DISTO: Textual Distractors for Multiple Choice Reading Comprehension Questions Using Negative Sampling

Bilal Ghanem
University of Alberta
Department of Computing Science
Alberta Machine Intelligence Institute
bilalhgm@gmail.com

Alona Fyshe
University of Alberta
Department of Computing Science
Department of Psychology
Alberta Machine Intelligence Institute
alona@ualberta.ca

ABSTRACT

Multiple choice questions (MCQs) are a common way to assess reading comprehension. Every MCQ needs a set of distractor answers that are incorrect, but plausible enough to test student knowledge. However, good distractors are hard to create. Distractor generation (DG) models have been proposed, and their performance is typically evaluated using machine translation (MT) metrics. However, MT metrics can misjudge the suitability of generated distractors. We propose DISTO: the first *learned* evaluation metric for generated distractors. We show that DISTO scores are highly correlated with human ratings of distractor quality. At the same time, DISTO ranks the performance of state-of-the-art DG models very differently from MT-based metrics, showing that we should be cautious when using MT metrics for distractor evaluation.

Keywords

Multiple Choice Questions, Question Distractors, Distractors Evaluation, Reading Comprehension

1. INTRODUCTION

With the rise of online learning, it has become increasingly important to have large question banks so that student tests can be unique in terms of question content and question order. In addition, there is increasing interest in diversifying content to appeal to students with multiple backgrounds and interests, again requiring that novel questions be generated for new material. Multiple choice questions (MCQs) are a common choice for assessing reading comprehension (RC) because they allow for quick automatic evaluation and consistent scoring. However, MCQs also require the creation of *distractor* answers, and good distractors are crucial to the utility of a MCQ [11]. Creating good distractors is a challenging task, and identifying a good distractor can be equally challenging.

B. Ghanem and A. Fyshe. Disto: Textual distractors for multiple choice reading comprehension questions using negative sampling. In B. Paaßen and C. D. Epp, editors, *Proceedings of the 17th International Conference on Educational Data Mining*, pages 6–17, Atlanta, Georgia, USA, July 2024. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.12729766

Article:
Come and see the India elephants and the new tigers from America. The bears are waiting to meet you, and the monkeys from China are waiting to throw things to you. The lovely dogs from Australia are waiting to laugh at you. The giraffes from Brazil are waiting to look down on you.

Question:
From the passage we can guess the animal "giraffe" must be very _____ .

Answer:
Tall

Gold Distractors:
1) Fat 2) Long 3) Strong

Generated Distractors:
1) Hairy 2) Loud 3) Wide

Figure 1: An example MCQ from the RACE dataset [25], with generated distractors produced using a T5 model. Though the generated distractors are reasonable, many MT metrics would assign them a score of zero because they share zero words in common.

To reduce the effort and time needed to create good distractor answers for MCQs, many groups have developed models for distractor generation (DG) (see Table 1). DG models are often evaluated with machine translation (MT) metrics (e.g. BLEU score) [54, 21, 50]. But, MT metrics were not designed to evaluate MCQ distractors, and so they do not consider several important characteristics of good distractors (e.g., context consistency). In addition, MT metrics require reference texts, making them less useful. To the best of our knowledge, we are the first to propose a specialized *learned* metric to evaluate textual distractors in an automatic way. Our method evaluates distractors by considering the context, question and the correct answer, thus scoring the distractor in a more holistic fashion.

An example of distractor evaluation appears in Figure 1, where we present an MCQ from the RACE dataset [25]. The T5-generated distractors [44] are good plausible answers, but they do not match the gold distractors, so MT metrics like BLEU [37] and ROUGE [28] will give them a score of

zero. This highlights the fact that distractor evaluation metrics should not simply consider the textual overlapping with gold distractors, as they are not an exhaustive set of all possible distractors. Similar observations have been made for answer evaluation for free form answers [3]. In Figure 1, we can select many other adjectives (e.g. fast, short, thin) to create other good distractors for this question.

The semantic relatedness of distractors and the answer is another important aspect ignored by MT metrics. For example, for a question that asks for the capital of France, "*Paris Hilton*" is a bad distractor. This celebrity is not contextually or semantically related to the answer, though her name does share a word in common with the correct answer.

If we want to accurately and automatically score distractors, we must consider the DG task and build a scoring method from scratch, not simply borrow metrics from machine translation. Thus we propose DISTO, the first *learned* distractor evaluation metric, which uses a negative sampling (NS) strategy to differentiate good distractors from bad. We show that DISTO's scores correlate highly with human ratings of distractor quality. We then re-evaluate several state-of-the-art DG models and find that, compared to MT metrics, DISTO produces a different performance ranking of those models. Our contributions are as follows:

- We describe automatic distractor *evaluation*, an application that has been largely overlooked. We show that MT metrics are likely not suitable for distractor evaluation.
- We propose a distractor evaluation metric (DISTO) that uses a negative sampling technique to model the consistency of a given set of distractors with respect to the context. Unlike previous approaches, DISTO does not apply any kind of text-based similarities to evaluate the generated distractors [1].

In the next section, we set the stage with a review of previous work on distractor generation and evaluation. In Section 3, we describe the methodology behind DISTO and our proposed negative sampling technique. From there we perform several evaluations of DISTO: a human evaluation, an ablation study, and a comparison of DISTO against MT techniques commonly used for evaluating DG models.

2. RELATED WORK

Distractor evaluation

Having DG-specific model-based scoring metrics is important because the current evaluation techniques are either inaccurate (MT metrics) or very time-consuming (human evaluation). Table 1 shows past DG techniques, along with the methods used for evaluating the quality of generated distractors. Almost all use manual evaluation which is costly, and cannot give real-time results during model development. We need a proper automatic metric, but current automatic scoring methods can produce incorrect results (see example in Figure 1). In fact, tuning a machine learning model to maximize MT scores could lead to overfitting, which may be one of the reasons DISTO rankings are so different from those previously reported (see Section 4.5).

Previous work has used several methods to manually evaluate the generated distractors. Most use a Likert scale to assess the quality of distractors for a given set of categories (e.g. fluency, coherence, relevance, diversity, etc) [24, 20, 48, 55, 59, 42, 31, 54, 6, 7]. Other works infer quality by administering an MCQ test that includes the new distractors [32, 15, 9]. Others evaluate the quality by asking the annotators questions like: "*does the answer make sense in relation to the question?*" or by asking them to select only good distractors [34, 21].

And finally, the NLP community has explored learned metrics for tasks like MT, for which BLEURT has become popular [57, 47]. These metrics have been taken up in DG research, which inspired us to explore a learned metric tailored to the distractor evaluation task.

Creating learned metrics has been explored in the NLP community for tasks like MT, for which BLEURT has become popular [57, 47].

Distractor Generation (DG)

DG research can be divided mainly into two main lines of research: generative models and ranking models. The former uses LLMs to generate distractors for a given MCQ. In Table 1, we only list generative approaches as we focus mainly on these approaches for evaluation. On the other hand, ranking models frame DG as a ranking problem where the model must rank a distractor within a given candidate set [56, 40, 52, 26, 13, 10]. Ranking models do not generate distractors, as they assume that the distractor set is provided. This type of model can be evaluated using information retrieval metrics since the problem has been reformulated as a ranking task.

3. METHODOLOGY

We now outline the methodology behind DISTO, which requires several steps. First we created a dataset by combining several RC datasets, and converting MCQs with >1 distractor into several MCQs with only one distractor. We then used four negative sampling methods to create training examples that represent bad distractors. We then train three distractor evaluation architectures on this expanded and negatively-sampled dataset and select the most promising to become the architecture underlying DISTO.

3.1 Data Augmentation & Negative Sampling

Good distractors are semantically consistent with the question context. In order to evaluate distractor context, we propose to learn the distractors' consistency from the currently available RC datasets that were created by human experts. In those datasets, each *Article-Question* pair is associated with an answer and $N \in [1,3]$ distractors. Using these datasets, we can learn the characteristics of a good distractor set, but not what makes a *bad* distractor. Thus, we use a negative sampling (NS) technique with distractor augmentation to create examples of bad distractor sets. In this way, we can model both cases (good and bad distractor sets) and assign a consistency score for a given distractor in a context.

[1]The code is available at: `https://github.com/bilalghanem/DISTO`

Table 1: A survey of the existing distractor generation models. The $_\delta$ sign in the "Domain/Source" column means an online published dataset/corpus. For the Language column: En-English, Cn-Chinese, Se-Swedish. MT in the "Evaluation" column means Machine Translation metrics were used in the evaluation. Note that Guo et al.[12] proposed an assessment system using MCQs and only focused on evaluating the created questions.

Study	Approach	Lang.	Domain/Source	Evaluation
[32]	Hypernyms from Word-Net lexicon	En	Textbooks	Manual
[41]	Phonetic and morphological similarities	En	Pronouncing Dictionary	Manual
[24]	Word2Vec semantic similarity	En	Textbooks	Manual
[12]	Word2Vec semantic similarity	En	Wiki	N/A [2]
[15]	Ngrams co-occurrence likelihood	En	Google Ngrams$_\delta$	Manual
[20]	Word2Vec semantic similarity	Cn	Textbooks, Wiki	Manual
[48]	Structural similarities in an ontology	En	Educational ontology	Manual
[55]	BERT with [MASK] filling	Cn	Textbooks	Manual
[9]	LSTM encoder-decoder	En	RACE$_\delta$	MT + Manual
[4]	BERT with [MASK] filling	En	RACE$_\delta$	MT
[59]	LSTM encoder-decoder	En	RACE$_\delta$	MT + Manual
[34]	GPT-2 Transformer	En	RACE$_\delta$	MT + Manual
[42]	LSTM encoder-decoder	En	RACE$_\delta$	MT + Manual
[31]	LSTM encoder-decoder(s)	En	RACE$_\delta$	MT + Manual
[21]	BERT with [MASK] filling	Se	SweQUAD-MC$_\delta$	Manual
[54]	T5 Transformer	En	RACE$_\delta$, Cosmos QA$_\delta$	MT + Manual
[50]	T5 Transformer	En	RACE$_\delta$	MT
[6]	GPT-3 Transformer	En	RACE$_\delta$	MT + Manual
[36]	FAIRSEQ + BERT + Word2Vec	En	ESL tests	Manual
[14]	T5 Transformer	En	RACE$_\delta$, EduQG$_\delta$	MT

Each instance in an RC dataset contains an article Ar, question Q, answer An, and N distractors D=$\{d_1..d_N\}$. We want to replace those contextually consistent distractors (good distractors) with inconsistent ones (bad distractors). Our proposed model takes $[Q, An, d, Ar]$ as an input and outputs a consistency score [0-1].

We design our model to take a *single distractor* d in each instance. For questions with more than one distractor, we create N total training instances, one for each distractor. Each training instance contains one of the N distractors, along with the same $[Q, An, Ar]$ set. Since our goal is to design a metric that evaluates DG models, we use a regression model (see Section 3.2). We will train the model to predict a score of one for instances with good distractors and a score of zero for instances with our newly-created bad distractors. In other words, the model is trained to predict 0 for a bad distractor, 1 for a good distractor, and at test time will produce an intermediate value $[0, 1]$ depending on the distractor plausibility.

In order to build our bad distractor training instances, we create bad distractors using one of the following techniques:

1) Answer Replication:. Here, distractors are a copy of the correct answer. This teaches our models to produce a low score in cases where a DG model generates a distractor too similar or identical to the answer.

2) Random Distractor:. We build a pool of all distractors (∼310K distractors) using the RC datasets. From this we randomly select distractors to build new training instances. We ensure that the random distractor is not equal to a current good distractor in a given instance. This technique teaches the model to penalize generated distractors that are totally inconsistent with the context.

3) Farthest Point in a Cluster. The previous two negative sampling techniques (replicated and random) are easy to sample, but they are also easy to detect and thus not very challenging. To build a good distractor evaluation metric, we need negative samples that are more plausible, but still bad. We need to sample from our pool of all distractors in a more targeted way.

The true distractors of an MCQ will have characteristics in common with the correct answer (similar length, semantically related, etc.). Thus, we use a clustering technique to identify bad distractors that share those characteristics with true distractors. We use the following set of features to represent distractor characteristics:

- BERT Embeddings. These capture the semantic relatedness. [3]
- Bag-of-POS Tags. Good distractors have similar POS structures [39]. This approach involves creating a representation similar to the concept of a "bag of words," but at the level of Part-of-Speech (POS) tags. Thus, we build a Term Frequency (TF) vector of POS tags for each distractor, using the spaCy POS tagger.
- Bag-of-Named Entity Types. As noted for POS tags,

[3]We use the "bert-base-uncased" model from the BERT-as-Service library to extract the embeddings.

we have noticed that relevant distractors contain similar named entity types. We build a TF vector of named entity types for each distractor.

- Distractor Length: good distractors usually have similar numbers of tokens. Thus, we include the number of tokens for a given distractor.

Using the pool of all the distractors (∼310K distractors) and the distractor representation outlined above, we use K-means clustering to build distractor clusters. We set the number of clusters (k) to 200. [4]

After building the clusters, for each $[Q, An, d, Ar]$, we determine the cluster for the true (good) distractor d. Our goal is to replace the good distractor with another one that is somewhat similar (but not too similar). Using Euclidean distance, we choose the *farthest* point in d's cluster as the negatively sampled bad distractor.

We did experiment with using the *nearest* distractor in a cluster, but found that method produced distractors that were often good, rather than the bad distractors we desire for negative sampling. For instance, for a question about a group of animal friends, for the distractor "*A tiger named benny*" the closest distractor in the cluster is "*A dog called buck*". On the other hand, the farthest point in the cluster is "*A midnight madness event*". The nearest distractor tended to be too plausible, whereas the furthest gave a distractor that is close, but not too close.

4) BERT [MASK] Filling:. BERT is trained to fill masked tokens in tokenized sentences. We leverage this functionality to rewrite distractors by replacing nouns, verbs, and adjectives in a good distractor to create an augmented bad distractor. For each masked token BERT returns the top N most probable tokens along with their probabilities. We discard these probabilities and select uniformly from the top N tokens, ensuring the selected token is not equal to the original masked token. This technique introduces lexical alterations to the good distractor components while maintaining fluency. For instance, the good distractor "*They focus on bible stories*" is replaced with "*They drew on the sand*". Note that BERT can choose to replace words with something other than nouns, verbs and adjectives, if the replacement is deemed probable.

From each original $[Q, An, d, Ar]$ tuple, we create *four negative instances* using one of the above mentioned distractor-creation techniques. Each time we substitute a good distractor d with newly-created one, we set the regression score of the new instance to zero. We validated these techniques manually by examining the corresponding articles, questions, and answers to make sure that the newly-created bad distractors do not fit the context. We found the newly-created bad distractors to be valid bad distractors, with some being very far from the given context and others being closer but still invalid. It is worth mentioning that we found a few cases where the created bad distractors could be considered as good distractors in the given context. As in all negative sampling techniques, this is rare and often unavoidable.

[4]We tested $k = [50, 100, 200, 300]$ but found that 200 gave us the most coherent clusters.

Table 2: A summary of the data splits for each dataset after preprocessing. "Flattened" refers to the process of taking one instance that has N distractors and creating N instances, each with one distractor.

Dataset	Train	Val.	Test
Cosmos QA	21,397	2,726	2,369
DREAM	6,107	2,035	2,036
MCScript	14,189	2,020	3,610
MCtest	1,200	200	599
Quail	9,215	1,025	2,164
RACE	40,385	2,234	2,201
SCIQ	10,480	887	883
Total	102,973	11,127	13,862
Flattened	274,366	27,303	32,322
+ NS	1,043,464	104,485	124,724

We use several MCQ datasets: CosmosQA [18], DREAM [49], MCScript [35], MCtest [45], Quail [46], RACE [25], and SCIQ [52]. We preprocess these datasets to remove instances that have a corrupted answer, question, or article (e.g. empty texts, filled with punctuation marks, etc.), and instances that have a "none of the above" answer/distractor. [5] Also, we remove instances with no distractors or instances that have duplicate distractors. We use the original data splits for all the datasets except Quail where we sample 0.1 for validation as it does not have a defined validation set. In Table 2, we present the size of dataset splits and the final dataset size after applying NS.

3.2 Distractor Evaluation Architectures

We use a pretrained encoder transformer model and we add a linear layer with a sigmoid function as an output. The model takes the $[Q, An, d, Ar]$ as an input text and outputs a score. We experimented with BERT [5], RoBERTa [29], Longformer [1], and the distilled versions of BERT and RoBERTa from HuggingFace library [53] [6]. We found that the DistillRoBERTa model gave us the best results in our initial experiments, thus we based our subsequent experiments on that model. In order to capture the relation between the distractors with the context, we experiment with two architectures:

Separated Text (SepT):. In this architecture, we feed the input texts separated with special tokens (surrounded by square brackets) to the encoder model. The input text structure looks like the following: [QUES] Q [ANS] An [DIS] D [ART] Ar. After that, we use the first token from the encoder (classification token [CLS] [7]) and feed it to a sigmoid function to map the logits into the 0-1 range.

[5]The "none of the above" answer/distractor is not a useful training example because there is no consistency between it and its given context.

[6]We use the base version of these models.

[7]We refer to the classification token in the DistillRoBERTa model as [CLS] although the classification token for this model was renamed to <s>.

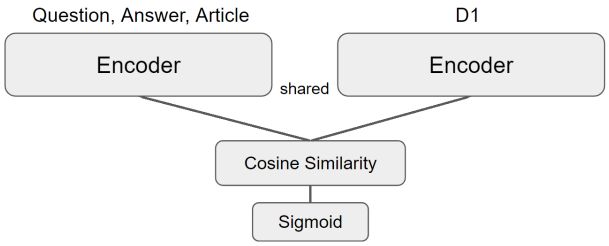

Figure 2: The SIAM-COS-SIM model structure. The structure has two parallel encoders; one encoder is fed the question, answer and article, the other encoder receives the distractor (D1). We then measure the cosine similarity of the classification token for the two inputs, and pass that through a sigmoid to produce a probabilistic output representing the goodness of the distractor..

SIAM-COS-SIM:. This model uses a Siamese model architecture [33] with DistillRoBERTa as an encoder to capture the similarity between the distractors and the context. For that, we feed [QUES] Q [ANS] An [ART] Ar to the first branch and the [DIS] D to the second branch. After that, we measure the cosine similarity between the [CLS] tokens of both branches, and finally we apply sigmoid function to the similarity scores. Our hypothesis in this model is that, if both [CLS] tokens are similar then the distractor is relevant to the context, and thus a good distractor. Figure 2 illustrates the structure.

Bag-of-words (BOW):. In addition to the two architectures, we create a baseline using Bag-of-words with Tf-Idf weighting scheme and Linear Regression classifier (BOW) [8]. Here, we use the same input format as in the SepT architecture (Q [ANS] An [DIS] D [ART] Ar) when we transform it to Tf-Idf vectors.

Model Settings and Metrics For our architectures, we use the Adam optimizer [23] and 1e-5 learning rate value. We set the maximum sequence length to 512. Since we formulate the problem as a regression problem, we use the Mean Squared Error (MSE) loss function. In all of our experiments, we use the early stopping regularization technique. To evaluate the models, we use Mean Absolute Error (MAE), and we follow the Workshop on Machine Translation (WMT) Metrics shared task [30] by using the Pearson Correlation [2].

4. EVALUATION OF DISTO

Now we turn to the evaluation of DISTO. First is an intrinsic evaluation, in which we consider the fit of each model to held out distractor examples. Then we perform a human evaluation to test for correlation of DISTO vs Amazon Mechanical Turk (AMT) workers vs gold labels. We perform a subsequent human evaluation for correlation of DISTO vs AMT workers on *generated* distractors (rather than gold distractors). We then perform an ablation test to study the importance of each DISTO input. We end by using DISTO alongside several MT metrics to illustrate the differences that arise when evaluating with each.

[8]We use the a Linear Regression implementation from the Scikit-Learn library.

Table 3: Performance of distractor evaluation models: mean absolute error (MAE) and Pearson correlation between the true and predicted distractor quality scores. The results show that SepT produces the best predictions of distractor quality.

Model	MAE (%)	Pearson_{corr}
BOW	69.0	02.9
SIAM-COS-SIM	11.4	80.2
SepT (DISTO)	**03.8**	**94.1**

4.1 Evaluation of Model fit

Table 3 gives model performance based on held out data. The results show that both of the proposed architectures show a large performance improvement over the baseline BOW model. We attribute this to the ability of both architectures to model the semantics of the inputs, where the Tf-Idf vectors in the BOW cannot properly capture the semantic meaning. This confirms the importance of semantic meaning for this task. Regarding the two architectures, SepT is most accurate, with a very high positive correlation and almost zero MAE value. This could be because one of the training objectives of the transformers-based encoder models is the Next Sentence Prediction (NSP). NSP models the semantic relatedness of the input texts, which is also important in distractor evaluation. In the rest of our experiments, we use SepT model and we refer to it as *DISTO*.

4.2 Human Evaluation

To validate DISTO's performance, we conduct a human evaluation experiment using AMT. Though AMT workers are not trained educators, we provide this evaluation as a necessary first step for measuring DISTO performance.

We sample 50 good distractors from the datasets, and 50 bad distractors using the NS technique (Section 3.1) for a total of 100 instances. For this evaluation, the NS bad distractors were created using either the "Farthest Point in a Cluster" or "BERT [MASK] Filling" techniques (distractors created by the duplicated answer and random-based distractors are easily spotted). For each worker, we display one distractor along with its article, question, and answer. We ask workers to rate the distractor as either bad, neutral, or good, within the given context. In Appendix A, we show a sample from the annotation interface. Because this is an announced task in AMT, we prepared a short quiz (see Figure 4) to select the strongest workers. From this list of strong workers, we request five workers to score each of the 100 instances, and then average their ratings. Since this is a relatively difficult task, we add instructions and several example questions to ensure that workers understand the task completely.

One worker performed very poorly on this task (less than 30% accuracy according to the gold labels). Thus, we discarded this worker's data and proceeded with the remaining four. We compute the annotation agreement among the workers using Fleiss-Kappa [8], and find a moderate agreement (0.45). This shows the difficulty of the task; even humans disagree about what makes a distractor good or bad. Since the agreement between the annotators is not high, we

Table 4: Correlation of distractor suitability scores for a $[Q, An, Ar]$ tuple. We calculate Pearson Correlation between DISTO, AMT (workers) average ratings, and gold data labels. All sources of quality judgements are highly correlated ($p < 0.001$).

Experiment	Pearson_{corr}	P_{value}
Gold Data vs. Workers	0.78	< 0.001
Gold Data vs. DISTO	0.94	< 0.001
Workers vs. DISTO	0.81	< 0.001

can also conclude that the "Farthest Point in a Cluster" and "BERT [MASK] Filling" techniques produce distractors that are not easily discarded by the annotators.

Table 4 shows the Pearson Correlation between DISTO, workers averaged ratings, and the gold data. The correlation of worker annotations to the gold data is high (0.78), but not as high as DISTO's correlation with the gold data (0.94). This demonstrates the difficulty of the task for the human annotators. This also demonstrates the effectiveness of the data used to train DISTO, which was curated by professional educators. DISTO is also highly correlated with human annotations (Pearson Correlation 0.81). In general, all correlation results are high and significant, especially for the "Gold Data vs. DISTO", which is consistent with the results in Table 3. In Appendix 4.3 we use distractors generated by the DG models described in Section 4.5 and find that DISTO is still significantly correlated. Note that it is not possible to run this human evaluation with MT metrics. But, as we will see in Section 4.5, DISTO is negatively correlated with MT metrics, implying that a suitability score derived from MT metrics would not fare well here.

As we show in the example in Figure 1, MT metrics may not be suitable for DG evaluation because MT metrics treat gold distractors as the only correct good distractors. In truth there is much more diversity amongst good distractors than amongst correct translations. In the extreme case the unigram overlap can be 0 between two good distractors for the same MCQ; the same is likely not true for two acceptable translations of the same sentence. Our results show that DISTO is a coherent evaluation metric for DG that considers the semantics of the distractors within a given context, making it more likely to assign a high score to many examples from the diverse set of all possible good distractors.

4.3 Out-of-Domain Human Evaluation

In Section 4.1, we used a human evaluation to validate the DISTO model on data sampled from DISTO's expanded negatively-sampled test set. This makes that experiment domain-dependent, as DISTO is trained on the same type of data used in the human evaluation, introducing the possibilkity that our results are biased. To address this, we conduct another human evaluation using the distractors generated by the DG models from the previous section. This way we are evaluating DISTO on distractors that were not created using the sampling techniques in Section 3.1. Similar to our previous experiment, we sample 100 instances from each DG model for the same question, answer, and article sets. AMT workers settings (number of workers, quiz, etc.) are as in the previous experiment.

Table 5: Correlation between DISTO and AMT workers using the generated distractors (rather than gold and negatively sampled distractors as in Table 4). We generate distractors using each DG model and collect both AMT and DISTO scores for those distractors. P_{value} ranges: $\leq$ 0.001, $\leq$ 0.05. DISTO and human scores are significantly correlated for all models, implying that DISTO generalizes to generated distractors after being trained only on human-created distractors.

Model	Pearson_{corr}	P_{value}
GDRCQ	0.75	≤ 0.001
BDG	0.3	≤ 0.05
GPT-2	0.28	≤ 0.05
T5	0.63	≤ 0.001
$\text{T5}_{disjoint}$	0.6	≤ 0.001

Table 6: Evaluation results of the distractor evaluation models. "DISTO - X" means without "X" included in the context.

Model	MAE (%)	Pearson_{corr}
DISTO	03.8	94.1
DISTO - Question	05.5	91.1
DISTO - Article	08.1	88.1
DISTO - Answer	18.0	71.2

The results in Table 5 show varying degrees of correlation across different DG models. The GDRCQ model demonstrates a substantial correlation of 0.75, indicating a strong alignment between DISTO-generated distractors and human-evaluated quality. This suggests that the distractors produced by GDRCQ are effective in evaluating DISTO's performance. In contrast, the BDG model exhibits a moderate correlation of 0.3, while GPT-2 shows a slightly lower correlation of 0.28. These values, though lower than GDRCQ, still signify a significant association between the distractors generated by these models and the human evaluators' judgments. Moreover, the T5 model and its disjoint variant ($\text{T5}_{disjoint}$) showcase high correlations of 0.63 and 0.6, respectively. This implies a robust alignment between DISTO's performance and the assessments made by human evaluators when using distractors generated by T5 models.

All correlation results are statistically significant, with p-values less than 0.001 for GDRCQ, T5, and $\text{T5}_{disjoint}$, and less than 0.05 for BDG and GPT-2. This statistical significance reinforces the reliability of the observed correlations. The out-of-domain human evaluation using various DG models underscores the adaptability of DISTO to different distractor generation approaches. The correlations indicate that DISTO performs well across a range of distractors, emphasizing its versatility and robustness in handling diverse types of generated content.

4.4 The Importance of Context

DISTO models the consistency of good distractors with their context (article, question, and answer). Here we perform an ablation test on the context tuple to quantify the importance of each element in computing accurate DISTO scores. We train three new DISTO models, ablating one of the three context elements and compare those results to DISTO trained on all three context elements. Table 6 presents the

results, and shows that the most valuable context element is the answer. Removing the answer results in a 14.2% and 22.9 drop in terms of MAE and Pearson Correlation, respectively. The results also show, unexpectedly, that including the *question* in the context is the least important. Removing the article results in a modest drop in MAE and correlation, likely because good distractors have some relation to the article. Thus, the semantic relatedness between the distractors and the answer is most important for determining the suitability of a distractor, whereas the question itself appears to be of little importance.

4.5 DISTO for DG Models

As another method of evaluating DISTO's utility, we consider the relative performance of existing DG models using DISTO scores, and compare that relative performance to those derived from MT metrics. Several such models have been previously proposed, each using different training algorithms. As seen in Table 1, transformer models have been a popular approach to generating distractors over the last few years.

We use previously proposed models developed [9, 4, 34, 54]. [9] In greater detail, our models are:

1) GDRCQ:. This work [9] uses an LSTM [16] encoder-decoder model with dynamic and static attention. This method is an example of a distractor generation model based on RNN seq2seq architectures. The authors use Glove word embeddings [38] for initialization and finetune them during the training process.

2) BDG:. This approach uses the initial training task from BERT (filling masked tokens) to generate distractors [4]. Given the context, the approach appends a [MASK] token at the end of the context to let the model generate the distractors, word by word. This work also proposes an answer-negative regularization technique to combat answer duplication.

3) GPT-2:. This model uses the GPT-2 [43] transformer model to generate distractors. We follow the instructions from [34] to implement the model. For decoding, we use beam search sampling with a beam size of 6. We use Adam optimizer with a 3e-4 learning rate.

4) T5:. The T5 language model [10] achieved SOTA results on several generative NLP tasks, so we use it here to generate distractors. Inspired by the work [54], we use a T5 model that takes the article, question, and the answer, and generates three distractors at once with a separator token [SEP] between them. For decoding, we use the Adam optimizer with a 5e-4 learning rate and Nucleus sampling (Top-p) [17] with a 0.9 P value. [11]

5) $T5_{disjoint}$:. Natural language generation tasks can use multiple correct references at once, as in MT [58], image captioning [22], and question generation [19] tasks. This T5 model is trained to generate one distractor at a time. Then, we apply a min-loss function following [19] work to generate several diverse distractors. During generation, we use the Diverse Beam Search sampling method [51] to generate three distractors for each input. Similar to the T5 model, we use Adam optimizer with a 5e-4 learning rate.

GDRCQ and BDG used an edited version of the RACE dataset for training and evaluation [25]. For the fairest comparison, we train all models on the original RACE dataset. Each model generates three distractors for a given context. Thus we feed the context to DISTO three separate times, once for each of the three generated distractors. We then average the DISTO scores. Once we have these models trained, we evaluate them using MT metrics BLEU, and BLEURT [47], the former of which is a learned SOTA MT metric.

In Table 7, under the "MT Evaluation" header, we present BLEU 1-4 and BLEURT results on the RACE test set. The results show that the BDG model clearly outperforms the other models considering each BLEU variant, except for BLEU-1. The $T5_{disjoint}$ model also performs well for the BLEU metrics. Similarly, for BLEURT, the BDG model performs best, followed by the GPT-2 model.

However, when we evaluate these models using DISTO, we see a very different story. In Table 7, under "Distractor Evaluation," we present the DG models' results using DISTO and several MT metrics. The Table also gives model rank based on BLEURT and DISTO scores (B-rank and D-rank respectively); note that DISTO gives a completely different ranking of models. In Figure 3 we can see that the two score types are actually negatively correlated (-0.69)! The model that performs the best in terms of BLEU and BLEURT metrics (BDG) has the lowest DISTO result, and the second best BLEURT model (GPT-2) performs the second worst model considering DISTO. Overall, the $T5_{disjoint}$ model performs the best. GDRCQ and T5 models have competitive performance with only 0.72% difference in DISTO scores. To summarize: when evaluating DG models, relying on MT metrics may be misleading.

5. CONCLUSION AND FUTURE WORK

The existence of distractor evaluation metrics is important for proper evaluation of new DG models. This importance is not limited to the final evaluation process. It is also essential to have an automated metric during the experimentation process to monitor model improvement at each stage, and to allow researchers to pick the best model during hyperparameter tuning. Incorrect or imprecise evaluations can

[9] To the best of our knowledge, these are the only models available online.

[10] We use the T5-base version from the Huggingface library.

[11] We tested Beam Search as well, but we found that Top-P gave better results for this model.

Table 7: Evaluation of DG models using BLEU, BLEURT, and DISTO. Best performance for each column is in bold, second best is underlined. Model ranks are given for BLEURT and DISTO (B-rank and D-rank respectively). See Figure 3 for another comparison of BLEURT vs. DISTO scores.

Model	MT Evaluation						Distractor Eval.	
	$BLEU_1$	$BLEU_2$	$BLEU_3$	$BLEU_4$	BLEURT	B-rank	DISTO	D-rank
GDRCQ	19.0	05.3	01.6	00.6	22.97	5	<u>82.47</u>	2
BDG	<u>30.2</u>	**18.9**	**13.0**	**08.9**	**31.90**	1	67.25	5
GPT-2	19.9	03.9	00.9	00.3	<u>31.07</u>	2	68.75	4
T5	25.1	08.4	02.7	00.9	30.10	3	81.75	3
$T5_{disjoint}$	**32.0**	<u>13.7</u>	<u>05.6</u>	<u>02.3</u>	26.42	4	**92.91**	1

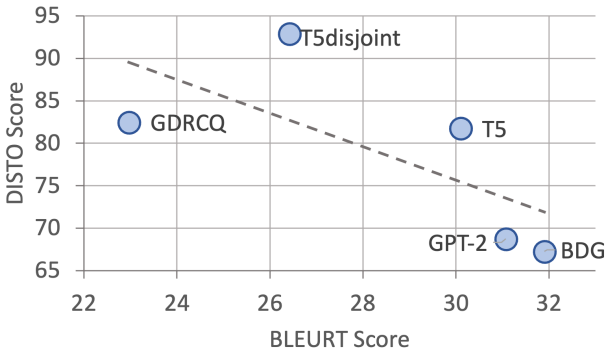

Figure 3: DISTO vs BLEURT scores for several DG models. A linear fit line is given, showing negative correlation (-0.69). This implies that MT metrics may not be a good metric for evaluating distractors.

lead to invalid results and the selection of a weak model for deployment.

Here, we studied the problem of textual distractor evaluation. We proposed DISTO, the first learned distractor evaluation metric. Unlike MT metrics that use a text-based comparison process to compare generated distractors to the gold ones, DISTO uses a negative sampling strategy with distractor augmentation techniques to model the characteristics of good and bad distractors within a given context.

We validated DISTO with extensive experiments coupled with a human evaluation. Our results show that DISTO is accurate and correlates highly with human ratings. Previous work that evaluated DG models using MT metrics may be less reliable, possibly leading to incorrect conclusions about the relative utility of a model. We plan to extend our work in two directions: First, we plan to integrate more sampling and augmentation techniques to cover more negative cases by including, e.g., grammatical modifications. Second, we will work to make DISTO multilingual to support the evaluation of distractors in other languages. We hope that DISTO and our exploration of distractor evaluation fosters new conversations in the DG community.

Limitations

In this work, we integrated several sampling and augmentation techniques that can help us to generate bad distractors. However, we assume that the current DG models generate grammatically correct distractors, thus we did not create augmented instances that cover grammatically incorrect distractors. We are not sure how DISTO will handle those cases in its current form.

Using both "Farthest Point in a Cluster" or "BERT [MASK] Filling" distractors augmentation techniques, we were able to create new bad distractors that are lexically modified. We found that these techniques are very effective to modify the original distractors in a way that the new distractors share some characteristics with the original ones, but at the same time, they are sometimes less contextually relevant. However, this was not always the case. We found in some cases that the two aforementioned techniques generate new good distractors. These good distractors might confuse our model since we assign low scores for them but they are contextually consistent. Finally, we want to highlight that those two techniques are computationally expensive, especially the "Farthest Point in a Cluster" technique.

Finally, we trained and evaluated DISTO on English only. Future work should consider distractor evaluation for different languages.

Ethics Statement

Human Annotation. We estimated the amount of time AMT workers need to finish a HIT and then we compensated them so that the payment rate was higher than the local living wage per hour. Each AMT worker received $0.4 USD for completing one HIT, which we estimated would take on average less than one minute.

Bias in Language Models. Language models have several types of bias, e.g. gender, race, religion, etc., and this is due to the data used to train them [27]. We acknowledge that the DISTO model we trained might cause ethical concerns, e.g. assigning a high score to biased distractors. We also acknowledge that DISTO is trained only on English, which disadvantages non-English speaking learners.

6. ACKNOWLEDGMENTS

This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), the Social Sciences and Humanities Research Council (SSHRC), the Digital Research Alliance of Canada (alliancecan.ca), and the Canadian Institute for Advanced Research (CIFAR). Alona Fyshe holds a Canada CIFAR AI Chair.

References

[1] I. Beltagy, M. E. Peters, and A. Cohan. Longformer: The Long-document Transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[2] J. Benesty, J. Chen, Y. Huang, and I. Cohen. Pearson Correlation Coefficient. In *Noise reduction in speech processing*, pages 1–4. Springer, 2009.

[3] J. Bulian, C. Buck, W. Gajewski, B. Boerschinger, and T. Schuster. Tomayto, Tomahto. Beyond Token-level Answer Equivalence for Question Answering Evaluation, Oct. 2022.

[4] H.-L. Chung, Y.-H. Chan, and Y.-C. Fan. A BERT-based Distractor Generation Scheme with Multi-tasking and Negative Answer Training Strategies. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4390–4400, 2020.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[6] R. Dijkstra, Z. Genç, S. Kayal, and J. Kamps. Reading Comprehension Quiz Generation using Generative Pre-trained Transformers. In *Proceedings of the Fourth International Workshop on Intelligent Textbooks 2022*, pages 4–17, 2022.

[7] J. Doughty, Z. Wan, A. Bompelli, J. Qayum, T. Wang, J. Zhang, Y. Zheng, A. Doyle, P. Sridhar, A. Agarwal, C. Bogart, E. Keylor, C. Kultur, J. Savelka, and M. Sakr. A Comparative Study of AI-Generated (GPT-4) and Human-crafted MCQs in Programming Education. In *Proceedings of the 26th Australasian Computing Education Conference*, pages 114–123, Jan. 2024.

[8] J. L. Fleiss. Measuring Nominal Scale Agreement Among Many Raters. *Psychological bulletin*, 76(5):378, 1971.

[9] Y. Gao, L. Bing, P. Li, I. King, and M. R. Lyu. Generating Distractors for Reading Comprehension Questions from Real Examinations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6423–6430, 2019.

[10] H. Gonçalo Oliveira, I. Caetano, R. Matos, and H. Amaro. Generating and Ranking Distractors for Multiple-Choice Questions in Portuguese. In *Workshop on Speech and Language Technology in Education*, pages 9 pages, 552283 bytes, Trinity College Dublin, Ireland, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[11] H. C. Goodrich. Distractor efficiency in foreign language testing. *Tesol Quarterly*, pages 69–78, 1977.

[12] Q. Guo, C. Kulkarni, A. Kittur, J. P. Bigham, and E. Brunskill. Questimator: Generating Knowledge Assessments for Arbitrary Topics. In *IJCAI-16: Proceedings of the AAAI Twenty-Fifth International Joint Conference on Artificial Intelligence*, 2016.

[13] L. A. Ha and V. Yaneva. Automatic Distractor Suggestion for Multiple-Choice Tests Using Concept Embeddings and Information Retrieval. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 389–398, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[14] A. Hadifar, S. K. Bitew, J. Deleu, C. Develder, and T. Demeester. EduQG: A Multi-Format Multiple-Choice Dataset for the Educational Domain. *IEEE Access*, 11:20885–20896, 2023.

[15] J. Hill and R. Simha. Automatic Generation of Context-based Fill-in-the-blank Exercises using Co-occurrence Likelihoods and Google N-grams. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 23–30, 2016.

[16] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural computation*, 9(8):1735–1780, 1997.

[17] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi. The Curious Case of Neural Text Degeneration. In *International Conference on Learning Representations*, pages 1–10, 2019.

[18] L. Huang, R. Le Bras, C. Bhagavatula, and Y. Choi. Cosmos QA: Machine Reading Comprehension with Contextual Commonsense Reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 2391–2401, 2019.

[19] X. Jia, W. Zhou, X. Sun, and Y. Wu. How to Ask Good Questions? Try to Leverage Paraphrases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6130–6140, Online, July 2020. Association for Computational Linguistics.

[20] S. Jiang and J. S. Lee. Distractor Generation for Chinese Fill-in-the-blank Items. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 143–148, 2017.

[21] D. Kalpakchi and J. Boye. BERT-based Distractor Generation for Swedish Reading Comprehension Questions using a Small-scale Dataset. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 387–403, 2021.

[22] A. Karpathy and L. Fei-Fei. Deep Visual-semantic Alignments for Generating Image Descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015.

[23] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[24] G. Kumar, R. E. Banchs, and L. F. D'Haro. Revup: Automatic Gap-fill Question Generation from Educational Texts. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 154–161, 2015.

[25] G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy. RACE: Large-scale Reading Comprehension Dataset From Examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, 2017.

[26] C. Liang, X. Yang, N. Dave, D. Wham, B. Pursel, and C. L. Giles. 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*, pages 284–290, 2018.

[27] P. P. Liang, C. Wu, L.-P. Morency, and R. Salakhutdinov. Towards Understanding and Mitigating Social Biases in Language Models. In *International Conference on Machine Learning*, pages 6565–6576. PMLR, 2021.

[28] C.-Y. Lin. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[29] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*, 2019.

[30] N. Mathur, J. Wei, M. Freitag, Q. Ma, and O. Bojar. Results of the WMT20 Metrics Shared Task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 688–725, 2020.

[31] K. K. Maurya and M. S. Desarkar. Learning to Distract: A Hierarchical Multi-Decoder Network for Automated Generation of Long Distractors for Multiple-Choice Questions for Reading Comprehension. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1115–1124, 2020.

[32] R. Mitkov et al. Computer-aided Generation of Multiple-Choice Tests. In *Proceedings of the HLT-NAACL 03 workshop on Building educational applications using natural language processing*, pages 17–22, 2003.

[33] J. Mueller and A. Thyagarajan. Siamese Recurrent Architectures for Learning Sentence Similarity. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1):1–10, Mar. 2016.

[34] J. Offerijns, S. Verberne, and T. Verhoef. Better Distractions: Transformer-based Distractor Generation and Multiple Choice Question Filtering. *arXiv preprint arXiv:2010.09598*, 2020.

[35] S. Ostermann, M. Roth, and M. Pinkal. MCScript2.0: A Machine Comprehension Corpus Focused on Script Events and Participants. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 103–117, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[36] S. Panda, F. P. Gomez, M. Flor, and A. Rozovskaya. 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*, pages 391–401, 2022.

[37] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA, 2002. Association for Computational Linguistics.

[38] J. Pennington, R. Socher, and C. D. Manning. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[39] V.-M. Pho, T. André, A.-L. Ligozat, B. Grau, G. Illouz, and T. François. Multiple Choice Question Corpus Analysis for Distractor Characterization. In *International Conference on Language Resources and Evaluation*, pages 1–10, 2014.

[40] V.-M. Pho, A.-L. Ligozat, and B. Grau. Distractor Quality Evaluation in Multiple Choice Questions. In *International Conference on Artificial Intelligence in Education*, pages 377–386. Springer, 2015.

[41] J. Pino and M. Eskenazi. Semi-automatic Generation of Cloze Question Distractors Effect of Students' l1. In *International Workshop on Speech and Language Technology in Education*, pages 65–68, 2009.

[42] Z. Qiu, X. Wu, and W. Fan. Automatic Distractor Generation for Multiple Choice Questions in Standard Tests. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2096–2106, 2020.

[43] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language Models are Unsupervised Multitask Learners. *OpenAI blog*, 1(8):1–9, 2019.

[44] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21:1–67, 2020.

[45] M. Richardson, C. J. Burges, and E. Renshaw. 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*, pages 193–203, 2013.

[46] A. Rogers, O. Kovaleva, M. Downey, and A. Rumshisky. Getting Closer to AI Complete Question Answering: A Set of Prerequisite Real Tasks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34-05, pages 8722–8731, 2020.

[47] T. Sellam, D. Das, and A. Parikh. BLEURT: Learning Robust Metrics for Text Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, 2020.

[48] K. Stasaski and M. A. Hearst. Multiple Choice Question Generation Utilizing an Ontology. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 303–312, 2017.

[49] K. Sun, D. Yu, J. Chen, D. Yu, Y. Choi, and C. Cardie. Dream: A Challenge Data Set and Models for Dialogue-based Reading Comprehension. *Transactions of the Association for Computational Linguistics*, 7:217–231, 2019.

[50] R. R. Torrealba, E. Garcia-Lopez, and A. Garcia-Cabot. End-to-End Generation of Multiple-Choice Questions using Text-to-Text Transfer Transformer Models. *Expert Systems with Applications*, 208:1–12, 2022.

[51] A. K. Vijayakumar, M. Cogswell, R. R. Selvaraju, Q. Sun, S. Lee, D. Crandall, and D. Batra. Diverse Beam Search: Decoding Diverse Solutions from Neural Sequence Models. *arXiv preprint arXiv:1610.02424*, 2016.

[52] J. Welbl, N. F. Liu, and M. Gardner. Crowdsourcing Multiple Choice Science Questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 94–106, 2017.

[53] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al. Huggingface's Transformers: State-of-the-art Natural Language Processing. *arXiv preprint arXiv:1910.03771*, 2019.

[54] J. Xie, N. Peng, Y. Cai, T. Wang, and Q. Huang. Diverse Distractor Generation for Constructing High-Quality Multiple Choice Questions. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:280–291, 2021.

[55] C. Y. Yeung, J. S. Lee, and B. K. Tsou. Difficulty-aware Distractor Generation for Gap-fill Items. In *Proceedings of the The 17th Annual Workshop of the Australasian Language Technology Association*, pages 159–164, 2019.

[56] T. Zesch and O. Melamud. 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*, pages 143–148, 2014.

[57] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi. BERTscore: Evaluating Text Generation with BERT. *arXiv preprint arXiv:1904.09675*, 2019.

[58] R. Zheng, M. Ma, and L. Huang. Multi-Reference Training with Pseudo-References for Neural Translation and Text Generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3188–3197, 2018.

[59] X. Zhou, S. Luo, and Y. Wu. Co-attention Hierarchical Network: Generating Coherent Long Distractors for Reading Comprehension. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34-05, pages 9725–9732, 2020.

APPENDIX

A. AMAZON MECHANICAL TURK (AMT) ANNOTATION INTERFACE

In Figure 4 we present a sample from the AMT interface.

B. NEGATIVE SAMPLING DATA SAMPLE

In Table 8, we present a sample from the human data annotation process (see Section 4.2) for distractors created using the "Farthest Point in a Cluster" ($\bigtriangledown$) and "BERT [MASK] Filling" ($\bigcirc$) techniques; we discard the other two techniques ("Answer Replication" and "Random Distractor") as they are straightforward.

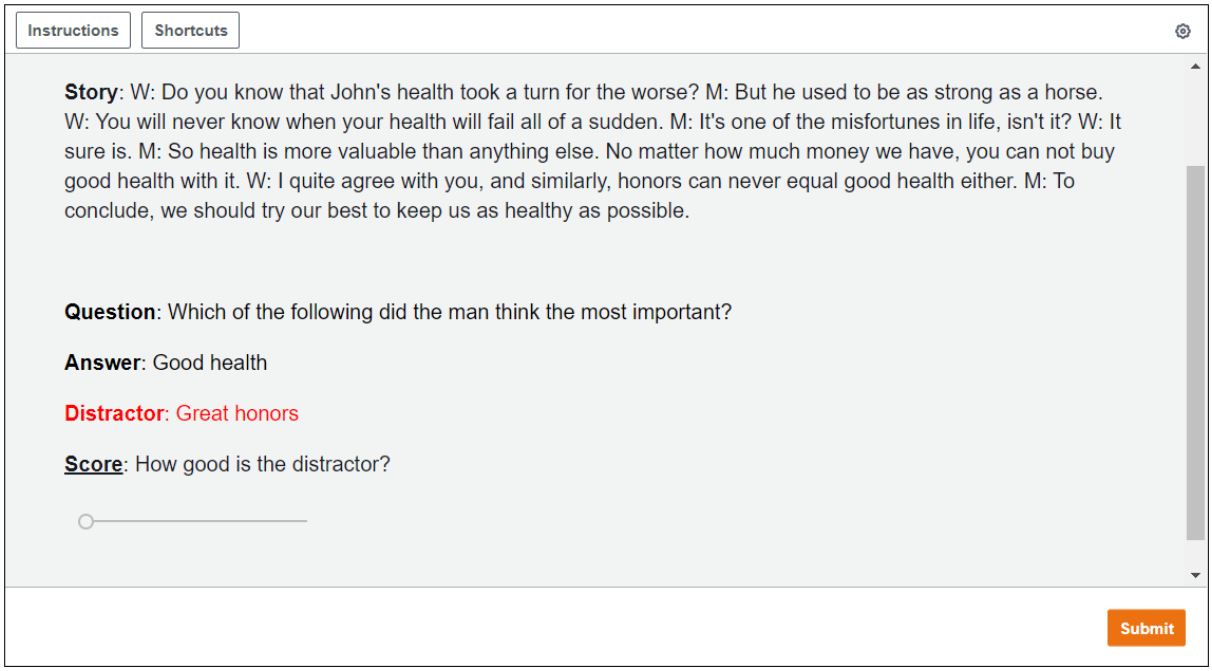

Figure 4: A sample from the AMT interface.

Table 8: A sample from the human annotation data for the negative sampling distractors (NS) with the original distractor (original) before applying the negative sampling technique. We including the question (Q) and the answer (Ans) for each instance for better understanding. "Farthest Point in a Cluster" (▽) and "BERT [MASK] Filling" (◯).

Context	Options	Technique
Q: When is the newspaper very close to their front porch? **Ans**: In the morning, every Sunday	**Original**: At noon and just mondays **NS**: At the end of the morning service	◯
Q: Who signed everyone up for salsa dancing lessons? **Ans**: Sharon or author or friends	**Original**: The teacher **NS**: The federal reserve	◯
Q: What do we know about the match? **Ans**: It can't be much fun	**Original**: It may be put off **NS**: It is part of the american museum of natural history	◯
Q: How many ways for studying does the passage tell us? **Ans**: Four	**Original**: Six **NS**: Miranda	◯
Q: Why did you start making stew ? **Ans**: I like chicken stew	**Original**: I was bored **NS**: He was from new england	◯
Q: Why was the bow put on? **Ans**: To decorate the gift	**Original**: To make the gift look bigger **NS**: Shouting them to dead	◯
Q: Who decided to leave the restaurant, besides the narrator? **Ans**: Their husband	**Original**: The band **NS**: The less farmland	◯
Q: Where did Rachel look at lost kitten ads? **Ans**: The internet	**Original**: The park **NS**: A 15th-century tower	◯
Q: When did they hit the on button? **Ans**: After filter basket and filling the carafe	**Original**: Before scooping the coffee **NS**: Before making the bed	▽
Q: What kind of person was she in her dream? **Ans**: She was not human, born from fire elemental of justice	**Original**: A normal person **NS**: A good sign	▽
Q: What was so necessary? **Ans**: Making a detailed shopping list	**Original**: Missing items from the store **NS**: The guy from the diner	▽
Q: What can we infer from the conversation? **Ans**: Joe probably failed in the exam	**Original**: The exam was easier than the previous one **NS**: The rest was different than the next time	▽
Q: Why did the narrator have a breakdown ? **Ans**: They didn't like being on the road	**Original**: They didn't like people from kansas **NS**: They didn't have money from him	▽
Q: Why were they gathered? **Ans**: For a birthday party	**Original**: For a dinner **NS**: For a week	▽
Q: When did they get a new bag out of their cabinet? **Ans**: After they took the old trash to the big garbage bin	**Original**: After they went to bed **NS**: After they retired to Germany	▽
Q: What does the man mean? **Ans**: He had a good time at the party	**Original**: He didn't enjoy the party at all **NS**: The study had been kept secret before finished	▽