:

- Selecting the entities of the same class as the answer with the highest number of common types with the answer;
- Extracting the entities referred to by links in the Wikipedia page corresponding to the answer;
- Picking the entities with the same hypernym as the correct answer.

Pre-trained Language Models A pre-trained language model, namely T5 (Roberts et al., 2019), is used to generate candidate foils for quiz questions. The general approach is to generate the most likely stems that would either fill in or answer a certain question without additional context. A list of possible candidates for the answer is sampled from the model's distribution. T5 was chosen for its masked language prediction pre-training objective, which perfectly matches the fill-in-the-blanks scenario in the dataset. T5 was chosen for its masked language prediction pre-training and versatile zero-shot capabilities, ensuring quality results even with limited resources and adaptability to various educational contexts.

In the case of fill-in-the-blank items, the question comes as a sequence of text containing a marked blank space. That space should be completed with a choice that maintains the sequence's fluency, context, and factual correctness. For this sub-task, the blank space is replaced with a special masking token near which a reference to the correct answer is appended, thus ensuring that the model is answer-aware, guiding the generation towards a different answer. This prompt is forwarded to a T5 model, more specifically, in the form of "<mask_token> (or gravity) causes rocks to roll downhill.", for the item: "____ causes rocks to roll downhill.", and the correct answer: "gravity".

4.2.2 Invalid Distractors Filtering

We consider it highly important not to propose distractors that may invalidate the test item, such as distractors containing, implicating, or having the same meaning as the correct answer. These invalid distractors are filtered out by leveraging natural language inference models to detect whether the correct answer implies the proposed foils. More formally, two sequences are computed: one that contains the correct answer (e.g., "igneous rocks form from cooled magma or lava."), and one with the generated candidate (e.g., "granite rocks form from cooled magma or lava."). If the first sentence entails the second one, it means that the candidate is not suitable, as it can be inferred from the correct answer, making the test item invalid since the question would have two correct answers. A T5 model fine-tuned for natural language inference is used for this part. If the model output for a (correct answer sentence, candidate sentence) pair is an entailment, then the candidate is not part of the final proposal.

Along with this filtering, we also discarded distractors that do not have the same part of speech as the correct answer for the dataset format.

4.2.3 Candidate Ranking

At this stage, the remaining candidates are valid choices that would be proposed. Their ranking for the final result is established as the likelihood of being the correct answer since the correct answers should have been filtered out in the previous stage. The score is computed as the negative log-likelihood of the candidate being generated instead of a special token. A T5 model is used for this, and the prompt forwarded to the model has the same format as the one described in Subsection 4.2.1, except this time the model will not be used for inference, but rather to compute the Cross-Entropy loss for a given label (the candidate). The loss value is the score computed for that candidate, and a lower score will imply a better candidate for the distractor.

We experimented with two different approaches: (1) a zero-shot setting in which a vanilla T5 model is used to compute the score and (2) a fine-tuned T5 model on the train partition of the dataset. The fine-tuning was done by forwarding the prompt with the ground-truth

distractors replacing the special tokens.

4.3 Experimental Setup

Knowledge Bases We considered WordNet from NLTK (Bird & Loper, 2004) and a library for Sense2vec search[1], where we selected the top 20 most similar entities. We retrieved the top 10 results using the lookup service[2] to gather DBpedia URIs for the correct answer and selected the most appropriate ones according to a pre-trained T5 model. SPARQL queries were run on DBpedia using their endpoint[3].

Language Models All our language models are obtained from the Huggingface Transformers library (Wolf et al., 2020). For tasks that involve the masked language approach, such as ranking lookup results and generating and ranking distractor candidates, we use the T5v1.1 model[4]. For the fine-tuned distractor ranking variant, we adapted it on the Train partition of the DGen dataset for 3 epochs, with the 1e-4 learning rate and AdamW optimizer. For the NLI filtering, we used the T5 fine-tuned model[5] (Honovich et al., 2022).

4.4 Evaluation Metrics

We replicated the setup for the automatic evaluation on which Chiang et al. (2022) evaluated their performance. We considered relevant the following metrics:

- P@1 - the precision of having the first proposed candidate in the ground-truth set;
- F1@X - the F1 measure of the first X proposed candidates in regard to the ground-truth set;
- MRR@X - the mean reciprocal rank measure (Craswell, 2009) of the first X proposed candidates in regard to the ground-truth set;
- NDCG@X - the normalized discounted cumulative gain (Järvelin & Kekäläinen, 2002) of the first X proposed candidates in regard to the ground-truth set.

4.5 Results

The evaluation was performed on the test partition of the dataset with the metrics described above and is presented in Table 4. We reported our best approaches (ranking fine-tuning with

[1] https://github.com/explosion/sense2vec
[2] https://lookup.dbpedia.org/api/search
[3] http://dbpedia.org/sparql
[4] `https://huggingface.co/google/t5-v1_1-xl`
[5] `https://huggingface.co/google/t5_xxl_true_nli_mixture`

and without NLI filtering) and the current state-of-the-art on this dataset, CDGP (Chiang et al., 2022).

CDGP leverages pre-trained language models for generating distractor candidates. It comprises two stages: Candidate Set Generator and Distractor Selector. For distractor generation, pre-trained language models are fine-tuned to predict dataset distractors using the question stem and answer. Distractor candidates are ranked based on features like confidence scores, word embedding similarity, sentence-level contextual similarity, and part-of-speech matching. The top candidates with the highest scores are selected as final distractors.

Our method achieves higher or comparable results on all metrics, especially on P@1, meaning that our first proposed distractor is often found also in the ground-truth set.

Method	P@1	F1@3	F1@5	MRR@5	MRR@10	NDCG@5	NDCG@10
CDGP	12.40	12.74	**12.93**	23.22	25.00	28.49	**34.42**
Ours w/ NLI	18.53	12.22	11.38	24.51	25.40	28.17	31.02
Ours w/o NLI	**20.07**	**13.25**	11.58	**26.15**	**27.06**	**30.14**	32.97

Table 4: Automatic evaluation of our methods with the current state-of-the-art

Moreover, we conducted a study to assess how different approaches for the candidate generation perform by separately using each component with NLI filtering and ranking fine-tuning (see Table 5). The component that leveraged a pre-trained language model is by far the most influential in performance, as it contextualizes the semantic meaning of a possible answer in the sentence. The semantic knowledge base approach is the next best one since it provides related candidates that closely resemble the answer without considering the context at generation time. Leveraging a general knowledge base such as DBpedia to generate related distractors for an entity performs worse than the other two variants; however, it is still a valuable resource of complementary initial alternatives.

Method	P@1	F1@3	F1@5	MRR@5	MRR@10	NDCG@5	NDCG@10
Semantic KB	13.12	7.85	6.75	16.20	16.52	18.22	19.54
General KB	8.30	4.44	4.72	11.53	12.13	13.79	15.66
PLM	**17.76**	**10.68**	**9.55**	**22.91**	**23.20**	**26.74**	**27.80**

Table 5: Automatic evaluation of different candidate generation methods

4.6 Discussion

4.6.1 Error Analysis

Given the performance reported from the automatic evaluation, we expected some cases where our models failed to perform as expected. Nevertheless, our assumption is that our distractors

are not necessarily of lower quality, although different, than the ground truth. A large number of possible distractors are suitable for a specific item, so the comparison with just three choices available on the dataset is not enough. As seen in Figure 3, some of our candidate distractors are as suitable as the ones proposed, even more challenging since, in this particular example, we generated foils actually part of the respiratory system and are not easily dismissable as incorrect. However, the overlap between our candidates and the ground truth from the dataset was rather low. This highlighted the need for human evaluation, as we cannot conclude that a choice is bad because it is not found in the proposed set.

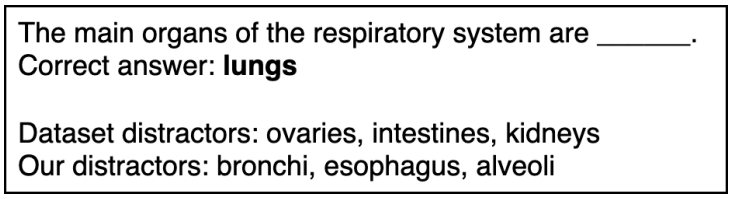

Figure 3: Example of good distractors highly related but with no overlap with the answer.

Other cases of mismatch between the ground truth and our distractors are caused by the NLI filtering. In particular cases, an implication is detected but not with high confidence; thus, the candidate foil is filtered out. In other specific cases, such as the one depicted in Figure 4, some ground-truth distractors (i.e., *granite* is an *igneous* rock) are actually an implication of the correct answer and, hence, invalid.

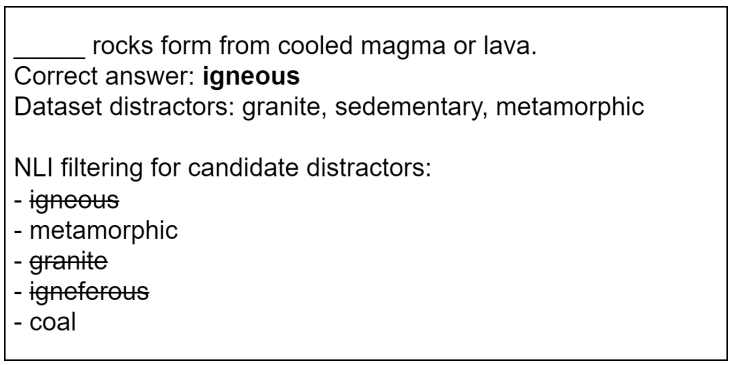

Figure 4: Example of NLI filtering

4.6.2 Human Evaluation

As we argue that the comparison with 3 ground-truth distractors is far from suitable for assessment and for comparing the effectiveness of an automated model, we experimented with human annotators to decide the best distractors generated with different methods. Four participants were involved in this undertaking, all possessing proficient English language abilities. The initial annotation phase involved the participation of three undergraduate students, while the adjudication stage included a Ph.D. student as the fourth rater. In order to mitigate previous knowledge bias, the participants were encouraged to search the Internet or other resources for knowledge, but without using AI generative models.

Annotation Methodology

The first 100 entries of the DGen's test partition were used for human annotation. For the annotation data, we employed and generated distractors with the 6 methods targeted for comparison and selected the top 3 candidates from each approach:

- DGen dataset (Ren & Q. Zhu, 2021) distractors;
- Distractors proposed by Chiang et al. (2022).

Our proposed distractors:

- without NLI filtering and without ranking fine-tuning (w/o NLI, w/o FT);
- without NLI filtering and with ranking fine-tuning (w/o NLI, w/ FT);
- with NLI filtering and without ranking fine-tuning (w/ NLI, w/o FT);
- with NLI filtering and ranking fine-tuning (w/ NLI, w/ FT).

The above distractors were combined in a set of candidates for each item. Three annotators were asked to assign a score to each candidate in the set based on the question and correct answer provided. The scores were assigned as follows:

- 0 - Invalid: This distractor repeats the answer, is a synonym of the answer, or is correct in the context of the question, making the item have multiple correct answers;
- 1 - Poor: This distractor is not related to the question, would not fit in the sentence, is easily identifiable as incorrect, does not test knowledge of the subject;
- 2 - OK: This distractor is related to the question and would fit in the sentence but is relatively easily identifiable as incorrect, and it tests a shallow understanding of the subject;
- 3 - Good: This distractor is related to the question, shares the semantic field with the correct answer, would fit in the sentence, it can be easily mistaken as being the correct answer or has very few distinctions with the correct answer, and it tests a deep understanding of the subject.

As a further notation, each proposed distractor received a score $adn_{i,j}^{k}$ denoting the value assigned by the annotator k for the distractor i of the set corresponding to the test item j. Distractors from the same set appeared in a randomized order, and every annotator assigned a score for all candidates. At this stage of the annotation phase, we computed the inter-rater agreement as the ICC(2,k) correlation (Cicchetti, 1994). We obtained a score of 0.67, which is rated by Cicchetti (1994) as a good agreement.

A fourth annotator was involved in the adjudication process when any of the following situations were encountered. First, if only one rater considered an option as invalid (i.e., a score of 0), while the others assigned scores between 1 and 3; the goal of this double-check is

to ensure a clear separation between valid and invalid disruptors, as it is possible that the other annotators missed a reason why the candidate is actually a correct answer. Second, the fourth annotator covered all cases where annotators had given three different scores to the same distractor, so a majority could not be established. The final score of a candidate distractor was the mode rating given by the three annotators.

More formally, $adn\_score_{i,j}$ is the final score of distractor i from the set corresponding to the test item j calculated as:

$$adn\_score_{i,j} = Mode(adn^1_{i,j}; adn^2_{i,j}; adn^3_{i,j}) \tag{1}$$

Annotations Results

We first computed the average score of the top 3 distractors for each method in regard to the score assigned after the human annotation phase.

More formally, $score_{m,j}$ is the average score obtained by the top 3 distractors $top3_m$ generated with method m for the test item j, in regards to the average rating computed at Equation 1, calculated as:

$$score_{m,j} = \frac{1}{3} \sum_{i \in top3_m} adn\_score_{i,j} \tag{2}$$

The scores for each method are plotted in Figure 5. Based on the human annotations that take into account the quality of a distractor, our method with fine-tuning yields scores comparable to the dataset's ground truth. This demonstrates the high performance of our method which matches the quality of the manually curated dataset and emphasizes the necessity of relying on human annotation rather than relying solely on ground-truth comparisons in the context of foil generation.

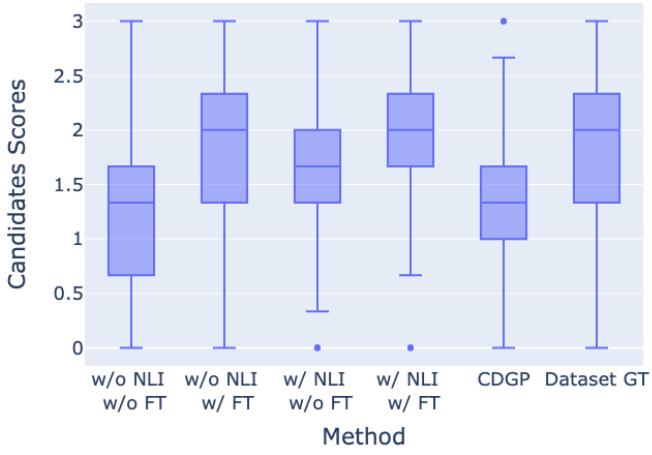

Figure 5: Human annotation scores distribution

Moreover, we observe the importance of employing NLI filtering to ensure a sound approach. Our corresponding variants significantly surpass the current state-of-the-art and obtain a similar quality as the ground truth. The fact that our method employing just NLI filtering manages to have high-quality distractors is especially important since it proves zero-shot capabilities and impressive adaptability for future domains and items. Our method that employs NLI filtering, without any fine-tuning, receives higher results than the current state-of-the-art fine-tuned on the dataset. Moreover, with no fine-tuning, the NLI filtering has a high impact on the performance, as can be observed from the w/o NLI w/o FT versus w/ NLI w/o FT comparison.

Considering that the data are not part of a parametric distribution, the comparison between methods is conducted by first employing the Friedman test. According to Pereira et al. (2015), the application of the Friedman test can be subsequently complemented by Wilcoxon post-hoc pairwise comparisons, incorporating the Bonferroni adjustment, in order to see which of the differences in average scores highlighted by Figure 5 are relevant.

For the Friedman test, we state the null hypothesis threshold as 0.001. The Friedman test values (w $= 0.191$, p $= 4.41 \cdot 10^{-9}$) highlight significant differences between the methods' performances. Moreover, the above findings drawn from Figure 5 are supported by computing the pairwise Wilcoxon scores (see Table 6). Specifically, there is no significant difference (as demonstrated by the above-the-threshold adjusted p-values) in the quality scores obtained by our methods that employ NLI filtering, ranking fine-tuning, or both of them, and the dataset's ground truth. However, a large difference (characterized by under-the-threshold adjusted p-values) is observed between these methods and the current state-of-the-art.

	w/o NLI w/ FT	**w/ NLI w/o FT**	**w/ NLI w/ FT**	**CDGP**	**Dataset GT**
w/o NLI, w/o FT	-5.95 (***)	-8.16 (***)	-6.59 (***)	-2.04 (-)	-5.61 (***)
w/o NLI, w/ FT		-3.73 (-)	-8.23 (*)	-5.65 (***)	-3.01 (-)
w/ NLI, w/o FT			-5.48 (-)	-5.77 (***)	-3.10 (-)
w/ NLI, w/ FT				-6.11 (***)	-3.37 (-)
CDGP					-6.30 (***)

Table 6: Wilcoxon post-hoc pairwise comparisons (z-values and significance)
The significance is calculated based on the p-value of the test, as follows: inconclusive (-) $p > 0.05$, slightly conclusive (*) $p < 0.05$, highly conclusive (***) $p < 0.001$.

Table 7 highlights what type of distractors each method tends to propose and the improvements introduced by different variations of our approach. In terms of invalid distractors (i.e., a score of 0), our NLI filtering manages to reduce them by almost 50% in regards to the current state-of-the-art. Moreover, our best method has the highest frequency of distractors rated as high-quality (i.e., a score of 3).

Method	**0**	**1**	**2**	**3**
w/o NLI, w/o FT	107	71	49	73
w/o NLI, w/ FT	66	63	60	111
w/ NLI, w/o FT	43	96	72	89
w/ NLI, w/ FT	41	72	78	109
CDGP	82	102	49	67
Dataset GT	28	93	86	93

Table 7: The number of distractors with a certain score

5 CONCLUSIONS AND FUTURE WORK

In conclusion, this report introduces a comprehensive toolkit for generating questions and distractors, leveraging knowledge bases, and pre-trained language models. Additionally, we include a filtering method based on natural language inference that substantially reduces the generation of invalid distractors. By leveraging NLI, we can robustly eliminate confusing options that share similar meanings or represent alternative correct answers, ensuring that the generated distractors engage learners in a sound way. This feature sets our toolkit apart from existing methods that often fail to effectively filter out incorrect choices. The proposed method surpasses current standards in both automatic and human evaluation and is comparable with the quality of human-curated distractors.

More importantly, we go beyond relying solely on automatic evaluation measures and highlight the value of manual assessment with sound statistical validations. Involving human annotators in the evaluation process reveals that automated measures alone are inadequate for accurately assessing the quality of generated distractors.

Our approach is easily adaptable to various changes. The two state-of-the-art language models that we used (vanilla T5 and T5 fine-tuned for NLI) can be replaced with smaller or different language models, given that they have masked token prediction capabilities for distractor generation (e.g., BERT; Devlin et al., 2019) and NLI training (e.g., DeBERTa; He et al., 2021).

As many traditional questions are used in quizzes and knowledge tests, a future study direction can lead towards adapting the current method to also consider these types of questions, not only fill-in-the-blanks sentences. The T5 model can be easily adapted to generate possible answers to such questions; however, the NLI filtering requires the premise and hypothesis to be in the form of statements. This can be artificially achieved by a language model that makes this transformation. Nevertheless, the quality and effectiveness of different kinds of questions cannot be guaranteed, and further effort is needed to study the possibility of achieving higher generalizability.

REFERENCES

Bai, J., Bai, S., Chu, Y., Cui, Z., Dang, K., Deng, X., ... others (2023). Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Beltagy, I., Lo, K., & Cohan, A. (2019, November). SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)* (pp. 3615–3620). Hong Kong, China: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D19-1371 doi: 10.18653/v1/D19-1371

Bird, S., & Loper, E. (2004, July). NLTK: The natural language toolkit. In *Proceedings of the ACL interactive poster and demonstration sessions* (pp. 214–217). Barcelona, Spain: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P04-3031

Chakrabarty, T., Lewis, J., & Muresan, S. (2022, December). CONSISTENT: Open-ended question generation from news articles. , 6954–6968. Retrieved from https://aclanthology.org/2022.findings-emnlp.517

Chiang, S.-H., Wang, S.-C., & Fan, Y.-C. (2022, December). CDGP: Automatic cloze distractor generation based on pre-trained language model. In *Findings of the association for computational linguistics: Emnlp 2022* (pp. 5835–5840). Abu Dhabi, United Arab Emirates: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2022.findings-emnlp.429

Chung, H.-L., Chan, Y.-H., & Fan, Y.-C. (2020, November). A BERT-based distractor generation scheme with multi-tasking and negative answer training strategies. In *Findings of the association for computational linguistics: Emnlp 2020* (pp. 4390–4400). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.findings-emnlp.393 doi: 10.18653/v1/2020.findings-emnlp.393

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., ... Wei, J. (2022). *Scaling instruction-finetuned language models.*

Cicchetti, D. V. (1994). Guidelines, criteria, and rules of thumb for evaluating normed and standardized assessment instruments in psychology. *Psychological assessment*, *6*(4), 284.

Craswell, N. (2009). Mean reciprocal rank. In L. LIU & M. T. ÖZSU (Eds.), *Encyclopedia of database systems* (pp. 1703–1703). Boston, MA: Springer US. Retrieved from https://doi.org/10.1007/978-0-387-39940-9_488 doi: 10.1007/978-0-387-39940-9_488

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, *abs/1810.04805*. Retrieved from http://arxiv.org/abs/1810.04805

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019, June). 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)* (pp. 4171–4186). Minneapolis, Minnesota: Association for Computational Linguistics. Retrieved from https://aclanthology.org/N19-1423 doi: 10.18653/v1/N19-1423

Evans, W. (1984). Test wiseness: An examination of cue-using strategies. *The Journal of Experimental Educational*, 141–144.

Fei, Z., Zhang, Q., Gui, T., Liang, D., Wang, S., Wu, W., & Huang, X. (2022, May). CQG: A simple and effective controlled generation framework for multi-hop question generation. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 6896–6906). Dublin, Ireland: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2022.acl-long.475 doi: 10.18653/v1/2022.acl-long.475

Gao, L., Gimpel, K., & Jensson, A. (2020, July). Distractor analysis and selection for multiple-choice cloze questions for second-language learners. In *Proceedings of the fifteenth workshop on innovative use of nlp for building educational applications* (pp. 102–114). Seattle, WA, USA → Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.bea-1.10 doi: 10.18653/v1/2020.bea-1.10

He, P., Liu, X., Gao, J., & Chen, W. (2021). Deberta: Decoding-enhanced bert with disentangled attention. In *International conference on learning representations*. Retrieved from https://openreview.net/forum?id=XPZIaotutsD

Honovich, O., Aharoni, R., Herzig, J., Taitelbaum, H., Kukliansy, D., Cohen, V., ... Matias, Y. (2022, July). TRUE: Re-evaluating factual consistency evaluation. In *Proceedings of the 2022 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 3905–3920). Seattle, United States: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2022.naacl-main.287 doi: 10.18653/v1/2022.naacl-main.287

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., ... Chen, W. (2021). *Lora: Low-rank adaptation of large language models.*

Järvelin, K., & Kekäläinen, J. (2002). Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, *20*(4), 422–446.

Kočiský, T., Schwarz, J., Blunsom, P., Dyer, C., Hermann, K. M., Melis, G., & Grefenstette, E. (2018). The NarrativeQA reading comprehension challenge. *Transactions of the Association*

for Computational Linguistics, *6*, 317–328. Retrieved from https://aclanthology.org/Q18-1023 doi: 10.1162/tacl_a_00023

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., ... others (2015). Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, *6*(2), 167–195.

Miller, G. A. (1994). WordNet: A lexical database for English. In *Human Language Technology: Proceedings of a workshop held at Plainsboro, New Jersey, March 8-11, 1994.* Retrieved from https://aclanthology.org/H94-1111

Panda, S., Palma Gomez, F., Flor, M., & Rozovskaya, A. (2022, May). Automatic generation of distractors for fill-in-the-blank exercises with round-trip neural machine translation. In *Proceedings of the 60th annual meeting of the association for computational linguistics: Student research workshop* (pp. 391–401). Dublin, Ireland: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2022.acl-srw.31 doi: 10.18653/v1/2022.acl-srw.31

Pereira, D. G., Afonso, A., & Medeiros, F. M. (2015). Overview of friedman's test and post-hoc analysis. *Communications in Statistics-Simulation and Computation*, *44*(10), 2636–2653.

Qiu, Z., Wu, X., & Fan, W. (2020, December). Automatic distractor generation for multiple choice questions in standard tests. In *Proceedings of the 28th international conference on computational linguistics* (pp. 2096–2106). Barcelona, Spain (Online): International Committee on Computational Linguistics. Retrieved from https://aclanthology.org/2020.coling-main.189 doi: 10.18653/v1/2020.coling-main.189

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016, November). SQuAD: 100,000+ questions for machine comprehension of text. In J. Su, K. Duh, & X. Carreras (Eds.), *Proceedings of the 2016 conference on empirical methods in natural language processing* (pp. 2383–2392). Austin, Texas: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D16-1264 doi: 10.18653/v1/D16-1264

Ren, S., & Q. Zhu, K. (2021, May). Knowledge-driven distractor generation for cloze-style multiple choice questions. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(5), 4339-4347. Retrieved from https://ojs.aaai.org/index.php/AAAI/article/view/16559 doi: 10.1609/aaai.v35i5.16559

Roberts, A., Raffel, C., Lee, K., Matena, M., Shazeer, N., Liu, P. J., ... Zhou, Y. (2019). *Exploring the limits of transfer learning with a unified text-to-text transformer* (Tech. Rep.). Google.

Towns, M. H. (2014). Guide to developing high-quality, reliable, and valid multiple-choice assessments. *Journal of Chemical Education*, *91*(9), 1426–1431.

Trask, A., Michalak, P., & Liu, J. (2015). sense2vec-a fast and accurate method for word sense disambiguation in neural word embeddings. *arXiv preprint arXiv:1511.06388*.

Wang, L., Xu, Z., Lin, Z., Zheng, H., & Shen, Y. (2020, December). Answer-driven deep question generation based on reinforcement learning. In *Proceedings of the 28th international conference on computational linguistics* (pp. 5159–5170). Barcelona, Spain (Online): International Committee on Computational Linguistics. Retrieved from https://aclanthology.org/2020.coling-main.452 doi: 10.18653/v1/2020.coling-main.452

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... Rush, A. M. (2020, October). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: System demonstrations* (pp. 38–45). Online: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/2020.emnlp-demos.6

Xu, Y., Wang, D., Yu, M., Ritchie, D., Yao, B., Wu, T., ... Warschauer, M. (2022, May). Fantastic questions and where to find them: FairytaleQA – an authentic dataset for narrative comprehension. In S. Muresan, P. Nakov, & A. Villavicencio (Eds.), *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 447–460). Dublin, Ireland: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2022.acl-long.34 doi: 10.18653/v1/2022.acl-long.34

Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W., Salakhutdinov, R., & Manning, C. D. (2018, October-November). HotpotQA: A dataset for diverse, explainable multi-hop question answering. In E. Riloff, D. Chiang, J. Hockenmaier, & J. Tsujii (Eds.), *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 2369–2380). Brussels, Belgium: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D18-1259 doi: 10.18653/v1/D18-1259

Zesch, T., & Melamud, O. (2014, June). Automatic generation of challenging distractors using context-sensitive inference rules. In *Proceedings of the ninth workshop on innovative use of NLP for building educational applications* (pp. 143–148). Baltimore, Maryland: Association for Computational Linguistics. Retrieved from https://aclanthology.org/W14-1817 doi: 10.3115/v1/W14-1817

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2020). Bertscore: Evaluating text generation with bert. In *International conference on learning representations*. Retrieved from https://openreview.net/forum?id=SkeHuCVFDr