Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.062004

ARTICLE

Joint Generation of Distractors for Multiple-Choice Questions: A Text-to-Text Approach

Ricardo Rodriguez-Torrealba, Eva Garcia-Lopez* and Antonio Garcia-Cabot

Departamento de Ciencias de la Computación, Universidad de Alcalá, Alcalá de Henares, Madrid, 28801, Spain

*Corresponding Author: Eva Garcia-Lopez. Email: eva.garcial@uah.es

Received: 08 December 2024; Accepted: 18 March 2025; Published: 16 April 2025

ABSTRACT: Generation of good-quality distractors is a key and time-consuming task associated with multiple-choice questions (MCQs), one of the assessment items that have dominated the educational field for years. Recent advances in language models and architectures present an opportunity for helping teachers to generate and update these elements to the required speed and scale of widespread increase in online education. This study focuses on a text-to-text approach for joints generation of distractors for MCQs, where the context, question and correct answer are used as input, while the set of distractors corresponds to the output, allowing the generation of three distractors in a single model inference. By fine-tuning FlanT5 models and LongT5 with TGlobal attention using a RACE-based dataset, the potential of this approach is explored, demonstrating an improvement in the BLEU and ROUGE-L metrics when compared to previous works and a GPT-3.5 baseline. Additionally, BERTScore is introduced in the evaluation, showing that the fine-tuned models generate distractors semantically close to the reference, but the GPT-3.5 baseline still outperforms in this area. A tendency toward duplicating distractors is noted, although models fine-tuned with Low-Rank Adaptation (LoRA) and 4-bit quantization showcased a significant reduction in duplicated distractors.

KEYWORDS: Text-to-text; distractor generation; fine-tuning; FlanT5; LongT5; multiple-choice; questionnaire

1 Introduction

The educational field employs a diverse array of assessment instruments, each serving different purposes and learning outcomes. Among those, multiple-choice quizzes have been fundamental, playing an important role over the years [1]. Today, they continue to be a valuable assessment tool [2], and their success and acceptance in education are due to two main reasons. First, they facilitate the measurement of different types and levels of acquired knowledge in different domains, including measuring higher-order cognitive abilities such as synthesis and problem-solving [3]. Secondly, they are easy and quick to manage, allowing objective qualifications [4].

The basic structure of a multiple-choice item consists of three elements: the question to be answered (also called stem), the correct answer, and the incorrect (or in some cases, partially incorrect) options called distractors [2,5,6]. In some cases, the stem also includes the context of the question, something common in reading comprehension assessments. When developing multiple-choice questions (MCQs), selecting plausible and effective distractors is crucial for setting the difficulty level of items, minimizing random guessing, and distinguishing between the different cognitive levels of students [5,6].

Latest advancements in Natural Language Processing (NLP) and new Large Language Models (LLMs) offer the possibility to assist educators in routine tasks like assessment generation, so they can spend more

time with the students, motivating them, and sharing their knowledge [7]. Although distractor generation (DG) is a key and time-consuming component of MCQs for student assessments, it has not gained as much attention in the natural language processing (NLP) community as other tasks like question answering (QA) or question generation (QG) [8]. This lack of popularity can be attributed to several factors, including the absence of standard benchmarks, metrics, and specific datasets dedicated to DG [9].

However, recent studies have begun to bridge this gap by exploring the automatic generation of distractors using Reading Comprehension (RC) datasets and MCQs datasets as a source of data [9–13]. This shift shows an emerging interest in understanding how the DG task can be improved.

The fundamental role of distractors in MCQs is to confuse students by introducing plausible alternatives that challenge their understanding and application of knowledge. If this is not achieved, the quality of the multiple-choice item is compromised, as students might identify the correct answer without requiring the application of the knowledge or skills that are being assessed [13].

To address this problem, guidelines and recommendations for manually generating high-quality distractors have been developed in the past [4]. However, a concrete methodology for evaluating the quality of automatically generated distractors remains elusive. Currently, researchers rely on standard text generation metrics such as BLEU [14] and ROUGE [15], which may not fully capture the nuanced effectiveness of a distractor.

Despite the growing interest in the automatic generation of distractors, many existing approaches often miss an integrated mechanism to ensure that all generated distractors share a semantic relationship while remaining distinct from both the correct answer and from each other. Methods that generate one distractor at a time [9,10,13] (e.g., approaches based on beam-search [16]) may struggle to keep the plausibility or semantic diversity across all distractors. In addition, they frequently rely on ranking or filtering steps, increasing complexity and computation. This opens an opportunity to explore approaches that jointly generate all distractors, allowing the model to capture cross-dependencies and potentially prevent too evident or overly similar distractors. A joint generation can improve semantic relevance, diversity, and overall quality of distractors compared to more traditional, single-output methods.

In response to these challenges, this research focuses on improving the creation of distractors for MCQs in the context of RC datasets. A joint generation of distractors (i.e., all at once) using a text-to-text approach is proposed, by fine-tuning the Flan-T5 [17] and LongT5 [18] models using the RACE dataset [19]. This approach is designed to generate all distractors at once in a single inference step, potentially offering a more cohesive and contextually relevant set of options, as an alternative to more common methods that output a single distractor at a time, as previously mentioned.

For evaluating the generated distractors, in addition to the standard BLEU and ROUGE metrics, an analysis of the semantic distance between the distractors and the correct answers is incorporated, with outputs compared across different models and datasets. In addition to the RACE dataset, the MCTest, SciQ, and OpenBookQA datasets are included in the evaluation framework, enabling the assessment of performance across various contexts that differ from the training data. Additionally, BERTScore [20] is utilized to assess the semantic relevance of distractors in relation to the references. Finally, a grammar check for each distractor is performed, comparing results across datasets and models. This approach helps to narrow the gap in the evaluation of distractor effectiveness.

In summary, the main contributions of this research are (1) Different versions of Flan-T5 and LongT5 models fine-tuned for DG task; (2) An alternative approach to jointly generate distractors using a text-to-text paradigm; (3) Implementation of BERTScore and cosine similarity analysis in the evaluation framework, offering a comprehensive assessment of the semantic proximity and diversity of the generated distractors.

The rest of this paper is organized into five sections. Section 2 reviews current research and models related to the distractor generation task. In Section 3, Materials and Methods, we describe our text-to-text approach for generating distractors jointly and detail our experimental setup. Section 4 then presents our evaluation data and benchmarks, followed by the Discussion (Section 5), where we analyze these findings and explore future work and Limitations (Section 6). The Section 7, Conclusion, summarizes our main contributions.

2 Related Work

2.1 Distractor Generation Approaches

Distractor generation for MCQs has long been a focus in the fields of education and assessment. Recently, there has been interest in using automated methods to create these distractors. Various approaches have been explored, including sequence-to-sequence models [21] along with the application of large Transformer-based language models [22], such as GPT-2 [23], BERT [24], or T5 [25], usually fine-tuned specifically for the DG task.

During the construction of the SciQ dataset [26], the DG problem was addressed more traditionally. The focus was to help crowdsources in selecting the best options from a large set of distractors created using a GloVe vocabulary [27]. To facilitate this, a classifier was trained to rank good candidates based on multiple features, including embeddings, POS-tagging, the distance between the correct answer and candidate distractor, token length, token overlap, and hypernymy/hyponymy indicators. Similarly, for technical domains like engineering, the use of ontologies to generate distractors in MCQs has been suggested [6]. Another study used the T5 model for producing English grammar MCQs [28]. However, distractors were generated based on inputs composed of a keyword and a part-of-speech template and then selected using a rule-based algorithm.

Another study introduced a framework called EDGE (quEstion and answer guided Distractor GEneration) [13]. This approach generates distractors based on the context, the question, and the correct answer, using a sequence-to-sequence model. Two important characteristics of distractors are improved with this model: incorrectness (using a gate mechanism that constrains answer-relevant words based on distance) and plausibility (employing the semantic representation of the question and the context). The study used a modified version of the RACE dataset named DG-RACE [10].

Exploring transformer-based approaches, the DG-RACE dataset was also used to fine-tune a T5 model specifically for the DG task in the context of the end-to-end generation of MCQs [9]. This study proposed a text-to-text approach, which generates a single distractor by leveraging the context, the question, and the correct answer.

The aforementioned studies share a common area for improvement: the output of newly generated distractors is not conditioned on the ones previously generated for the same MCQ. Also, they rely on beam-search methods for regulating the output [9,13].

In addition to T5, alternative Transformer models such as GPT-2 have also been explored. For example, another work approached the DG task by fine-tuning a GPT-2 language model with the RACE dataset to generate three distractors for a given question, correct answer, and context [12]. Following this, an additional step was incorporated, utilizing a DistilBERT model [29] fine-tuned as a classifier. This classifier, also trained using the RACE dataset, had the objective of filtering out MCQs composed of the generated distractors that could be answerable. However, this latter step did not show a meaningful improvement.

2.2 Datasets

Distractors, integral to the structure of MCQs, are typically found in RC datasets, such as MCTest, RACE, OpenBookQA, SciQ, CosmosQA, ARC [30], and CommonsenseQA [31], among others. Consequently, these datasets can be used as rich resources when training models for the DG task due to the inclusion of context for each question-answer pair, along with carefully constructed distractors. Many studies referenced above utilize the context paragraph, the question, and the correct answer as inputs to guide the generation of distractors.

The subject domain across these datasets is varied. Topics in the RACE dataset include narratives, ads, information, and passages across multiple subjects and domains like history, science, and geography, designed with the focus of evaluating the comprehension of texts built from English language exams for middle and high school students [19]. MCTest is an open-domain dataset, mainly composed of fictional narratives that a child could understand [32]. In the case of SciQ and OpenBookQA, topics are specific to the science domain, including biology, physics, chemistry, earth science, and others [26,33]. While the correct answer for a given context and question can be inferred from the passage for RACE, MCTest, and SciQ, OpenBookQA is designed to use multi-step reasoning and common-sense knowledge to answer the questions [33].

In another line of research, a study utilized these RC datasets to train a unified model for question-answering (QA) tasks [34,35]. The model, which implemented answering of MCQs (with and without context paragraphs), used a text-to-text approach based on T5 and BART [36]. This model could be used as a tool to validate automatically generated distractors by incorporating it as an answerability filter, a method proposed by other studies [12,37].

2.3 T5 Model Variants

The inherent flexibility of T5 models and their text-to-text approach for a variety of NLP tasks have led to the development of new T5 model variants that can potentially be used for DG tasks. Among these, LongT5 has exhibited superior performance in processing long sequence inputs and offers a solution to the issue of size scalability often associated with the standard T5 model [18]. Furthermore, a version of LongT5 has been enhanced with an attention mechanism known as Transient Global (TGlobal) attention. This mechanism divides the input sequence into blocks, each one of k tokens. A global token is then calculated based on the summation and normalization of the embeddings associated with the tokens within the block. As a result, the attention mechanism enables the input tokens to attend not only to their immediate neighbors but also to the global token collection [18].

Another significant development is Flan-T5, which represents a T5 model fine-tuned on a larger corpus comprising 473 datasets and 1836 tasks. This comprehensive fine-tuning has enabled Flan-T5 to surpass the performance of published T5 checkpoints, in some instances by a margin exceeding 10% [17].

2.4 Evaluation Challenges

Despite these advancements in model development, some challenges remain in the realm of DG evaluation. Multiple studies have highlighted the absence of standardized metrics and benchmarks for evaluating the quality of distractors [9,12]. Traditionally, these studies have relied on metrics like BLEU and ROUGE, which measure word overlap and are widely used in machine translation tasks. However, distance measures, which have been employed both as features for distractor generation [26] and as supplementary evaluation metrics [9], offer an interesting alternative.

To improve the evaluation of text generation broadly, BERTScore has been introduced. This metric utilizes the contextual embeddings from BERT to calculate a similarity score between reference and generated text, thus providing, in the study, a better correlation with human judgment [20].

2.5 Research Gap and Proposed Approach

Multiple prior approaches to DG exhibit limitations in maintaining semantic coherence and diversity between distractors. Some methods rely on beam search, producing iterative outputs where distractors are generated independently, often leading to additional ranking and filtering steps. Others do not condition generated distractors on those already produced, increasing the possibility of redundant or trivial options. In addition, given the lack of a standard metric for DG, existing methods mostly use word overlap metrics and have not explored the deeper semantic characteristics of distractors, with only a few starting to incorporate cosine similarity.

To address these limitations, a joint distractor generation approach is proposed, utilizing fine-tuned FlanT5 and LongT5 models with a RACE-based dataset. By generating multiple distractors in a single inference step, this method leverages the text-to-text nature of the models to enhance the coherence and semantic relationship of the options. Additionally, BERTScore and cosine similarity are incorporated into the evaluation framework to assess the relevance of the generated distractors.

3 Materials and Methods

The appearance of the Transformer architecture has enabled the emergence of LLMs that revolutionized multiple domains of NLP [38]. These LLMs showcase an outstanding ability to capture linguistic patterns and dependencies at a level not seen before. Among these, the T5 model stands out due to its versatile interface, with a design approach where NLP problems such as summarization, classification, translation, and others, are framed as text-to-text tasks [25]. This characteristic allows the use of these models for multiple applications. T5 models use both encoder and decoder layers from the original Transformer architecture, unlike BERT or the GPT-X model family, which are based only on encoders or decoders, respectively.

LongT5 and Flan-T5 represent an evolution over the original T5 model, and their selection for this study was motivated by the improved efficiency in processing larger inputs and the adaptability to various tasks, respectively. These characteristics are interesting for the domain of MCQs because an efficient multi-task model can be optimized to perform QG, QA, and DG tasks, which are 3 dimensions of the MCQ generation problem [9].

To optimize LLMs for specific tasks, Parameter Efficient Fine-Tuning (PEFT) techniques [39] have recently emerged, such as LoRA, and quantization. These techniques offer ways to fine-tune larger models with fewer computational resources by introducing efficient parameter updates and adaptations, as well as weight precision reduction. In this study, full-finetuning for medium-size models is used, while PEFT is performed for larger models.

The approach of this research for DG uses a text-to-text paradigm, which is natural to T5-like multitask language models, as mentioned before. The context, question, and correct answer are used as input for the model while the set of distractors is the expected output (Fig. 1). The question is added at the beginning of the input, followed by the correct answer and the context paragraph, which are separated by the labels "CORRECT-ANSWER:" and "CONTEXT:", respectively. The output text is structured as a lettered list of distractors. To avoid ambiguity during fine-tuning (especially for Flan-T5), the task is prefixed with the label "GENERATE-DISTRACTORS:".

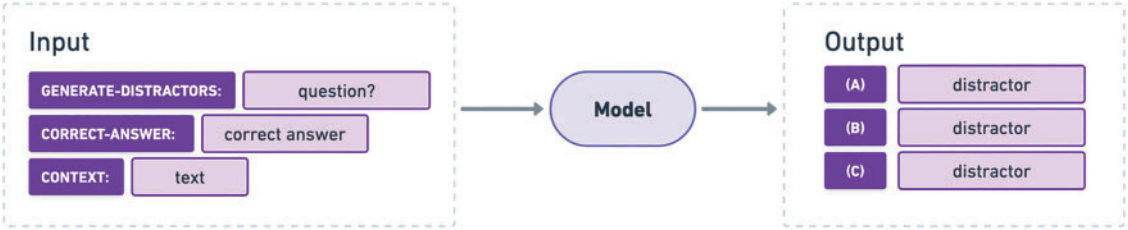

Figure 1: Text-to-Text approach for the joint generation of distractors

Both the Flan-T5 and LongT5 models are fine-tuned using the specified input and output structures. The maximum input length for these models is set to 1024 tokens and training examples that exceed this limit are excluded from the training set. The ordering of elements in the formatted input is intentionally designed to retain most of the pertinent information, even when inputs are larger than the limit during inferences. This ensures that the question and correct answer are always included, with only the less critical portions of the context potentially being truncated.

The training and evaluation processes are illustrated in Fig. 2. The training process begins by formatting the train and validation splits of the RACE dataset, according to the format presented in Fig. 1. Each example is then tokenized using the model-specific tokenizer (both Flan-T5 and LongT5 have their own tokenizers). Next, the process diverges based on the fine-tuning technique. For models undergoing full fine-tuning, the approach is straightforward: the pre-trained public model is fetched, and the training is executed using the Seq2SeqTrainer from the Transformers library in Python [40]. In the scenario of larger models, the methodology adopted is slightly different. Upon retrieving the pre-trained model, a LoRA adapter model, which is significantly smaller than the full model, is prepared. This adapter model will go through the fine-tuning process, updating only its parameters instead of the whole pre-trained model, which, in this case, is loaded to GPU memory using 4-bit quantization. This process effectively reduces memory demands during training. Following this, the training proceeds similarly to the full fine-tuning approach, utilizing a Seq2SeqTrainer from the Transformers library. However, in this instance, the output is an adapter model optimized for the DG task, which must be merged with the original pre-trained model. The output of this entire process is a model fine-tuned for the DG task.

For the evaluation, the test splits from the RACE, MCTest, SciQ, and OpenBookQA datasets are formatted according to the structure proposed in Fig. 1. The inputs undergo tokenization before using the models to generate distractors. The performance of each fine-tuned model is assessed using BLEU and ROUGE metrics, in addition to calculating the BERTScore. To further evaluate and understand the behavior of the generated distractors, an analysis of their grammatical correctness, as well as their distances with the correct answers is conducted.

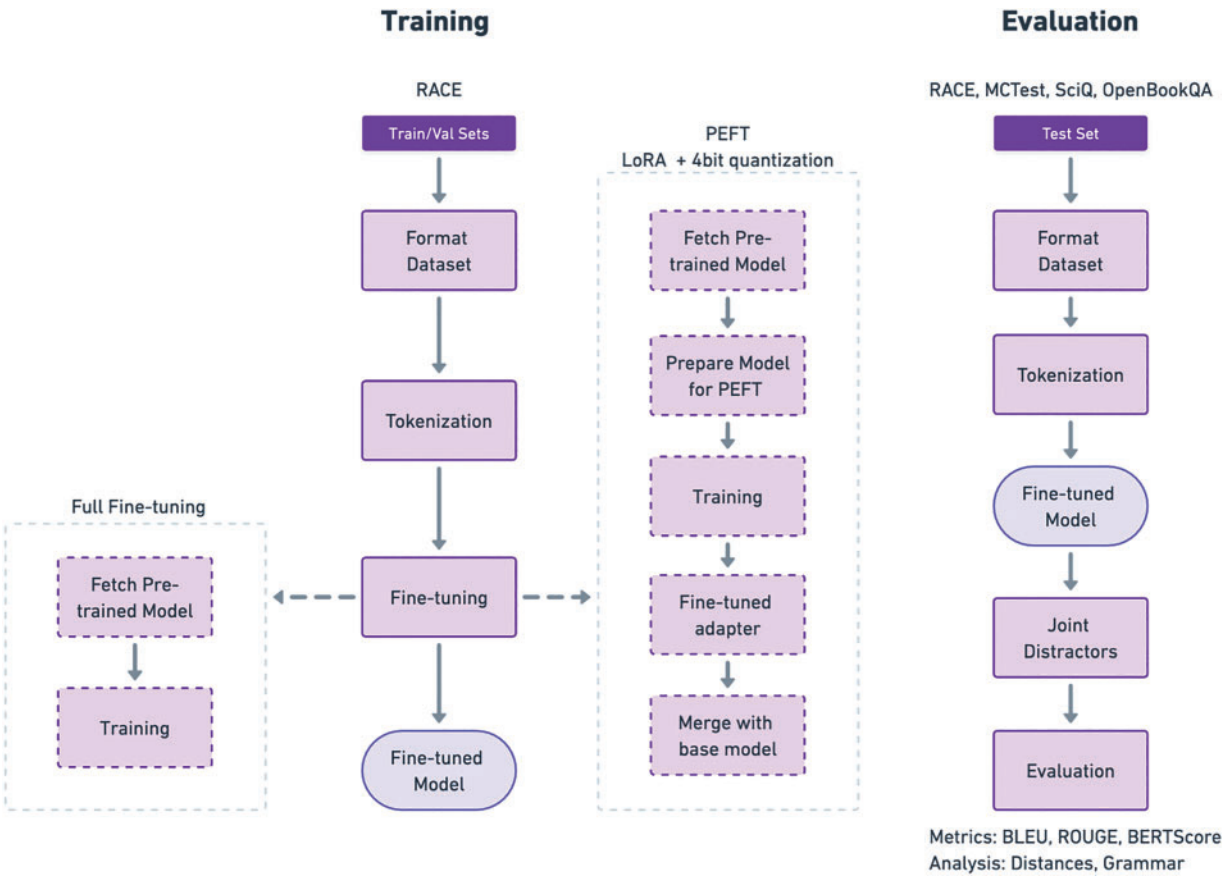

Figure 2: Training and evaluation process. Boxes with dashed lines showcase specific steps for the case of full fine-tuning and LoRA + quantization fine-tuning

3.1 Experimental Setup

3.1.1 Datasets

As previously noted, datasets featuring MCQ structures composed of context, question, correct answer, and distractors are well-suited for the DG task. In the scope of this study, the RACE dataset, collected from English examinations for both middle and high school students, was pre-processed and used for model fine-tuning. Table 1 shows the number of examples in the dataset. Each example in the RACE dataset is composed of an article (the context of the question), the question stem, a set of 4 options (including the correct one), and the answer (the correct option identified by a letter from A to D). The letter of the answer is used to identify the correct answer in the set of 4 options.

All dataset examples were transformed into the input-output format previously described in Fig. 1. Specifically, each input is structured as follows: *"GENERATE-DISTRACTORS: <question stem>\nCORRECT-ANSWER: <correct answer text>\nCONTEXT: <context/article>"*. Each output is a lettered list of distractors: *"(A) <distractor 1>\n(B) <distractor 2>\n(C) <distractor 3>"*. These distractors are composed of the options available in the original MCQ dataset but removing the correct one. A training example is shown in Fig. 3, where the *Formatted Input* block is used to feed the model, and the *Expected Output* block is the target to match.

Table 1: Number of formatted text-to-text examples per dataset, including train, validation, and test splits for RACE, and only the test splits for SciQ, MCTest (mc500), and OpenBookQA

Dataset (Split)	Examples
RACE (Train)	87,560
RACE (Val)	4867
RACE (Test)	4934
SciQ (Test)	1000
MCTest-mc500 (Test)	600
OpenBookQA (Test)	500

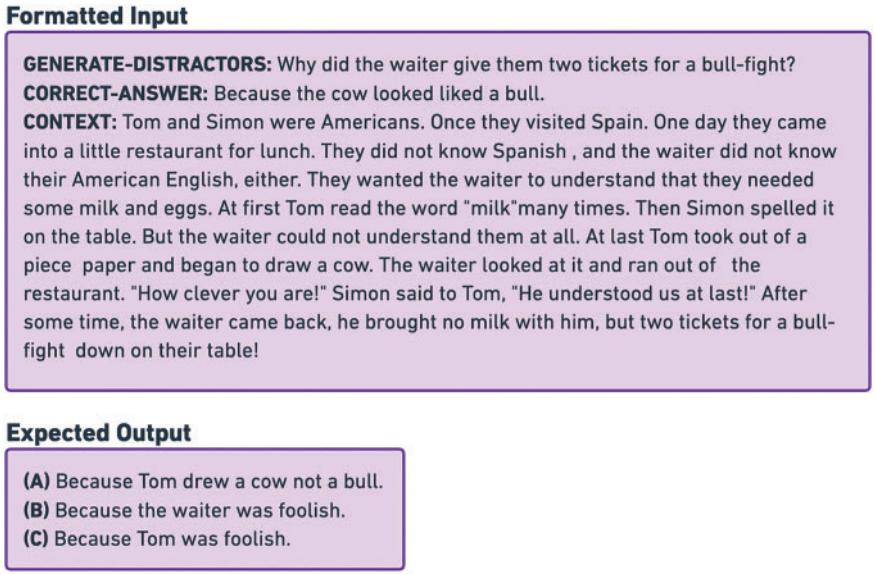

Figure 3: Formatted training example based on the RACE dataset for the distractor generation task

Once the input and output were structured accordingly, a tokenization process was applied to enforce the 1024-token input limit. As mentioned before, examples exceeding this limit were excluded, reducing the training samples for the train split of the RACE dataset from 87,866 to 87,560. This filtering procedure was consistently applied to all splits and additional datasets.

The test split was used to generate distractors for unseen inputs and then compared with baseline models. Additional evaluation was performed on test splits from the MCTest, SciQ, and OpenBookQA datasets, enabling the assessment of performance across various contexts that differ from the training dataset (Table 1).

3.1.2 Fine-Tuned Models

For the purpose of this study, three versions of pre-trained Flan-T5 and LongT5-TGlobal models were fine-tuned: Base (250 M parameters), Large (780 M parameters), and XL (3 billion parameters). The Base and Large models underwent a full fine-tuning process on an RTX A6000 GPU, with a cost of $1.89/h. Due to memory limitations and the availability of GPUs, the XL models were fine-tuned on a Nvidia A10 G Tensor Core GPU ($1.21/h), utilizing Low-Rank Adaptation (LoRA) [41] and 4-bit quantization techniques [42]. To

establish a reference for models employing LoRA and quantization, a LongT5 Base model was also fine-tuned using these methods.

In the fine-tuning process for all models, the max_source_length was set to 1024 tokens and the max_target_length to 256. These parameters were selected based on the distribution of text lengths in the RACE dataset, which has the longest inputs and outputs among the datasets used in the study. By using 1024 tokens for max_source_length, the training process was able to include 99.6% of the training examples using the proposed input format (Fig. 1) without truncation, while keeping the memory requirements within the GPU and resource availability constraints. A max_target_length of 256 tokens provides enough output length for generating distractors in the proposed format for the studied datasets.

3.1.3 Baseline Models

To benchmark the performance of fine-tuned Flan-T5 and LongT5 models, BLEU and ROUGE results were compared with those reported by models from three separate studies: GPT-2 + DistilBERT [12], T5-DG [9], and a Seq-to-Seq model [10]. In these studies, models were fine-tuned on the DG task using the RACE dataset, requiring the context, question, and correct answer as input, similar to the present study. In the case of GPT-2 + DistilBERT, the reported model sizes were 355 M and 66 M parameters, respectively. The T5-DG model was based on T5-Small (60 M parameters) and the Seq-to-Seq model was based on a custom long short-term memory (LSTM) network that used GloVE as embeddings (840 B.300 d version). More details on the fine-tuning process of these models can be found in their respective studies.

Furthermore, for a comprehensive baseline across various datasets, distractors were generated for the test splits of RACE, MCTest, SciQ, and OpenBookQA using GPT-3.5-turbo-1106 via the OpenAI API, incurring a total cost of approximately $3.05. A system prompt indicated the details of the DG task and the expected output format, while a user prompt was utilized to input the question, correct answer, and context, as illustrated in Fig. 4. The produced distractors and the corresponding metrics obtained serve as a reference for the DG task applied to all datasets evaluated in the research.

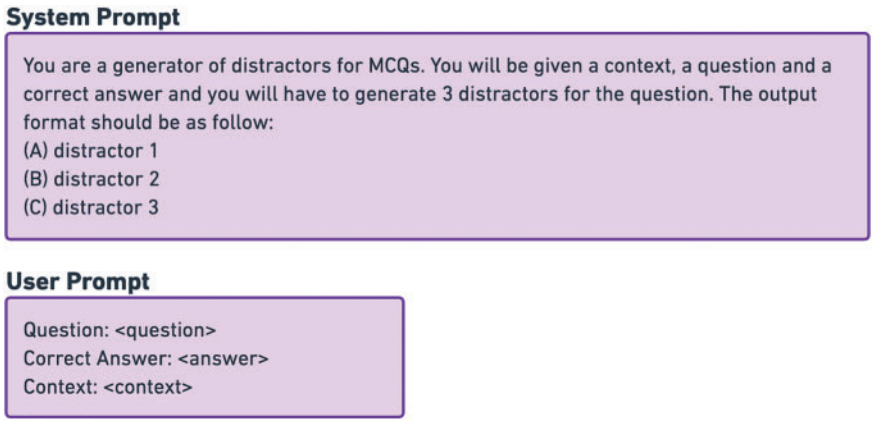

Figure 4: Prompts for generating distractors with GPT-3.5 and OpenAI API

3.1.4 Automatic Evaluation

To assess the quality of the generated distractors, three metrics were employed: BLEU (ranging from 1 to 4 n-grams) [14], ROUGE-L [15], and BERTScore [20]. These evaluation metrics were compared against the baseline models mentioned before.

While BLEU and ROUGE measure n-gram or subsequence token overlap, they can fail to capture semantic differences (or similarities). This is especially interesting when analyzing distractors because they should be contextually plausible but distinct from the correct answer. BERTScore leverages contextual embeddings from BERT-based models, offering a better approximation of semantic similarity, with improved correlation to human judgments [20,43] compared to traditional token overlap metrics.

An additional distance analysis was conducted on pairs consisting of a correct answer and its corresponding distractors, as shown in Fig. 5. These pairs were extracted from the test splits of all the datasets used in this study. Cosine similarity calculations were performed using embeddings from Sentence-BERT (all-MiniLM-L6-v2 version) [44]. The resulting similarity scores from the test splits served as a reference for comparison against the similarity scores obtained from distractors generated by both the fine-tuned models and the GPT-3.5 baseline.

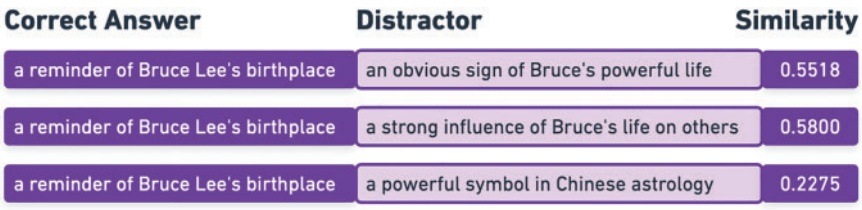

Figure 5: Example of correct-answer and distractor pairs extracted from the same question, with the respective cosine similarity measure

Both BERTScore and cosine similarity can offer opportunities for analyzing distractors. However, it is important to note that both metrics can overestimate similarity for pairs sharing common tokens or fail to capture small changes like negations [45]. The usage of contextual embeddings should help to mitigate this effect in similarities; however, these metrics are incorporated to provide an additional perspective to BLEU and ROUGE-L.

Finally, an analysis of the grammatical correctness of the generated distractors was performed. This is particularly relevant for the RACE dataset, where distractors are typically composed of multiple words. The analysis was based on LanguageTool[1], an open-source grammar checker that shows high accuracy and exhibits a significant correlation with human ratings [46]. For this evaluation, the original distractors from the test split of all datasets, as well as those generated by the fine-tuned models and the GPT-3.5 baseline, are used. Specifically, the percentage of distractors that contain errors in the "GRAMMAR" category is computed. This category covers issues related to verb usage, pluralization, tense, nouns, and more [46]. It is important to note that this evaluation focuses primarily on structural errors rather than on spelling, capitalization, punctuation, or whitespace issues.

[1] https://github.com/languagetool-org/languagetool (accessed on 1 January 2025).

4 Results

Table 2 presents the BLEU, ROUGE-L, and BERTScore F1 metrics for the models trained in this study, evaluated on the test split of the RACE dataset, alongside comparisons with baseline models. The table also includes the percentage of inferences that resulted in duplicated distractors, a phenomenon observed during the analysis of the generated outputs. BLEU (B1–B4) and ROUGE-L (R-L) scores range from 0 to 100, with higher values indicating greater n-gram and longest common subsequence overlap, respectively. BERTScore F1 (BS-F1) also ranges from 0 to 100, reflecting semantic similarity to the reference, where higher scores are better. The percentage of duplicated distractors (%DUP) ranges from 0 to 100, with lower values being preferable.

Table 2: Automatic evaluation results for DG on the RACE dataset. BLEU scores in columns B1 to B4, ROUGE-L in column R-L, BERT-Score F1 in column BS-F1. Percentage of instances with duplicated distractors in column %DUP. Results from GPT-2 + DistilBERT, T5-DG, and Seq-to-Seq were sourced directly from their published studies. Bolded values indicate the highest performance for that metric

Model	B1	B2	B3	B4	R-L	BS-F1	%DUP
FlanT5-Base	52.53	35.73	23.98	12.42	37.32	90.18	10.9
FlanT5-Large	54.05	37.2	25.24	**13.53**	**38.55**	90.53	3.95
FlanT5-Base (LoRA)	40.82	26.21	15.69	4.69	23.25	87.52	0.79
FlanT5-XL (LoRA)	55.62	37.78	24.77	11.58	35.18	90.19	1.34
LongT5-Base	51.74	35.05	23.62	12.41	37.38	90.2	7.36
LongT5-Large	52.83	35.21	23.12	11.18	35.51	89.98	6.97
LongT5-XL (LoRA)	55.94	**38.2**	**25.39**	12.59	36.6	90.6	0.85
GPT-3.5	46.98	29.65	18.04	6.85	32.01	**91.43**	**0**
GPT-2 + DistilBERT	**60.12**	26.56	13.64	9.17	12.36	–	–
T5-DG	14.80	7.06	3.75	2.16	14.91	–	–
Seq-to-Seq	26.93	13.57	8.0	5.21	14.54	–	–

It is visible that FlanT5 and LongT5 models, particularly in their Large (780 M parameters) and XL (3 B parameters) versions, achieve strong performance across BLEU scores, indicating an improvement in word overlap with reference texts, especially from 2 to 4 n-grams. The ROUGE-L scores, which focus on the longest common sequence between the generated text and the reference, are also robust for the fine-tuned models, with all fine-tuned versions improving the performance of the baselines, except by FlanT5-Base fine-tuned with LoRA.

Regarding BERTScore F1 metrics, values gravitate around 0.9, suggesting that the generated distractors are semantically close to the reference. However, GPT-3.5 scores slightly higher than all other models. This might suggest that despite lower n-gram overlap, distractors generated by GPT-3.5 might be semantically closer to the references.

As mentioned above, duplication of distractors in model outputs was observed. This can be an indicator of the inability of the models to generate diverse distractors. Notably, models fine-tuned with LoRA and quantization exhibit the lowest rates of duplication when compared to fully fine-tuned versions. This can be evidenced by the more than 10% reduction in duplication when comparing the fully fine-tuned and LoRA fine-tuned FlanT5-Base models. Nevertheless, GPT-3.5 generated zero duplicated distractors for all the inferences.

Another observation from the results data is that the TGlobal attention mechanism in the LongT5 models does not demonstrate a significant performance advantage over the FlanT5 models. In fact, the FlanT5-Large model clearly outperforms its LongT5 counterpart.

Table 3 reinforces the observed phenomenon of duplication across other RC datasets, indicating that in the presence of other contexts, there is a higher tendency to duplicate one distractor, especially by smaller models. Regarding BERTScore, values still gravitate around 0.89 and 0.9, suggesting that the generated distractors are semantically close to the reference. However, they still underperform in this metric when compared to the GPT-3.5 baseline.

Table 3: Automatic evaluation results for other popular RC datasets. BLEU scores in columns B1 to B4, ROUGE-L in column R-L, and BERT-Score F1 in column BS-F1. Percentage of instances with 1 duplicated distractor in column %DUP1 and with 2 duplicated distractors in %DUP2

Dataset	B1	B2	B3	B4	R-L	BS-F1	%DUP1	%DUP2
FlanT5-Base								
MCTest	66.19	50.96	35.15	18.93	50.77	91.6	20.33	4
SciQ	69.83	51.16	29.99	5.3	48.3	89.23	30.6	25.8
OpenBookQA	60.67	42.99	26.2	8.05	42.15	89.34	29.2	16.6
FlanT5-Large								
MCTest	68.97	53.97	37.4	20.45	51.15	92.08	3.67	0.17
SciQ	69.55	51.3	30.09	5.82	48.06	89.62	18.8	6.7
OpenBookQA	60.38	43.16	26.23	7.91	41.86	89.6	19.2	2.6
FlanT5-Base (LoRA)								
MCTest	45.36	31.22	18.54	4.78	29.7	88.18	0.33	0
SciQ	27.89	18.59	9.71	0.23	33.42	87.31	1.8	0.7
OpenBookQA	21.26	13.72	7.28	0.59	29.26	86.71	1	0.2
FlanT5-XL (LoRA)								
MCTest	63.27	46.9	30.54	13.25	43.69	91.08	0.17	0
SciQ	65.18	47.81	27.53	4.71	46.41	90.66	1.1	0.2
OpenBookQA	60.59	43.15	25.77	6.58	40.6	90.02	1.4	0
LongT5-Base								
MCTest	66.6	51.72	35.76	19.43	50.84	91.76	9.17	2.17
SciQ	68.08	49.57	28.89	5.17	47.81	89.29	25.9	16.3
OpenBookQA	59.05	41.81	25.42	7.64	41.33	89.11	24.6	14.6
LongT5-Large								
MCTest	61.76	45.03	29.93	13.91	45	90.88	9.33	2.17
SciQ	61.97	44.24	24.74	2.35	42.11	88.61	26.5	4.8
OpenBookQA	55.37	38.27	22.12	4.73	37.55	88.96	14.6	1
LongT5-XL (LoRA)								
MCTest	70.09	54.23	36.89	18.73	49.29	92.04	0	0
SciQ	48.92	34.35	18.5	1.45	44.31	89.54	0.3	0
OpenBookQA	45.41	30.7	16.5	1.82	36.07	88.61	1.2	0

(Continued)

Table 3 (continued)

Dataset	B1	B2	B3	B4	R-L	BS-F1	%DUP1	%DUP2
				GPT-3.5				
MCTest	58.12	41.44	25.71	9.83	43.42	93.41	0	0
SciQ	72.7	54.85	32.17	6.88	49.68	94.24	0	0
OpenBookQA	57.29	40.25	23.67	6.23	40.18	92.69	0	0

When looking at the XL models, LongT5-XL (LoRA) excels at MCTest, showing no duplication and outperforming GPT-3.5 in BLEU and ROUGE-L metrics. However, it is still behind on BERTScore. FlanT5-XL (LoRA) notably outperforms its LongT5 counterpart in SciQ and OpenBookQA. Additionally, it outperforms GPT-3.5 in BLEU and ROUGE-L metrics for OpenBookQA, while showing relatively low duplication. GPT-3.5 clearly outperformed in BERTScore and generated no duplicated distractors.

4.1 Distance Analysis

The cosine similarity between the distractors and correct answers, measured for the test splits in the four evaluated datasets, showed a median value of around 0.4, with the interquartile range (IQR) falling between 0.2 and 0.6. However, there were large whiskers and outliers present in the box plots, indicating that the measures were not distributed evenly (Fig. 6). It is visible that OpenBookQA behaves differently from the rest, with a range and median slightly lower than the other datasets.

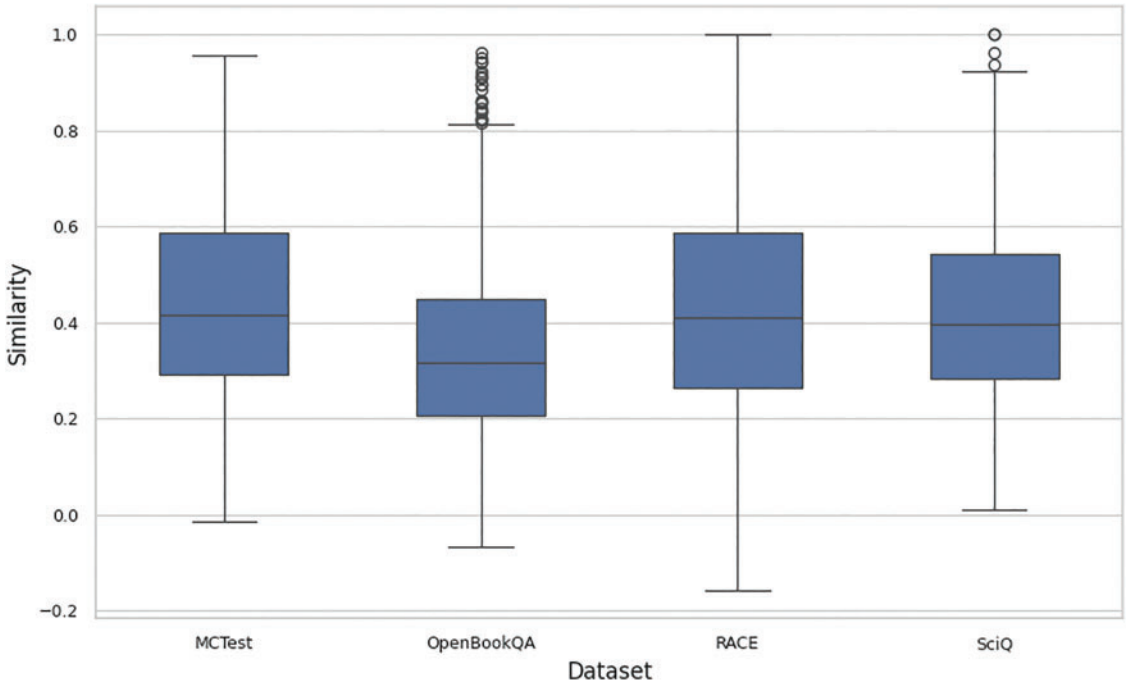

Figure 6: Ranges of distance measures (cosine similarity) between distractors and correct answers for the test splits from the MCTest, OpenBookQA, RACE, and SciQ datasets

A distance analysis based on the distractors generated by the fine-tuned models (Fig. 7) shows a similar spread when compared to the reference and tends to have a higher cosine similarity median than the reference. FlanT5-XL (LoRA) closely approximates the behavior of distance ranges observed in the reference and GPT-3.5. Additionally, the lower performance of FlanT5-Base (LoRA), as observed in Table 2, becomes more evident in Fig. 7, exhibiting a similarity median below 0.2.

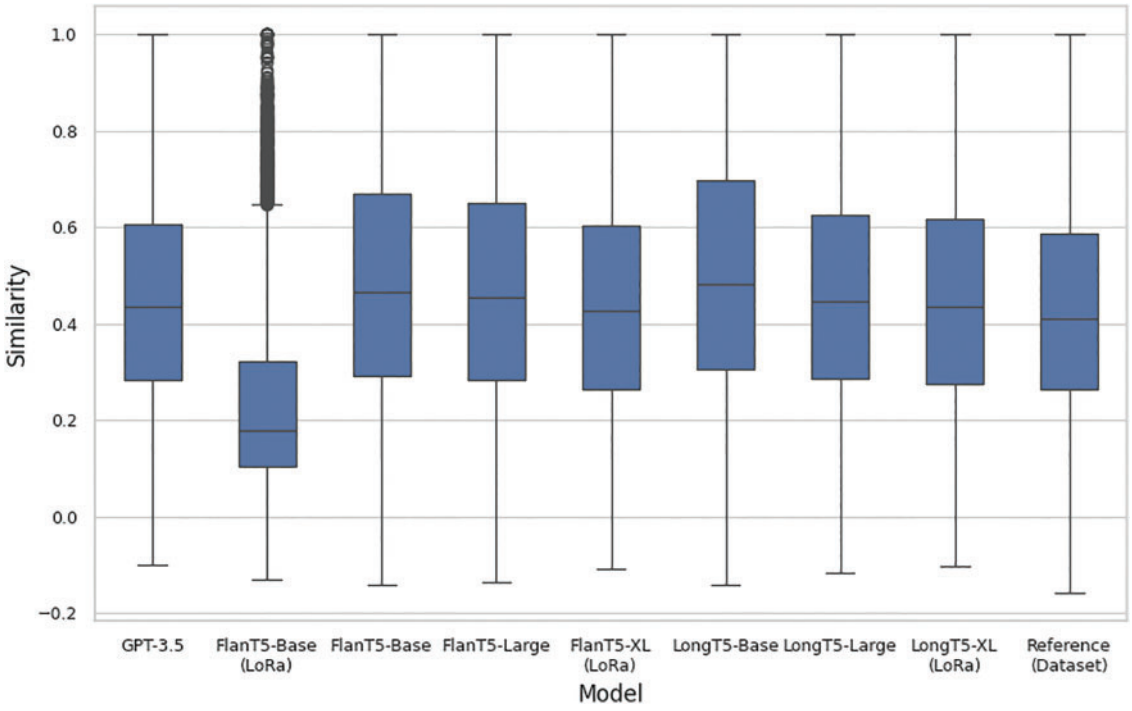

Figure 7: Ranges of distance measures (cosine similarity) between the correct answer and distractors generated by fine-tuned models for the test split from the RACE dataset, compared against the reference and GPT-3.5

A similar comparison for MCTest, SciQ, and OpenBookQA is shown in Fig. 8. It is evident that smaller models struggled to approximate the distances between the distractors and the correct answer across datasets. In particular, the Base versions with full fine-tuning, which also showed a high duplication rate, present a larger IQR, especially in the SciQ dataset. An upper quartile extending to 1 (the maximum similarity) indicates that these models are generating distractors that, in some form, are paraphrasing or synonymous with the correct answer. An example of this is shown in Table 4 for FlanT5-Base, where the generated distractors include the correct answer "nervous system" as a part of them. When compared to the reference and LongT5-XL (LoRA), the cosine similarity of the examples for FlanT5-Base is considerably higher. A similar phenomenon is also observed in OpenBookQA, as shown in Fig. 8.

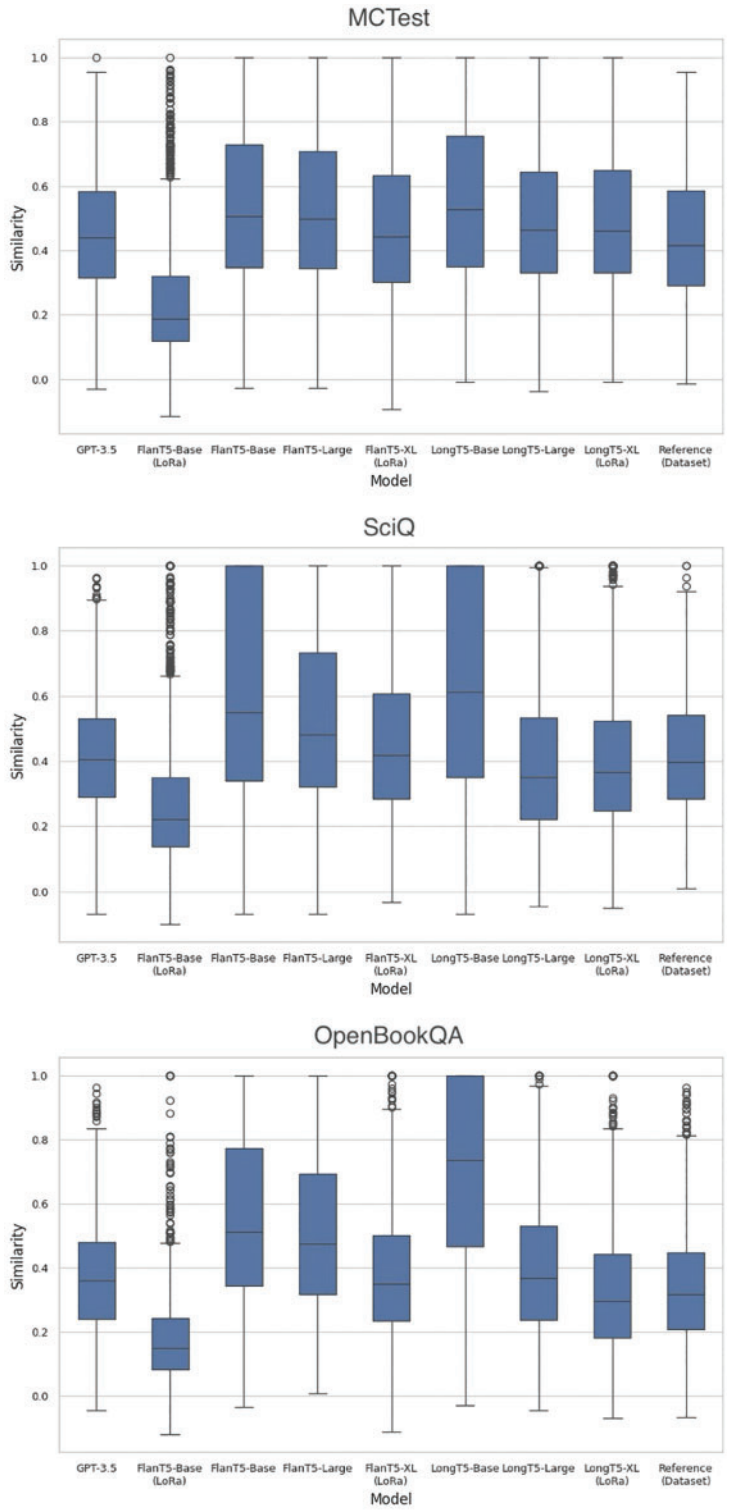

Figure 8: Ranges of distance measures (cosine similarity) between the correct answer and distractors generated by fine-tuned models for the test split from MCTest, SciQ, and OpenBookQA datasets, compared with reference and GPT-3.5

Table 4: Example of distractors generated by FlanT5-Base that include the correct answer as a part of the options and show high cosine similarity (CS) compared to distractors generated by a larger model and the SciQ reference

Model or reference	Correct answer	Distractor	CS
FlanT5-Base	Nervous system	The *nervous system*	0.915
FlanT5-Base	Nervous system	The *nervous system* of the brain	0.870
FlanT5-Base	Nervous system	The *nervous system* of the body	0.891
LongT5-XL (LoRA)	Nervous system	Respiratory system	0.490
LongT5-XL (LoRA)	Nervous system	Digestive system	0.431
LongT5-XL (LoRA)	Nervous system	Circulatory system	0.496
SciQ Reference	Nervous system	Cardiovascular system	0.542
SciQ Reference	Nervous system	Circulatory system	0.496
SciQ Reference	Nervous system	Central system	0.440

4.2 Grammatical Correctness

Overall, the larger models (XL versions) fine-tuned using LoRA and quantization tend to have a lower rate of distractors with grammatical issues (less than 0.25%), across all fine-tuned models (Fig. 9). However, it is worth noting that FlanT5-Base (LoRA) exhibits an even lower percentage of distractors with grammatical errors for the RACE dataset but is one of the higher for SciQ. In general, FlanT5 models show fewer grammatical problems across all datasets when compared to LongT5, with LongT5-Large being the worst performer in the grammar analysis. These results could be due to the nature of the pre-trained FlanT5, which has been fine-tuned on several other datasets and tasks.

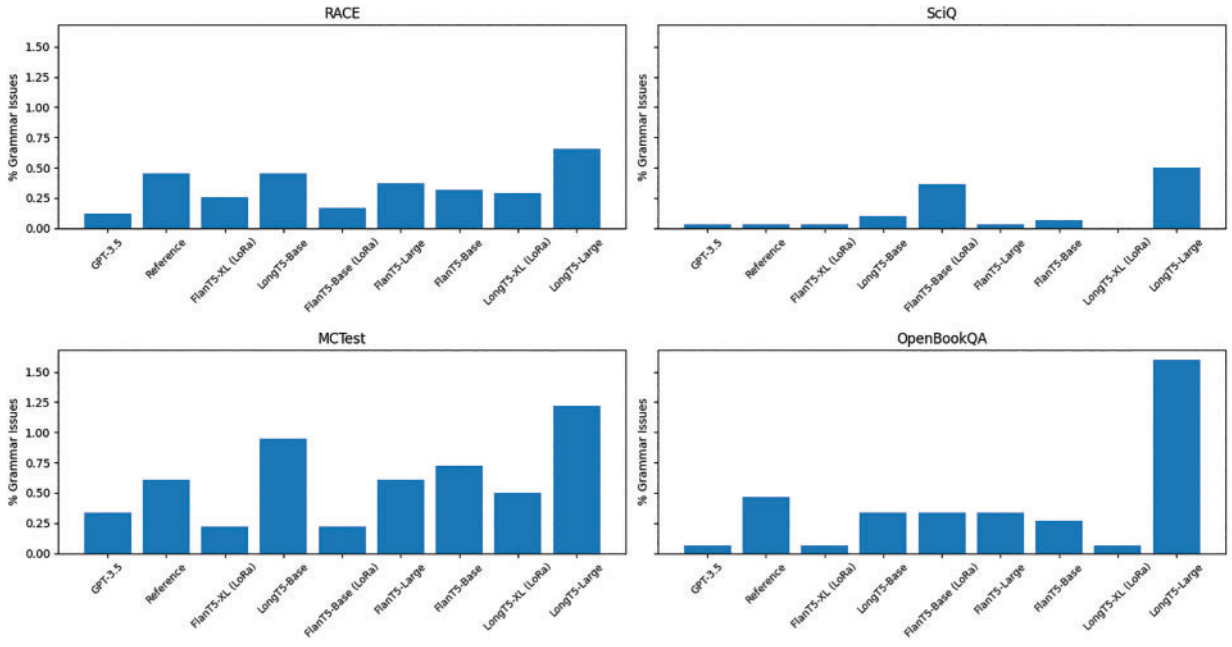

Figure 9: Percentage of distractors with grammar issues, generated by the fine-tuned models for the test split from RACE, MCTest, SciQ, and OpenBookQA datasets, compared with reference and GPT-3.5

The GPT-3.5 baseline has consistently the lowest percentage of distractors with grammatical errors, except for MCTest. Interestingly, the references from each dataset tend to be on the higher end of the percentage of errors (although still very low, around 0.50% for RACE, MCTest, and OpenBookQA). This could be caused by the fact that these datasets were built using a mix of crowdsourcing and semi-automated techniques.

5 Discussion

The results demonstrate that the proposed approach can outperform baseline models in the DG task, evidenced by the improvements in BLEU and ROUGE-L metrics in Table 2. When compared to previous works [9,10,12], the fine-tuned models in this study, which employ a text-to-text format to generate all distractors in a single inference jointly, improved BLEU-2 to 38.2, BLEU-3 to 24.39, BLEU-4 to 13.53 and ROUGE-L to 38.55 for the RACE dataset. These results showcase an improvement in the overlap between the generated distractors and the reference, especially for multiple n-grams. It is worth noting that one of the reference models (GPT-2 + DistilBERT) still shows a higher BLEU-1 score. However, the relevance of 1-g metrics for distractors in the RACE dataset is limited, given that most distractors are composed of multiple-word tokens. The baseline models, T5-DG and Seq-to-Seq, which generate a single distractor per inference without any ranking, exhibit significantly lower performance than the fine-tuned models, including the smaller ones.

Further analysis is needed to understand the extent to which some distractors generated by this approach may represent variations of the correct answer, for instance, through the use of synonyms or paraphrasing. Another recent study also observed this phenomenon, focused on generating MCQs related to programming [47]. This study used GPT-4, and in some instances, all distractors generated for a question were valid correct answers. The presented analysis of cosine similarity metrics between the correct answer and distractors can provide insights into this tendency, which is indicated by an increase in the median and upper quartile of cosine similarity scores.

It is also worth noting that the distractors contained in the datasets do not necessarily originate from the text or refer to it. Sometimes, options are plausible for a human reader but may not directly relate to the context, making them challenging to model. This is why conventional token overlapping metrics like BLEU or ROUGE do not reflect an accurate quality measure for distractors. The inclusion of BERTScore provides insights into the semantic proximity to the reference text. However, unlike this study, the metrics reported by baseline models of other works do not include this score, opening opportunities for future research in this area.

It is also important to mention that the BERTScore for GPT-3.5 outputs was consistently higher across all datasets. This is particularly evident for MCTest, SciQ, and OpenBookQA. The best-performing fine-tuned models in this study achieved BERTScores of 92.08, 90.66, and 90.02, respectively. In contrast, GPT-3.5 scored 93.41, 94.24, and 92.69, respectively, indicating that the fine-tuned models still fall behind in terms of semantic proximity of the distractors, when compared to a LLM like GPT-3.5. One potential explanation for this is the GPT-3.5 model size and the vast amount of data used for its training, allowing it to model semantic relationships better. Using larger T5 variants, like XXL, and increasing the diversity of the datasets used for fine-tuning them (not only RACE), could potentially improve this metric.

The observed tendency to duplicate distractors was not fully removed. Models fine-tuned using LoRA exhibited higher distractor diversity, evidenced by a significantly lower rate of distractor duplication. The precise reasons for this require further investigation, and future work could explore them. However, it could be due to the nature of LoRA, where only a small set of parameters is fine-tuned, and the majority remains

unchanged. This leads to more efficient learning of patterns necessary for generating diverse distractors and prevents overfitting, consequently leading to better generalization.

Given that the datasets used in the experiments were mostly composed of questions with 4 options (1 correct answer + 3 distractors), the flexibility to control the resulting number of distractors for each question is limited. This could be addressed by enriching the training dataset with a variable number of distractors per question (for example, generated by GPT-3.5) and adjusting the prefix of the DG task to specify the number of distractors to generate.

Also, the semantic proximity of distractors with the correct answer (and between them) cannot be controlled. The datasets utilized consist of MCQs with a single correct answer option. Therefore, further research is required to investigate the performance of the proposed approach when dealing with correct answers comprising multiple options.

Fine-tuned models were capable of generating distractors that are grammatically correct, sometimes matching the level of GPT-3.5 and even surpassing the reference. However, a deeper analysis and future research are needed to automatically evaluate their effectiveness and quality, including considerations such as the length of the distractor compared to the correct answer, the plausibility of the generated options, grammatical concordance with the question, and linguistic complexity, among other recommendations for writing effective multiple-choice items [4]. In addition, human evaluation could offer an opportunity to further assess the quality of generated distractors using the proposed method and explore to what extent these distractors can confuse examinees. Due to the constraints of the current study, this was outside of the scope, presenting an opportunity for future work.

When comparing the fine-tuned XL versions, FlanT5 outperformed LongT5 on SciQ and OpenBookQA. These datasets have significantly shorter inputs and distractors compared to RACE and MCTest, where LongT5 with the TGlobal attention mechanism exhibited better performance. However, for smaller models, FlanT5 outperformed LongT5 most of the time across all datasets. As a consequence, further research is needed to understand the impact and effectiveness of the TGlobal attention mechanism, particularly for the DG task.

LLMs typically require extensive fine-tuning and significant computational resources, even when using LoRA and quantization to reduce memory usage. Most of the cost comes from training, which can take many hours (XL models took between 60–65 h on a single GPU). However, the cost difference compared to API-based solutions like GPT-3.5 is smaller during inference. For instance, the fine-tuned Large models can generate distractors for all test splits for about $1.89 (1 h compute time), compared to $3.05 from GPT-3.5-Turbo.

Overall, the findings of this study demonstrate the potential of using a text-to-text approach for the joint generation of distractors for MCQs. Nevertheless, more comprehensive research is required to fully understand its limitations and potential and investigate alternate datasets, architectures, and methodologies for distractor generation via large language models.

6 Limitations

Due to GPU resource limitations, this study only fine-tuned models up to 3 billion parameters (XL versions), and it was not possible to fine-tune the larger models with 11 billion parameters (XXL versions).

The evaluation of distractor quality in this study relies mainly on automatic metrics, which do not capture the impact of the distractors on learning outcomes. Human evaluation, case studies, and psychometric analyses for examining item difficulty and discrimination are recommended to validate the educational effectiveness of the generated distractors and their applicability in educational settings.

The method was based on the RACE dataset, and the evaluation included other RC datasets with different domains, such as science and common knowledge. However, its generalizability to any other domain, question type, or language besides English remains unconfirmed.

Finally, while metrics like BERTScore provide useful perspectives for the DG task, they do not model all the characteristics of a good distractor. In fact, they can overestimate similarity for cases like negations.

7 Conclusion

This study presents a text-to-text approach for the joint generation (i.e., all at once) of distractors and evaluates its potential by fine-tuning FlanT5 models and LongT5 with TGlobal attention models using a RACE-based dataset. Both Base and Large model variants are fully fine-tuned, while XL variants are fine-tuned using LoRA and 4-bit quantization. Compared to previous works, the proposed method and models demonstrate an improvement in the standard metrics, BLEU and ROUGE-L, for distractors generated for the RACE dataset. They also show better performance on the same metrics than a baseline generated in this study using GPT-3.5. The fine-tuned models have been published and made available on the Huggingface platform (Appendix A).

An additional evaluation is performed by generating distractors for other MCQ datasets (MCTest, SciQ, and OpenBookQA). The FlanT5-XL model fine-tuned with LoRA outperformed its LongT5 counterpart on SciQ and OpenBookQA, but LongT5-XL performed better on MCTest and RACE. In the case of smaller models, FlanT5 typically outperformed LongT5 across all datasets.

This study introduces BERTScore as an additional metric in the evaluation framework for DG, given that research suggests token overlapping metrics like BLEU and ROUGE do not fully measure the quality of distractors. BERTScore results show that models fine-tuned using the proposed approach generate distractors that are semantically close to the reference. However, despite underperforming in BLEU and ROUGE, the GPT-3.5 baseline still scored better on this metric.

The presented approach generates multiple distractors per model inference, taking into consideration the relationship of all distractors with the context, question, and correct answer. This leads to better performance when compared to generating a single distractor per inference. Additionally, this method generates sets of grammatically correct distractors that can approximate the range of semantic distances with the correct answer observed in the references, especially those generated by the XL models. A tendency toward the repetition of distractors has been observed, with models fine-tuned using LoRA exhibiting a considerably lower rate of duplicated distractors when compared to fully fine-tuned models. Additional research is needed to fully understand how LoRA fine-tuning leads to better diversity with the proposed approach for DG.

Future work can explore how elements like distractor length, option plausibility, grammatical consistency, and linguistic complexity, alongside the relationship of distances between distractors, correct answers, and the context, could help develop better metrics to automatically assess the quality of distractors generated for MCQs. Although the ability of the generated distractors to confuse examinees is not analyzed, human evaluation offers an opportunity for future studies. Furthermore, this study suggests how the proposed text-to-text approach can be improved by enriching the training dataset and adjusting the task prefix to control the number of distractors generated in a single inference. Lastly, distractors generated using GPT-3.5-turbo-1106 for the test splits of RACE, MCTest, SciQ, and OpenBookQA datasets have been made available (Appendix B). These distractors can be used by other studies as baselines for comparing performance in future works.

Acknowledgement: This work was partially co-funded by the Comunidad de Madrid (Grant number: CM/JIN/2021-034) and the University of Alcala (Grant number: PIUAH21/IA-010 and PIUAH23/IA-007).

Funding Statement: This work was supported by the Universidad de Alcalá (UAH) under Grant PIUAH21/IA-010; and Comunidad Autonóma de Madrid under Grant CM/JIN/2021-034.

Author Contributions: The authors confirm their contribution to the paper as follows: Ricardo Rodriguez-Torrealba: Conceptualization, Investigation, Methodology, Software, Visualization, Writing—review & editing. Eva Garcia-Lopez: Conceptualization, Investigation, Methodology, Supervision, Writing—review & editing. Antonio Garcia-Cabot: Funding acquisition, Investigation, Methodology, Supervision, Writing—review & editing. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the corresponding author, EGL, upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

Appendix A

Fine-tuned FlanT5 and LongT5 models for DG have been made public under the Apache-2.0 License in the Huggin Face[2] platform (Table A1).

Table A1: List of published models

Variant	URL
FlanT5-Base	https://huggingface.co/rrodrigu3z/flan-t5-base-joint-dg (accessed on 1 January 2025)
FlanT5-Large	https://huggingface.co/rrodrigu3z/flan-t5-large-joint-dg (accessed on 1 January 2025)
FlanT5-Base (LoRA)	https://huggingface.co/rrodrigu3z/flan-t5-base (accessed on 1 January 2025)
FlanT5-XL (LoRA)	https://huggingface.co/rrodrigu3z/flan-t5-xl/tree/main (accessed on 1 January 2025)
LongT5-Base	https://huggingface.co/rrodrigu3z/long-t5-tglobal-base-joint-dg (accessed on 1 January 2025)
LongT5-Large	https://huggingface.co/rrodrigu3z/long-t5-tglobal-large-joint-dg (accessed on 1 January 2025)
LongT5-XL (LoRA)	https://huggingface.co/rrodrigu3z/long-t5-tglobal-xl/tree/main (accessed on 1 January 2025)

[2] https://huggingface.co/ (accessed on 1 January 2025).

Appendix B

Distractors generated for all datasets using GPT-3.5-turbo-1106 via OpenAI API can be downloaded in the following URL: https://dg-inferences.s3.amazonaws.com/gpt-3.5-baseline/chatgpt_predictions.jsonl (accessed on 1 January 2025).

References

1. Butler AC. Multiple-choice testing in education: are the best practices for assessment also good for learning? J Appl Res Mem Cogn. 2018;7(3):323–31. doi:10.1016/j.jarmac.2018.07.002.
2. Shin J, Guo Q, Gierl MJ. Multiple-choice item distractor development using topic modeling approaches. Front Psychol. 2019;10:825. doi:10.3389/fpsyg.2019.00825.
3. Lane S. Handbook of test development. New York, NY, USA: Routledge; 2015. doi:10.4324/9780203102961.
4. Haladyna TM, Rodriguez MC. Developing and validating test items. New York, NY, USA: Routledge; 2013. doi:10.4324/9780203850381.
5. Zhang L, VanLehn K. Evaluation of auto-generated distractors in multiple choice questions from a semantic network. Interact Learn Environ. 2021;29(6):1019–36. doi:10.1080/10494820.2019.1619586.
6. Kumar AP, Nayak A, Manjula Shenoy K, Goyal S. A novel approach to generate distractors for multiple choice questions. Expert Syst Appl. 2023;225(7):120022. doi:10.1016/j.eswa.2023.120022.
7. UNESCO. Artificial intelligence in education: challenges and opportunities for sustainable development [Internet]. Work Pap Educ Policy. 2019;7:46. [cited 2025 Jan 1]. Available from: https://en.unesco.org/themes/education-policy-.
8. Zhou Q, Yang N, Wei F, Tan C, Bao H, Zhou M. Neural question generation from text: a preliminary study. In: Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics). Berlin/Heidelberg, Germany: Springer; 2018. doi:10.1007/978-3-319-73618-1_56.
9. Rodriguez-Torrealba R, Garcia-Lopez E, Garcia-Cabot A. End-to-end generation of multiple-choice questions using text-to-text transfer transformer models. Expert Syst Appl. 2022;208:118258. doi:10.1016/j.eswa.2022.118258.
10. Gao Y, Bing L, Li P, King I, Lyu MR. Generating distractors for reading comprehension questions from real examinations. In: Proceedings of the AAAI Conference on Artificial Intelligence; 2019 Jan 27–Feb 1; Honolulu, HI, USA. doi:10.1609/aaai.v33i01.33016423.
11. Liang C, Yang X, Dave N, Wham D, Pursel B, Giles CL. Distractor generation for multiple choice questions using learning to rank. In: Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications; 2018 Jun 5; New Orleans, LA, USA. p. 284–90. doi:10.18653/v1/w18-0533.
12. Offerijns J, Verberne S, Verhoef T. Better distractions: transformer-based distractor generation and multiple choice question filtering [Internet]. [cited 2021 Nov 13]. Available from: http://arxiv.org/abs/2010.09598.
13. Qiu Z, Wu X, Fan W. Automatic distractor generation for multiple choice questions in standard Tests. arXiv:2011.13100. 2021. doi:10.18653/v1/2020.coling-main.
14. Papineni K, Roukos S, Ward T, Zhu WJ. BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics—ACL '02; 2002 Jul 7–12; Philadelphia, PA, USA.
15. Lin CY. Rouge: a package for automatic evaluation of summaries. In: Proceedings of the Workshop on Text Summa-Rization Branches Out (WAS 2004); 2004 Jul 25–26; Barcelona, Spain.
16. Vijayakumar AK, Cogswell M, Selvaraju RR, Sun Q, Lee S, Crandall D, et al. Diverse beam search: decoding diverse solutions from neural sequence models [Internet]. [cited 2021 Dec 23]. Available from: https://arxiv.org/abs/1610.02424.
17. Chung HW, Hou L, Longpre S, Zoph B, Tay Y, Fedus W, et al. Scaling instruction-finetuned language models. J Mach Learn Res. 2024;25(70):1–53.
18. Guo M, Ainslie J, Uthus D, Ontanon S, Ni J, Sung YH, et al. LongT5: efficient text-to-text transformer for long sequences. arXiv:2112.07916. 2021.

19. Lai G, Xie Q, Liu H, Yang Y, Hovy E. RACE: large-scale ReAding comprehension dataset from examinations. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing; 2017 Sep 9–11; Copenhagen, Denmark. doi:10.18653/v1/d17-1082.
20. Zhang T, Kishore V, Wu F, Weinberger KQ, Artzi Y. BERTScore: evaluating text generation with BERT. arXiv:1904.09675. 2019.
21. Sutskever I, Vinyals O, Le QV. Sequence to sequence learning with neural networks [Internet]. [cited 2025 Jan 1]. Available from: http://arxiv.org/abs/1409.3215.
22. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. Adv Neural Inf Process Syst. 2017;30:5999–6009.
23. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I. Language models are unsupervised multitask learners. OpenAI Blog. 2019;1(8):9.
24. Devlin J, Chang MW, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding [Online]. [cited 2025 Jan 1]. Available from: http://arxiv.org/abs/1810.04805.
25. Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, et al. Exploring the limits of transfer learning with a unified text-to-text transformer [Internet]. J Mach Learn Res. 2019;21:1–67. [cited 2025 Jan 1]. Available from: http://arxiv.org/abs/1910.10683.
26. Welbl J, Liu NF, Gardner M. Crowdsourcing multiple choice science questions. In: Proceedings of the 3rd Workshop on Noisy User-generated Text; 2017 Sep 7; Copenhagen, Denmark. doi:10.18653/v1/w17-4413.
27. Pennington J, Socher R, Manning C. Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2014 Oct 25–29; Doha, Qatar. doi:10.3115/v1/d14-1162.
28. Chomphooyod P, Suchato A, Tuaycharoen N, Punyabukkana P. English grammar multiple-choice question generation using text-to-text transfer transformer. Comput Educ Artif Intell. 2023;5:100158. doi:10.1016/j.caeai.2023.100158.
29. Sanh V, Debut L, Chaumond J, Wolf T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv:1910.01108. 2019.
30. Clark P, Cowhey I, Etzioni O, Khot T, Sabharwal A, Schoenick C, et al. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. arXiv:1803.05457. 2018.
31. Talmor A, Herzig J, Lourie N, Berant J. CommonSenseqa: a question answering challenge targeting com-monsense knowledge. In: NAACL HLT 2019—2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; 2019 Jun 2–7; Minneapolis, MN, USA.
32. Richardson M, Burges CJC, Renshaw E. MCTest: a challenge dataset for the open-domain machine comprehension of text. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing; 2013 Oct 18–21; Seattle, WA, USA.
33. Mihaylov T, Clark P, Khot T, Sabharwal A. Can a suit of armor conduct electricity? A new dataset for open book question answering. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing; 2018 Oct 31–Nov 4; Brussels, Belgium. doi:10.18653/v1/d18-1260.
34. Khashabi D, Min S, Khot T, Sabharwal A, Tafjord O, Clark P, et al. UNIFIEDQA: crossing format boundaries with a single QA system. arXiv:2005.00700. 2020.
35. Khashabi D, Kordi Y, Hajishirzi H. UnifiedQA-v2: stronger generalization via broader cross-format training. arXiv:2202.12359. 2022.
36. Lewis M, Liu Y, Goyal N, Ghazvininejad M, Mohamed A, Levy O, et al. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics; 2020 Jul 5–10; Online. doi:10.18653/v1/2020.acl-main.703.
37. Raina V, Gales M. Multiple-choice question generation: towards an automated assessment framework. arXiv:2209.11830. 2022.
38. Lin T, Wang Y, Liu X, Qiu X. A survey of transformers. AI Open. 2022;3(120):111–32. doi:10.1016/j.aiopen.2022.10.001.

39. Mangrulkar S, Gugger S, Debut L, Belkada Y, Paul S, Bossan B. PEFT: state-of-the-art parameter-efficient fine-tuning methods. San Francisco, CA, USA: GitHub; 2022.
40. Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi A, et al. Transformers: state-of-the-art natural language processing [Internet]. [cited 2025 Jan 1]. Available from: https://arxiv.org/abs/1910.03771.
41. Hu EJ, Shen Y, Wallis P, Allen-Zhu Z, Li Y, Wang S, et al. LoRA: low-rank adaptation of large language models. In: ICLR 2022—10th Inter-National Conference on Learning Representations; 2022 Apr 25–29; Virtual.
42. Dettmers T, Pagnoni A, Holtzman A, Zettlemoyer L. QLoRA: efficient finetuning of quantized LLMs. Adv Neural Inf Process Syst. 2023;36:10088–115.
43. Jauregi Unanue I, Parnell J, Piccardi M. BERTTune: fine-tuning neural machine translation with BERTScore. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers); 2021 Aug 1–6; Online. doi:10.18653/v1/2021.acl-short.115.
44. Reimers N, Gurevych I. Sentence-BERT: sentence embeddings using Siamese BERT-networks. 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); 2019 Nov 3–7; Hong Kong, China. doi:10.18653/v1/d19-1410.
45. Hanna M, Bojar O. A fine-grained analysis of BERTScore. In: Proceedings of the Sixth Conference on Ma-chine Translation; 2021 Nov 10–11; Punta Cana, Dominican Republic. p. 507–17. [cited 2025 Jan 1]. Available from: https://aclanthology.org/2021.wmt-1.59/.
46. Crossley SA, Bradfield F, Bustamante A. Using human judgments to examine the validity of automated grammar, syntax, and mechanical errors in writing. J Writ Res. 2019;11(2):251–70. doi:10.17239/jowr-2019.11.02.01.
47. Doughty J, Wan Z, Bompelli A, Qayum J, Wang T, Zhang J, et al. A comparative study of AI-generated (GPT-4) and human-crafted MCQs in programming education. In: Proceedings of the 26th Australasian Computing Education Conference; 2024 Jan 29–Feb 2; Sydney, NSW, Australia. doi:10.1145/3636243.3636256.