None of the above: Comparing Scenarios for Answerability Detection in Question Answering Systems

Anonymous ACL submission

Abstract

Question Answering (QA) is widely used for evaluating the reasoning capabilities of NLP systems, where an important ability is to decide on answerability — whether the question can be answered with the information at hand. Previous works have studied answerability by including a fixed proportion of unanswerable questions in a collection, without explaining the reasons for such proportion or its impact on systems' results. In this work, we study different scenarios for answerability detection and evaluate several Large Language Models using different rates of unanswerable questions by introducing unanswerable questions in the popular multiple-choice QA dataset *RACE*. We show that a 30% rate of unanswerable questions at training seems optimal across a variety of scenarios, and support this with a series of extended experiments. Despite this, we observe that systems tend to expect the same rate of unanswerable questions seen at training and that the ability to decide on answerability always comes at the expense of the ability to find the answer when it exists.

1 Introduction

The last few years have shown an increase in performance for Natural Language technologies. One of the main reasons for this improvement is the development of systems based on transformer architectures (Vaswani et al., 2017), which are the predominant architectures of the current models (Devlin et al., 2019; Yang et al., 2019a). Researchers have proposed several tasks for evaluating systems' capabilities and Question Answering (QA) is used as a way of evaluating reasoning. In the QA task, a system must extract the span of text containing the correct answer to a question (extractive QA) or select the correct answer among a set of candidates (multiple-choice QA) (Rogers et al., 2020).

QA benchmarking has tried to evaluate different reasoning capabilities (Weston et al., 2016), helping to detect room for improvement. Most benchmarks are of general domain, using documents from Wikipedia, e.g. WikiQA (Yang et al., 2015) or news articles, e.g. NewsQA (Trischler et al., 2017); while others are domain-specific, e.g. on the biomedical domain (Tsatsaronis et al., 2015).

One important ability of QA systems is answerability, the ability to detect if a question has a correct answer, which is tested by including questions without a correct answer in the datasets (Rogers et al., 2022). The objective of testing answerability is to detect missing information[1]. If the answer is not contained in the reference document(s), assuming that a question is answerable leads to a wrong answer. These questions require, as it is mentioned in Rajpurkar et al. (2018), to "know what you don't know´´. Otherwise, a system can return a random answer, which could be correct. The inclusion of such questions lead to a slight drop in performance (about 30%), which was quickly overcome. The best examples of these benchmarks are SQuAD 2.0 (Rajpurkar et al., 2018) for extractive QA and QuAIL for multiple-choice QA (Rogers et al., 2020).

The distribution of questions without a correct answer usually ranks between 30-50% of the whole dataset, depending on the benchmark. However, it is unclear: (1) why the authors selected such distributions and (2) how the distribution affects results. In fact, it may remain open if systems are learning about answerability.

In this paper, we study answerability using different distributions of questions without correct answers. We firstly modify RACE (Lai et al., 2017), a well-known multiple-choice collection, and create several versions containing different distributions of questions without correct answers, from 0% to

[1]This is different to the option of not responding, where a system is unsure about its ability to answer a question and prefers not to answer instead giving an incorrect answer (Peñas and Rodrigo, 2011).

100% in 10% splits[2]. Then, we train and evaluate a system in all the versions, testing all the possible combinations. We observe the model tends to reproduce the distribution seen at training and to train the with a 30% of unanswerable questions seems to be the best strategy. But, we show that any training strategy including unanswerable questions reduces the performance when answering answerable questions. So, we hope our study promotes new proposals for improving systems' abilities to predict question answerability.

2 Related Work

Question Answering (QA) requires inferring the answer to a given question from a given context. This formulation can adopt different forms (Chen, 2018): the context can be a short paragraph or a document with several paragraphs; the notion of question can be expanded to a cloze-style (fill-in-the-gap) task (Hermann et al., 2015); and the task can involve extracting a span of text from the context (Joshi et al., 2017), choosing an answer among multiple options (Sugawara et al., 2018), or even generating a free-form answer (Nguyen et al., 2016).

One of the main challenges of constructing large-scale datasets is how to obtain the questions. Several datasets obtain questions from crowdsourcing. This hampers the applicability of experiments to real-world scenarios, where users information needs are spontaneous and unconstrained (Clark et al., 2019). One solution is to build benchmarks based on naturally occurring questions such as MS MARCO (Nguyen et al., 2016), NarrativeQA (Kočiskỳ et al., 2018) and Natural Questions (Kwiatkowski et al., 2019). However, it is more difficult and costly to create such collections.

As systems reach human performance on the most popular QA benchmarks, different strategies has been followed to create more difficult datasets. For example, the ARC dataset discards questions if they are too easy for a word co-occurrence algorithm (Clark et al., 2018), and ComQA (Abujabal et al., 2019) discards questions whose answer could be found by existing search engine technologies. Other datasets focus on specific types of reasoning, such as sorting data (Dua et al., 2019) or finding coreferences (Dasigi et al., 2019).

Lai et al. (2017) establish five levels of reasoning difficulty (in increasing order): word matching, paraphrasing, single-sentence reasoning, multiple-sentence reasoning and insufficient/ambiguous. These authors claimed that many questions in popular datasets like CNN (Chen et al., 2016) or SQuAD (Rajpurkar et al., 2016) are simple factoid questions, or they can be solved by simple word matching or paraphrasing. Single-sentence reasoning is easier than multi-sentence reasoning (Richardson et al., 2013), while integrating the information contained in multiple sentences is also much more difficult for humans (Berninger et al., 2011). A dataset that focuses on multi-sentence reasoning is MultiRC (Khashabi et al., 2018), and this concept is extended to long documents in NarrativeQA (Kočiskỳ et al., 2018), and multiple documents in HotpotQA (Yang et al., 2018). A comprehensive approach to several reasoning phenomena is QuAIL (Rogers et al., 2020), a multiple-choice QA dataset where questions are annotated by type of reasoning skill. QuAIL also includes unanswerable questions.

Datasets such as CNN/Daily Mail (Hermann et al., 2015), SQuAD (Rajpurkar et al., 2016) or RACE (Lai et al., 2017) were constructed with the assumption that a correct answer for every question exist within the given context. However, this assumption does not hold in real-world QA applications. For example, in web search there can be multiple possible sources of information (typically web snippets) that

Adding onto the previous version of SQuAD, SQuAD 2.0 (Rajpurkar et al., 2018) included more than 50k unanswerable questions written by crowdworkers. The premise was to add relevant questions with plausible (yet incorrect) answers within the given passage, but these questions were unanswerable based on the passage alone. Analyses showed that systems' performance is overestimated in the presence of unanswerable questions.

Adversarial training Several studies have used automatic adversarial methods to probe model robustness with similar conclusions. Jia and Liang (2017) showed how model performance on SQuAD degrades by more than half when tested over examples adversarially modified with their *AddSent* algorithm, which appends a sentence that resembles the question to the reference passage. However, the data generated are similar to the original, resulting in a less diverse test set. Wang and Bansal (2018) proposed an improved version of

[2]We release the script to create these versions in [ANONYMOUS IN THE REVIEW PERIOD]

the algorithm, *AddSentDiverse*, and an improved training regime including adversarial data augmentation. Gan and Ng (2019) proposed adversarial question paraphrasing to test models' reliance on string matching, and also applyed the method to create training data, improving models' robustness. Yang et al. (2019b) experimented over both SQuAD and RACE, but instead of corrupting the datasets they applyed adversarial perturbations at the level of word embeddings during training.

In contrast to SQuAD 2.0, these adversarial methods have the advantage of needing less human labour. However, they do not necessarily produce unanswerable questions. An exception is the work by Zhu et al. (2019), but the question variations produced are too lexically similar to the original ones and do not clarify whether the model fully understands them or relies on superficial cues.

Answer removal While the above adversarial methods work by producing modified questions for extractive QA, other dataset formats allow simpler methods. Pradel et al. (2020) showed an example of unanswerable question generation in Knowledge-Based QA. They modified the Spider KB question answering dataset by deliberately removing some information from the underlying relational databases. The present work follows a similar approach over the multiple-choice QA format.

Most of these studies add changes to collections or make them more difficult for evaluating reasoning capabilities. However, it is unclear in what grade these changes affect results or evaluate reasoning capabilities. Besides, the studies lack of notions about the proportion of changes that should be included in a new collection. In our study, we try to fill this gap regarding unanswerable questions.

3 Dataset

Our definition of unanswerable question is akin to the one seen in QuAIL (Rogers et al., 2020), where a question is annotated as unanswerable when the supporting passage does not provide sufficient information, and world knowledge does not make one of the answers more likely. With this definition in mind, we modify a collection without unanswerable questions, RACE, creating different splits with different distributions of unanswerable questions.

The original RACE dataset is a canonical benchmark in Multiple-Choice QA. RACE collects real English as a Second Language exams for 12- to 18-year-old students in China. The exams are intentionally designed by human experts to evaluate human language understanding and reasoning, which makes RACE an adequate tool to examine QA systems. The dataset is also large enough to allow the training of current data-driven technologies. The collected exams consist of a supporting passage accompanied by a variable number of questions about it. Each of these questions is paired with 4 candidate answers, of which only one is correct. A sample passage and two corresponding questions from RACE-M can be seen in Figure 3 in Appendix B.

The exams originate from either middle- (12 to 15 years old) or high-school (15 to 18) examinations, thus allowing the dataset to be separated in two levels of difficulty, which the authors denominate RACE-M and RACE-H respectively. There is a wide gap in difficulty; passages, questions and candidate answers in RACE-H are 52% longer on average, and contain a much wider vocabulary (125120 tokens in RACE-H vs. 32811 in RACE-M). The authors claim that, since both the questions and candidate answers are human generated, RACE is more challenging than comparable-scale QA datasets. To support this claim, they annotate a sample of questions with the type of reasoning phenomena involved. Their statistics show that 33% of the questions in RACE involve single-sentence reasoning and 26% multi-sentence reasoning, while a combined 37% can be solved with word matching or paraphrasing – this last figure is 74% for SquAD.

RACE contains a total of 27933 text passages with 97687 questions. The authors provide predefined train, validation and test splits. Tables 1 and 2 in Appendix A detail the numbers of passages and questions per difficulty level and split.

To render a question unanswerable, we simply replace the correct answer option with a sentence that implies that no answer exists among the given options, i.e. *None of the answers are correct.* Eliminating the correct answer turns a question unanswerable regardless of the type of reasoning involved. The remaining three options are plausible but incorrect, thus the only correct answer is *None of the answers are correct.*

To prevent model overfitting (i.e. that systems learn to identify *None of the answers are correct.* as the correct answer to any question) and again following QuAIL, we also introduce the "unanswerable" option in questions that should remain

answerable. In these cases, we replace one of the incorrect options chosen at random, keeping the correct answer choice available, but at the same time introducing a different kind of distractor: one that indicates that the question may be unanswerable given its particular context and the other answer choices. We give an example of the original passage and questions turned into the new ones in Figures 3 and Figure 4 in Appendix B.

We create a series of altered versions of the original dataset to simulate scenarios with a different, measurable occurrence of unanswerable questions. For every version, we apply the modification procedure described above. A parameter C governs the rate of unanswerable questions in each version, and thus the probability of eliminating (replacing) the correct answer choice. We divide the dataset by split and difficulty level, and apply the parameter to each group separately, choosing $C \times N$ examples at random, where N is the number of questions in a particular difficulty level and split. On these chosen instances, we replace the correct answer by *None of the answers are correct.*. On the rest of the instances, we preserve the correct answer and replace an incorrect candidate at random. For test splits, the process is repeated 5 times, creating 5 test splits per dataset with differently altered instances.

The value of the parameter C is in the range $[0-1]$, where 0 indicates that the replaced option will always be an incorrect one and therefore all questions remain answerable, and 1 indicates that for all questions the correct option will be replaced, producing a scenario where all questions become unanswerable. Intuitively, these extreme scenarios are senseless, and we expect the middle values of C to produce the interesting results. Still, the aim of the experiment is to compare all possible scenarios. We give C the whole range of values $[0-1]$ in steps of 0.1, producing 11 modified copies of RACE with proportions of 0%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, and 100% of unanswerable questions.

4 Main experiment

We use the English BERT-base[3] model from huggingface (Wolf et al., 2019) in a Google Colab[4] instance with 8 TPUv2 cores. Furthermore, we make use of the PyTorch framework (Paszke et al., 2019) and the huggingface's Datasets (Lhoest et al., 2021) library.

On each of the 11 datasets we fine-tune a pre-trained BERT model with the same hyperparameters. We fine-tune on the train splits of both RACE-M and RACE-H. We give more details of pre-processing and training in Appendix C.

Each of these trained models is evaluated on 11 evaluation datasets. For each dataset, the model is evaluated on 5 different test splits (with the same distribution of unanswerable questions, but the set of questions turned unanswerable is different). We obtain results separately for each of these 5 test splits, and then average results by dataset. We use accuracy, which measures the proportion of correct answers, and is the common metric in multiple-choice QA.

5 Results

We show our results using (11×11) heatmap matrixes that relates the 11 models (each trained with a different version of the *RACE* dataset) with the 11 test sets. We intend to compare all the results in a single overview. Columns represent the 11 trained models ordered by the percentage of unanswerable questions on the training set, and rows represent the 11 test sets — also ordered by the percentage of unanswerable questions in them. In this setup, a cell contains the results of a particular model over a particular test set. For instance, on Figure 1, cell $(4, 2)$[5] contains the accuracy (0.54) obtained after evaluating on a test set with 30% of unanswerable questions, a model trained on a set with 10% of unanswerable questions.

We shall see that some cells are light grey in some matrixes focused on showing answerability. This indicates that their value is undetermined because it is caused by a zero division. For example, if we look at the results on unanswerable questions (Figure 7 in Appendix D), the bottom row is greyed because it represents a test set with zero unanswerable questions and, therefore, calculating results here involves a zero division.

We show column and row averages, respectively, at the upper and right ends of every heatmap. Column averages contain an overview of the same model across multiple testing scenarios, while row averages summarize the difficulty of a test set for different models.

[3] https://huggingface.co/bert-base-uncased

[4] https://colab.research.google.com

[5] Row tagged as 30%, column tagged as 10%

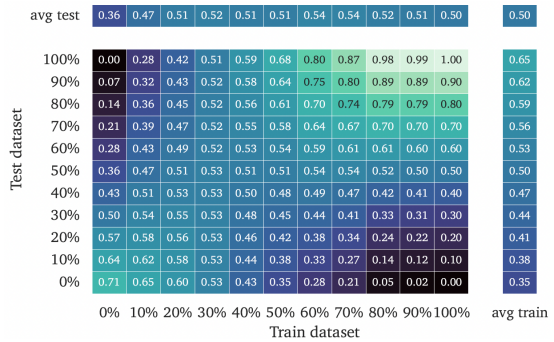

Figure 1: Model accuracy on modified *RACE* test sets

We have evaluated the models on (modified versions of) *RACE high* and *RACE middle* separately (models were trained on both), but we have aggregated the results. These results over (a version of) *RACE middle* are always better than those over (the corresponding version of) *RACE high*. However, we have dismissed these differences because they are always similar and we want to focus on comparing training strategies. For a breakdown by difficulty level, please see Appendix F.

5.1 Overall Results

We show the values of accuracy for each combination of model and test set in Figure 1. The bottom left cell displays the accuracy of a BERT model that has seen a 0% of unanswerable questions, neither during training nor during evaluation, which corresponds to the original RACE collection. The general observation on this table comes from looking at the diagonal starting from the bottom-left: a model's accuracy is better when it is tested on a dataset with an proportion of unanswerable questions similar to the dataset on which it was trained.

For models trained on datasets with a high proportion (80-100%) of unanswerable questions, the accuracy on any particular test set almost matches the amount of unanswerable questions in that set. This suggests that these models have learnt to identify the "unanswerable" option as the correct answer, and they fail to discern the small percentage of truly answerable questions.

To further break down these results, we have split the accuracy for each group of questions: answerable and unanswerable. Figure 6 (Appendix D) shows the general accuracy when only taking answerable questions into account. Here, we observe that model accuracy remains relatively constant across test sets (i.e. by column), but declines rapidly across models as the percentage of unanswerable questions seen in training rises (i.e. towards the right side of the table). The reason for this is predictions are independent of each other, thus when only looking at answerable questions, the number of unanswerable questions in a test set does not matter. What we are looking at here is each model's ability to correctly answer answerable questions and this ability is severely impacted by the presence of unanswerable questions in training. In fact, models trained on over 80% of unanswerable questions are almost completely unable to give proper answers.

We show in Figure 7 (Appendix D) the accuracy on unanswerable questions, where we can see the reverse pattern: models trained on a high proportion of unanswerable question can reliably detect them. The models that saw over 80% of unanswerable questions in training can almost detect all of them, but as we saw earlier (Figure 6) this is at the expense of the ability to deal with answerable questions. On the other hand, on the left-most column we see that the model that saw 0% of unanswerable questions in training never detects them. However, the model that saw only 10% of unanswerable questions in training does show a certain ability to detect them above expectations (though still unreliable). But as we saw on Figure 6, this comes at the expense of the capacity to deal with answerable questions.

Looking at the average accuracy over different datasets, we see that the model trained on 60% of unanswerable questions has the highest average accuracy of all models over all modified versions of *RACE*.

5.2 Answerability

To focus on answerability, we have dismissed the answer given to answerable questions, paying attention only to whether the system identifies unanswerable questions. So, we convert into a binary response the model's responses. That is, instead of *A*, *B*, *C* or *D*, we interpret the model's responses as *unanswerable* or *answerable*. A model decides a question is *unanswerable* when it chooses the option that contains *"None of the answers are correct."* and decides the question is *answerable* when it chooses any of the other 3 answers. In this way, we switch the problem from identifying the right answer to recognizing if the question is answerable given the candidates. Note that for the calculation of subsequent metrics, we consider *unanswerable*

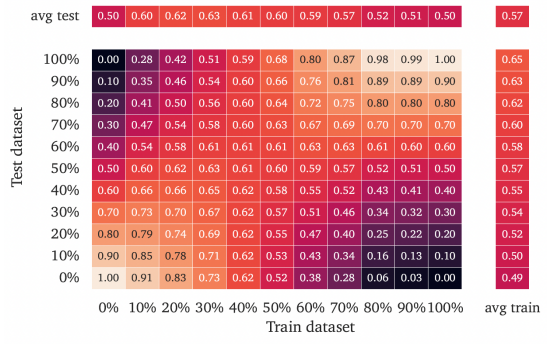

Figure 2: Answerability accuracy, i.e. accuracy at unanswerable question detection.

as the positive class.

We define answerability accuracy as:

$$\frac{\begin{array}{r}|\text{ unanswerable} \wedge \text{pred. unanswerable}| \\ +|\text{ answerable} \wedge \text{pred. other}|\end{array}}{N} \quad (1)$$

We show in Figure 2 the heatmap matrix for answerability accuracy. In this Figure, we observe that answerability accuracy has a distribution pattern similar to the general accuracy (shown in Figure 1). However, values towards the lower left corner of the table are higher in this case, indicating that models hardly ever choose the unanswerable option when it was hardly seen in training. We can see this on Figure 5 (Appendix D), which shows the proportion of unanswered options given by the models. In fact, this Figure shows how the model select the unanswered option in a similar proportion to the already seen at training.

The retrieval of answerable questions, or *specificity*, is shown in Figure 8 (Appendix D) and yields a pattern similar to the one seen on Figure 6. Values here are generally higher, indicating that models that saw few unanswerable questions in training tend to fail by choosing "proper" but incorrect answers, not by choosing the *unanswerable* option.

5.3 Comparing Results on Imbalanced Datasets

In this work, we have compared the results of testing a series of models on a series of imbalanced datasets (the proportion of answerable and unanswerable questions differ). While the datasets are (deliberately) imbalanced, we hypothesize that retrieving one class is as important as retrieving the other. In such a situation, the ideal scenario is a combination of model and test set that yields good accuracy over the two classes. But so far, the results indicate that the ability to retrieve one class is detrimental to the ability to retrieve the other. Thus, we need a metric that takes into account accuracy scores on each of the two classes at the same time. To that end, we propose to use **Youden's J statistic** or Youden's index (Youden, 1950), defined as:

$$J = \text{recall} + \text{specificity} - 1$$

Youden's J statistic is a measure of informedness that gives equal weight to the two types of error: false negatives (unanswerable questions for which the system chooses a "proper" answer) and false positives (answerable questions for which the system chooses "None of the answers are correct."). It produces values in the range $[0 - 1]$ (by definition $[-1 - 1]$, but a negative value can be corrected by switching the classes), and it can be seen as a linear transformation of the *balanced accuracy* (the arithmetic mean of recall and specificity). We have chosen Youden's J statistic over balanced accuracy because it produces a wider range of values.

We show results according to Youden's J statistic in Figure 9 (Appendix D). The values in Figure 9 reveal that both Figure 1 and Figure 2 are too optimistic. As we have seen above, a model's accuracy is generally good on test sets that are similar to the one the model was trained on, which leads to good values towards the lower left and upper right corners of the tables — where train and test sets have little uncertainty concerning answerability and also match. We see a different behaviour in Figure 9: as expected, the values on the leftmost and three rightmost columns are almost 0, again confirming that the respective models only have predictive capability for the class they have seen most. Results are not much better towards the centre of the table, and no value reaches 0.5, indicating all models' poor informedness concerning answerability. However, we see that the "30%" model is clearly better informed than the others.

Although our results generally speak of a big trade-off between recognizing answerability and correctly answering abilities, and do not allow us to prescript any particular training regime, the 30% of unanswerable questions in training (the proportion used in several collections) could be an interesting proportion in combination with the proposals we discuss in Section 7.

6 Additional Experiments

In this Section, we describe the results of additional experiments on answerability. We have modified several dimensions of the previous experiments to study their impact on results and, therefore, to learn more about the behavior of current technologies in scenarios where there are questions without correct answers. In the following subsections, we describe these experiments:

6.1 Testing other LLMs

In Section 4, we only used BERT-base for our experiments, what could narrow our conclusions to this model. This is why, in this section, we have performed the same experiments with two additional models: DeBERTa and T5.

DeBERTa (Decoding-enhanced BERT with Disentangled Attention) improves on BERT's attention mechanism by representing the word's content and position separately and by incorporating absolute positions in the decoding layer (He et al., 2020). We use DeBERTaV3, which also uses a more efficient pre-training task, and replaced token detection, instead of the usual mask language modeling. Besides, this model has obtained better results than BERT on several tasks. We fine-tune DeBERTaV3-base models on the 11 modified versions of RACE previously described, with the same learning rate and batch size as in the BERT experiment.

T5 is an encoder-decoder model which has to be used using text-to-text format (Raffel et al., 2020). We have selected T5 because it uses a different architecture than the other two models. For our purposes, we have used Flan-T5 (Chung et al., 2022), which uses a pre-training method based on prompting and has given us better results. We use a learning rate of 0.0001 and a batch size of 8. The details about the changes on the input are given in Appendix E.

We show results using DeBERTa in Figures 17, 20 and 23, and results using T5 in Figures 18, 21 and 24 in Appendix G.1. As expected, the results using DeBERTa are superior across all training-test combinations. Although in this case, it is the model trained on a 40% rate of unanswerable questions that performs best on average, the observed pattern is overall similar to the results using BERT. Regarding T5, the results are inferior to the ones obtained with the other two models, however, they display a similar pattern. Therefore, results on answerability seem to be not connected to the model used in the experiments. This is why we perform the following experiments using only BERT.

6.2 Augmenting the dataset

In the main experiment, we modified the dataset by replacement. That is, every instance of the RACE dataset was either made unanswerable or had one of the incorrect answers replaced by a distractor. In this experiment, we have augmented the training datasets by adding the modified instances to the originals, effectively having two versions of every question. Thus, models have more information for learning when a question cannot be answered with the given options.

We show the results of this experiment in Figures 25-30 in Appendix G.2. In the Figures with results using augmented data, the rates indicate the proportion of instances where the modified version is unanswerable.

In these Figures, we can see how the overall accuracy per column is slightly better when using the augmented dataset. However, according to the Figures with answerability accuracy, models do not improve their performance when deciding if they have to answer or not the question. Thus, it seems that the improvement in the overall accuracy is due to having more training data, despite the fact they are duplicates. According to these results, we think that the models are unable to learn the patterns that make a question unanswerable from the augmented datasets. In fact, the pattern observed is the same as in the original experiment.

6.3 Fewer Answer Options

As we have already pointed out, the distribution of questions without correct answers usually ranks between 30-50% in other datasets. This corresponds with the rates where we obtain the best results in our main experiment. Given that we have four options per question and therefore, each option has a probability of 0.25 of being correct, a rate of 30% seems to be natural. Therefore, the number of options could be another variable that affects answerability. To study the effect on results of the number of available options, we have changed the previously modified datasets (with different rates of unanswerable questions) and created one set where we remove one option per instance (three options remain) and another set where we remove two options per instance (two options remain).

We show the results of this experiment in Figures 31-39 in Appendix G.3. In general, the fewer

the options, the better the accuracy because the probability of finding the correct answer is higher. However, the patterns are similar to what we obtained in the original experiment. Thus, it seems that the number of options does not affect the proportion of questions without correct answers.

7 Discussion

Our experimental results indicate a strong preference for certainty regarding answerability, but not a clear path on how to deal with uncertainty regarding answerability, the main aim of the study. Models only obtain strong results when dealing with datasets that: *a)* were similar to the ones they had been trained on and *b)* contained a very low or very high number of unanswerable questions. Models were mostly unable to deal with distractors and only reproduced training bias.

Youden's J statistic (see Figure 9) has revealed that a proportion of 30% of unanswerable questions during training yields the most informed system, but this informedness always emerges at the expense of the ability to correctly answer genuinely answerable questions (see Figure 6). Once training includes unanswerable questions, at any rate, the general performance of the system decreases. This is why, at this point, we cannot recommend this setup, even with the settings that generate the most informedness model. We have observed the same behavior no matter the number of available options per question.

We can try to establish a pattern regarding if a system needs to see a higher proportion of unanswerable questions in training to identify them in test. In fact, looking at Figure 1 by row we see that in an evaluation scenario with 50% of unanswerable questions, the amount of them seen in training does not matter as long as there are some of them (over a 10%). For scenarios with less than 50% of unanswerable questions (presumably more likely), it is better to use models that saw a lower proportion. If we relax the criteria and look only at answerability detection (Figure 2), the evaluation scenario with a 50% of unanswerable questions is also better handled by models that saw a lower proportion during training. On the other hand, in scenarios with more than a 50% of unanswerable questions, it is better to train with a higher proportion of unanswerable questions. Therefore, the proportion of unanswerable questions a model should see during training largely depends on the end application.

Our results show that the models generally benefit from biased training. However, if we pay attention to the performance separately in each class, the ability to detect answerability or to correctly answer answerable questions remains constant across different scenarios. There is a trade-off between the two abilities which appears in any scenario, but while a model's performance depends on the evaluation scenario being biased in the same direction as the model, the model's informedness stays the same. Therefore, we would advise that it is unnecessary to test models in different scenarios regarding answerability. A single scenario with 10–50% of unanswerable questions, which matches what is proposed in other literature, would suffice.

8 Conclusions and Future Work

In this paper, we have studied different scenarios for testing answerability, the ability to detect unanswerable questions, in multiple-choice Question Answering (QA). Previous studies have tested answerability including a fixed proportion of unanswerable questions between 10-50% without explaining the reasons for such proportions or analyzing how it affects systems' results. So, we have used different distributions of unanswerable questions for both training and testing.

We have seen how systems tend to reproduce the distribution seen at training. That is, systems select that a question is unanswerable in the same proportion seen at training, no matter the distribution in the test collection. However, when systems improve answerability detection, they reduce their ability to correctly answer genuinely answerable questions, which is an unexpected and undesired behavior. So, further research should achieve a scenario where unanswerable questions can be recognized to a significant extent without harming the system's ability to answer answerable questions.

It remains unclear what makes a question unanswerable. In multiple-choice QA, a system is right selecting the option "None of the above" or "None of the answers are correct" (depending on how this option is introduced in the dataset), but we do not know if the system understands what this option means or if it truly detects that there is no correct answer. Hence, further research should also be oriented in this line, by studying the main features of unanswerable questions and how systems behave with these questions.

Limitations

This study is only applicable to multiple-choice QA, by introducing the option "None of the answers are correct" in datasets. For extractive QA, where the correct answer to unanswerable questions is an empty text span, systems could develop a different strategy for answering these questions and behave different when changing the distribution of unanswerable questions. In fact, as we have already discussed in Sections 7 and 8, it is unclear what makes a question unanswerable beyond lack of information in the source text. Besides, results could be biased for other reasons different from the unanswerability introduced in the modified collections.

On the other hand, we base results on the accuracy achieved by BERT, DeBERTa and T5 models. Although other transformer-based models should behave similarly, different technologies might show different results and abilities.

While RACE is a good QA benchmark, it was created for human evaluation. So, other collections created, for example, by crowd-sourcing, could be easier and then, systems may have a better ability detecting unanswerable questions.

References

Abdalghani Abujabal, Rishiraj Saha Roy, Mohamed Yahya, and Gerhard Weikum. 2019. Comqa: A community-sourced dataset for complex factoid question answering with paraphrase clusters. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 307–317.

Virginia W Berninger, William Nagy, and Scott Beers. 2011. Child writers' construction and reconstruction of single sentences and construction of multi-sentence texts: Contributions of syntax and transcription to translation. *Reading and writing*, 24(2):151–182.

Danqi Chen. 2018. *Neural reading comprehension and beyond*. Ph.D. thesis.

Danqi Chen, Jason Bolton, and Christopher D Manning. 2016. A thorough examination of the cnn/daily mail reading comprehension task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2358–2367.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Pradeep Dasigi, Nelson F Liu, Ana Marasović, Noah A Smith, and Matt Gardner. 2019. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. 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)*, pages 5925–5932.

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

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378.

Wee Chung Gan and Hwee Tou Ng. 2019. Improving the robustness of question answering systems to question paraphrasing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6065–6075.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman,

and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262.

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, et al. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. In *CoCo@ NIPS*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Anselmo Peñas and Álvaro Rodrigo. 2011. A simple measure to assess non-response. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 1415–1424. The Association for Computer Linguistics.

Camille Pradel, Damien Sileo, Álvaro Rodrigo, Anselmo Peñas, and Eneko Agirre. 2020. Question answering when knowledge bases are incomplete. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 43–54. Springer.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Matthew Richardson, Christopher JC Burges, and Erin Renshaw. 2013. 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.

Anna Rogers, Matt Gardner, and Isabelle Augenstein. 2022. Qa dataset explosion: A taxonomy of nlp resources for question answering and reading comprehension. *ACM Comput. Surv.* Just Accepted.

Anna Rogers, Olga Kovaleva, Matthew Downey, and Anna Rumshisky. 2020. Getting closer to ai complete question answering: A set of prerequisite real tasks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8722–8731.

Saku Sugawara, Kentaro Inui, Satoshi Sekine, and Akiko Aizawa. 2018. What makes reading comprehension questions easier? In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4208–4219.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200,

Vancouver, Canada. Association for Computational Linguistics.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael Alvers, Dirk Weißenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, Yannis Almirantis, John Pavlopoulos, Nicolas Baskiotis, Patrick Gallinari, Thierry Artieres, Axel-Cyrille Ngonga Ngomo, Norman Heino, Eric Gaussier, Liliana Barrio-Alvers, and Georgios Paliouras. 2015. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics*, 16:138.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Yicheng Wang and Mohit Bansal. 2018. Robust machine comprehension models via adversarial training. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 575–581, New Orleans, Louisiana. Association for Computational Linguistics.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M. Rush, Bart Van Merriënboer, Armand Joulin, and Tomas Mikolov. 2016. Towards ai-complete question answering: A set of prerequisite toy tasks.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, Lisbon, Portugal. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019a. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

Ziqing Yang, Yiming Cui, Wanxiang Che, Ting Liu, Shijin Wang, and Guoping Hu. 2019b. Improving machine reading comprehension via adversarial training. *arXiv preprint arXiv:1911.03614*.

William J Youden. 1950. Index for rating diagnostic tests. *Cancer*, 3(1):32–35.

Haichao Zhu, Li Dong, Furu Wei, Wenhui Wang, Bing Qin, and Ting Liu. 2019. Learning to ask unanswerable questions for machine reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4238–4248.

A RACE Distribution

division	train	validation	test	all
RACE-H	18728	1021	1045	20794
RACE-M	6409	368	362	7139
all	25137	1389	1407	27933

Table 1: Number of passages per difficulty level and split in RACE.

division	train	validation	test	all
RACE-H	62445	3451	3498	69394
RACE-M	25421	1436	1436	28293
all	87866	4887	4934	97687

Table 2: Total number of questions per difficulty level and split in RACE.

B Examples of the Datasets

A question modified to be unanswerable:

1) It took Mark _ to run the mile.
A. None of the answers are correct.
B. more than 13 minutes
C. only 12 minutes
D. less than 12 minutes

A question modified to remain answerable:

1) Why did Mark cry when he ran the last lap?
A. Because he was quite happy.
B. Because he was too upset.
C. Because he got a pain in his heart.
D. None of the answers are correct.

Figure 4: Modified sample questions from RACE. Correct answer in bold.

C Pre-processing and Training Details

A large portion of questions in RACE are not proper questions but cloze tasks, where a gap in a sentence must be filled with a word or short span of words. Candidate answers to cloze tasks usually do not constitute fully formed sentences. We identify cloze tasks by the character "_", used to signal the gap to be filled. By contrast, proper questions usually contain the character "?". We count the questions containing "_" and/or "?" (see Table 3) and manually examine questions that contain both or none and their corresponding answers, deciding to treat all questions containing "_" as cloze tasks and questions not containing that character as proper questions.

Passage:

In my second year of high school, the class was scheduled to run the mile. when the coach yelled, "Ready. Set. Go!", I rushed out like an airplane, faster than anyone else for the first 20 feet. I made up my mind to finish first. As we came around the first of four laps, there were students all over the track. By the end of the second lap, many of the students had already stopped. They had given up and were on the ground breathing heavily. As I started the third lap, only a few of my classmates were on the track. By the time I hit the fourth lap, I was alone. Then it hit me that nobody had given up. Instead, everyone had already finished. As I ran that last lap, I cried. And 12 minutes, 42 seconds after starting, I crossed the finishing line. I fell to the ground. I was very upset.
Suddenly my coach ran up to me and picked me up, yelling, "You did it. Mark! You finished, son. You finished" He looked at me straight in the eyes, waving a piece of paper in his hand. It was my goal for the day which I had forgotten. I had given it to him before class. He read it aloud to everyone. It simply said, "I, Mark Brown, will finish the mile run tomorrow, come what may." My heart lifted. My tears went away, and I had a smile on my face as if I had eaten a banana. My classmates clapped. It was then I realized winning isn't always finishing first. Sometimes winning is just finishing.

Questions (correct answer in **bold**):

1) It took Mark _ to run the mile.
A. about 13 minutes
B. more than 13 minutes
C. only 12 minutes
D. less than 12 minutes

1) Why did Mark cry when he ran the last lap?
A. Because he was quite happy.
B. Because he was too upset.
C. Because he got a pain in his heart.
D. Because he was hungry.

Figure 3: Original sample passage and corresponding questions from RACE.

BERT needs to be fed with sequences of sentences separated by a special token, `[CLS]`. Thus, to feed the model we need to transform the dataset's instances from a set `{passage, question, 4 options, answer}` to a sequence. The generation of this sequence depends on the type of questions. For proper questions, the resulting sequence has three items, of the form `[passage, question, option]`. By contrast, for cloze tasks we substitute the answer within the question, obtaining a sequence with two items of the form `[passage, question+option]`. For the answer option that has been replaced, "proper question" instances have the form `[passage, question, None of the answers are correct.]`, while cloze tasks have the form `[passage, None of the answers are correct.]`

level	split	contains "?"	contains "_"	contains "?" "_"	neither "?" nor "_"
RACE-H	train	29438	31340	557	1110
	validation	1610	1737	33	71
	test	1588	1815	33	62
RACE-M	train	10965	13629	549	278
	validation	620	771	34	11
	test	617	774	18	27

Table 3: Number of questions in RACE, per difficulty level and split, containing the characters "_" and/or "?"

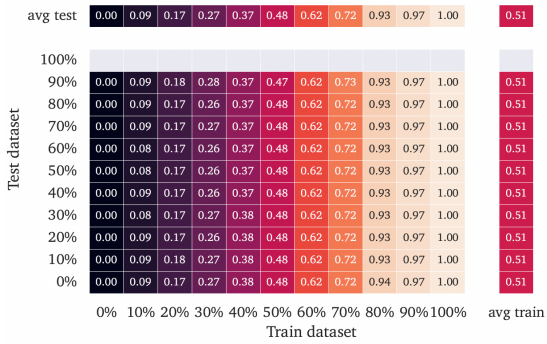

Figure 5: Proportion of times the "unanswerable" option is chosen.

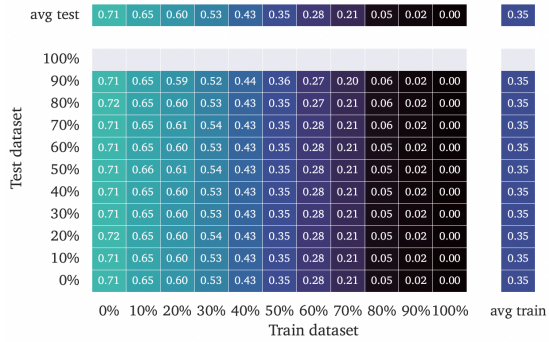

Figure 6: Model accuracy on modified *RACE* test sets when taking only answerable questions into account.

D Additional Results of the Main Experiment

E Input to T5 model

We have changed the input to the model with respect to BERT and DeBERTa, converting the task into a text-to-text format by substituting each instance's `question`, `options`, and `article` into the following template:

```
Question: {question}
Options:
Option A: {option A}
Option B: {option B}
```

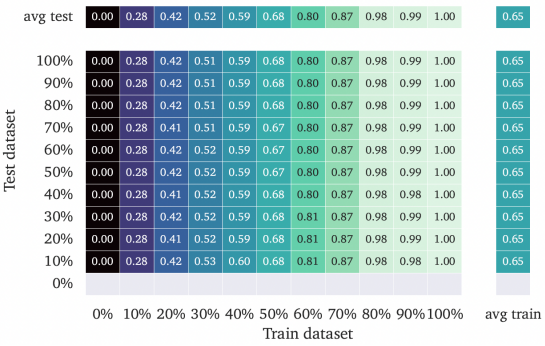

Figure 7: Accuracy on unanswerable questions only, or recall, on modified *RACE* test sets.

Figure 8: Specificity at unanswerable question detection.

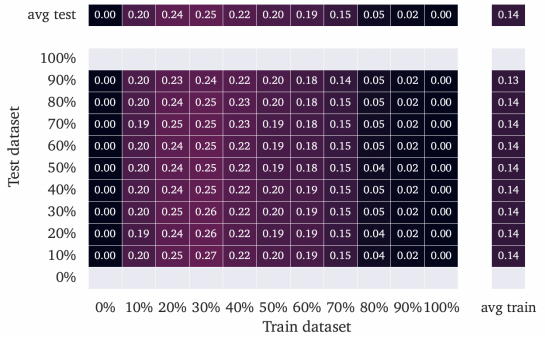

Figure 9: Youden's J statistic in terms of answerability.

```
Option␣C:␣{option␣C}
Option␣D:␣{option␣D}
Context:␣{article}
Answer:
```

F Comparison of results by level of difficulty

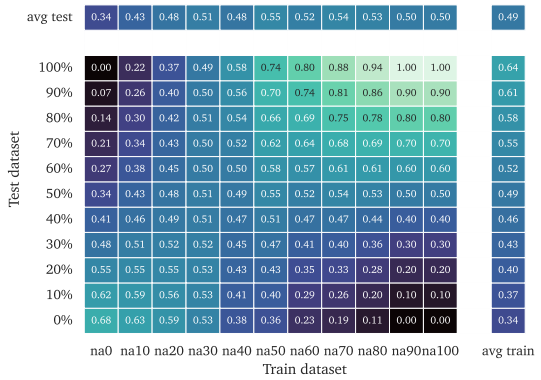

Figure 10: General accuracy on *high* test set.

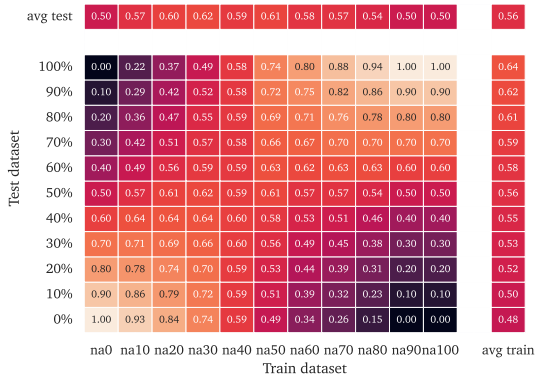

Figure 11: Answerability accuracy on *high* test set.

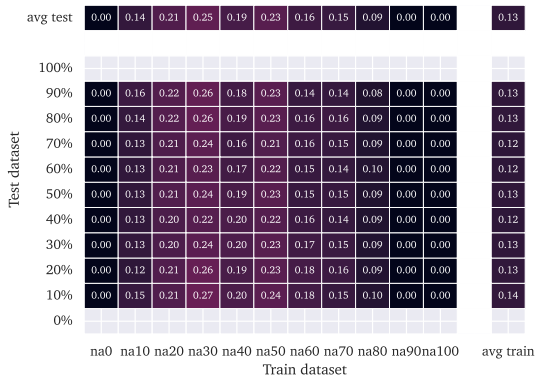

Figure 12: Youden's J statistic on *high* test set.

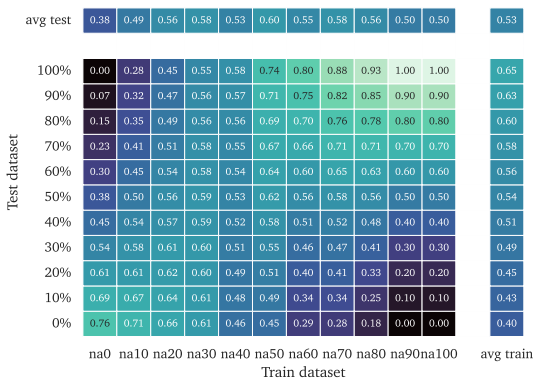

Figure 13: General accuracy on *middle* test set.

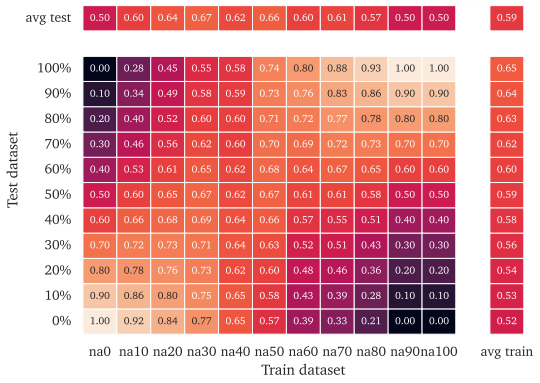

Figure 14: Answerability accuracy on *middle* test set.

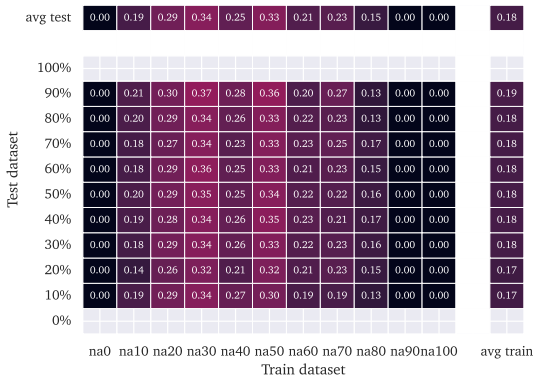

Figure 15: Youden's J statistic on *middle* test set.

G Comparison of results by experiment

G.1 Results by model

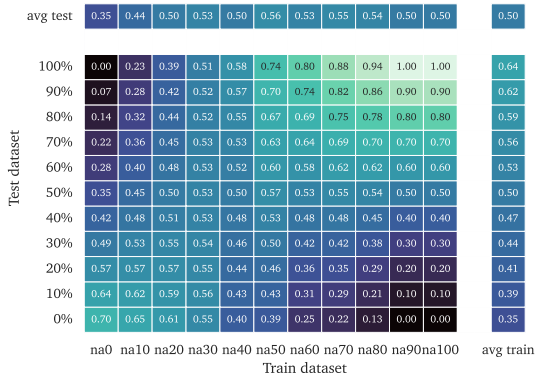

Figure 16: General accuracy of models based on BERT

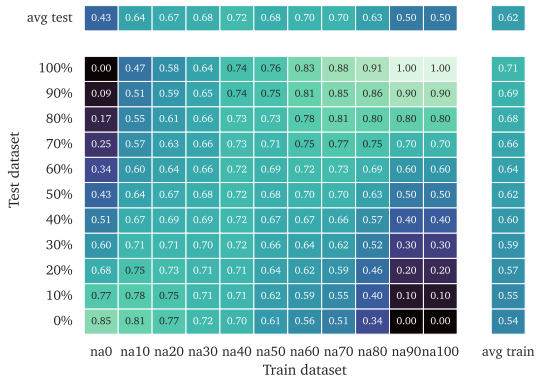

Figure 17: General accuracy of models based on DeBERTa

Figure 18: General accuracy of models based on T5

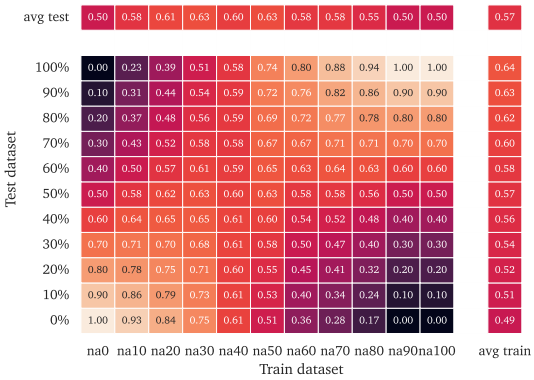

Figure 19: Answerability accuracy of models based on BERT

Figure 20: Answerability accuracy of models based on DeBERTa

Figure 21: Answerability accuracy of models based on T5

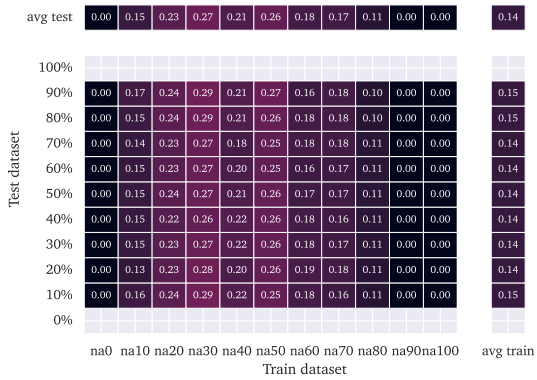

Figure 22: Youden's J statistic of models based on BERT

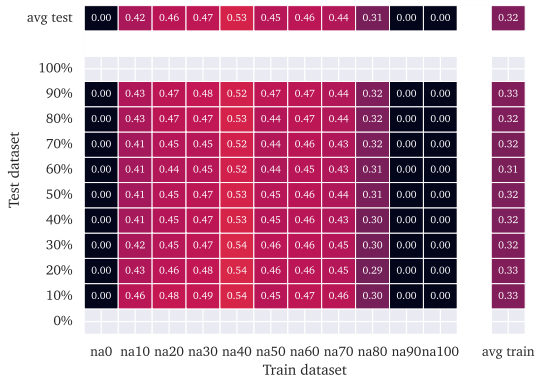

Figure 23: Youden's J statistic of models based on DeBERTa

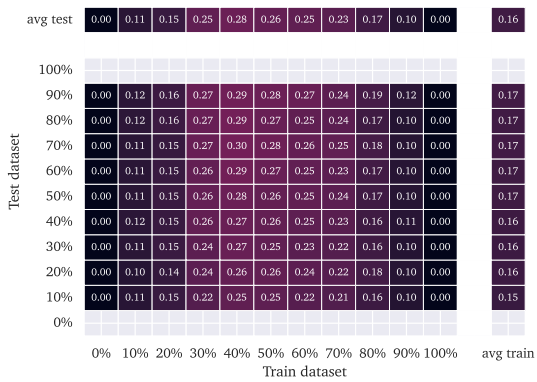

Figure 24: Youden's J statistic of models based on T5

G.2 Results by dataset modification strategy: replacement vs. augmentation

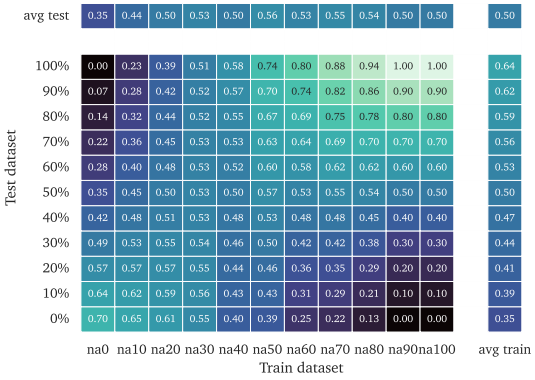

Figure 25: Accuracy using datasets generated by replacement only.

Figure 26: Accuracy using datasets augmented with unanswerable questions.

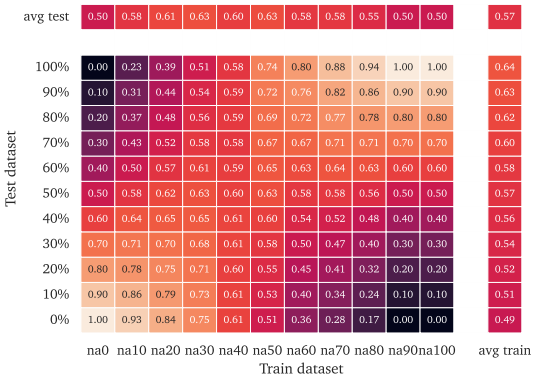

Figure 27: Answerability accuracy using datasets generated by replacement only.

Figure 28: Answerability accuracy using datasets augmented with unanswerable questions.

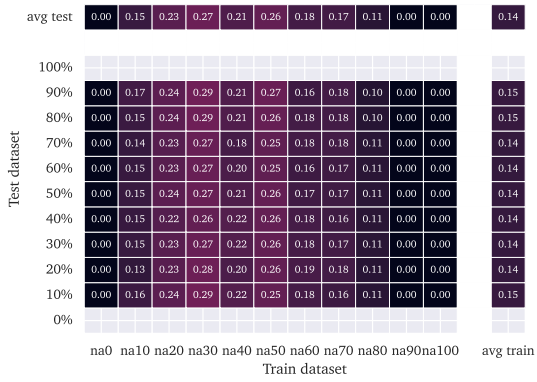

Figure 29: Youden's J statistic using datasets generated by replacement only.

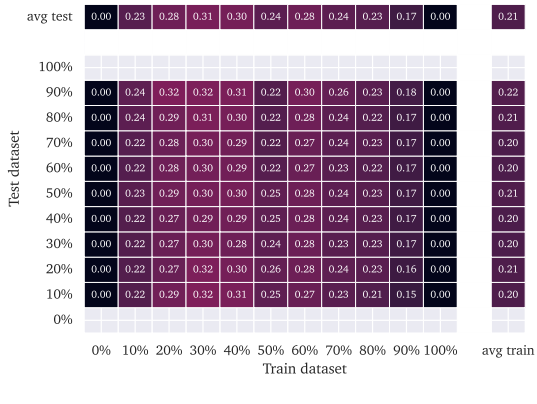

Figure 30: Youden's J statistic using datasets augmented with unanswerable questions.

G.3 Results by number of options per question

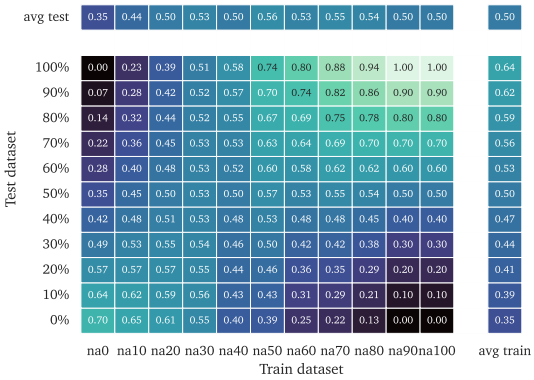

Figure 31: Accuracy: 4 options per question (A, B, C, D)

Figure 32: Accuracy: 3 options per question (A, B, C)

Figure 33: Accuracy: 2 options per question (A, B)

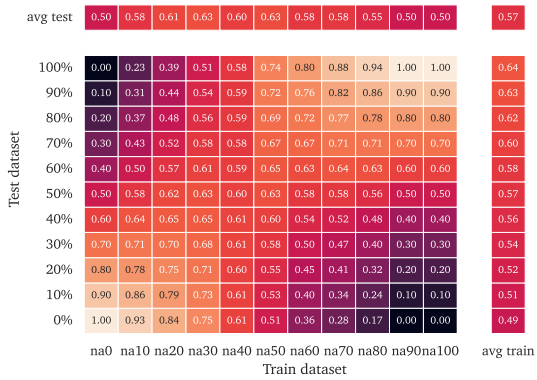

Figure 34: Accuracy: 4 options per question (A, B, C, D)

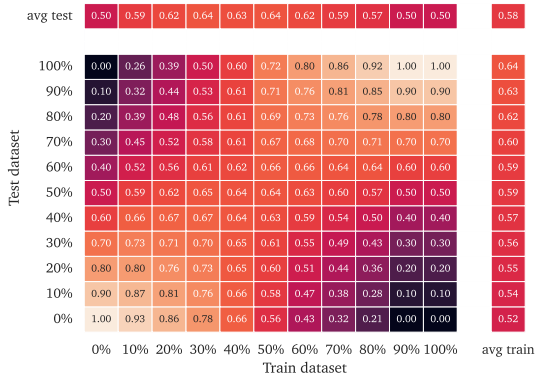

Figure 35: Accuracy: 3 options per question (A, B, C)

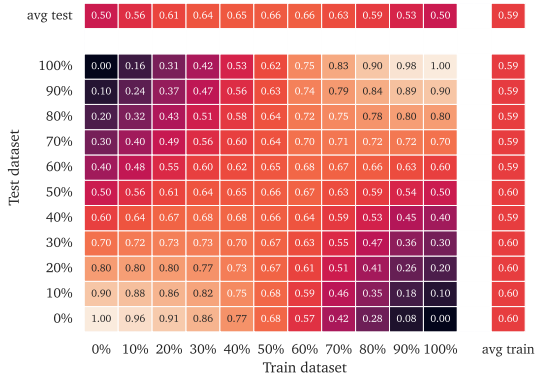

Figure 36: Accuracy: 2 options per question (A, B)

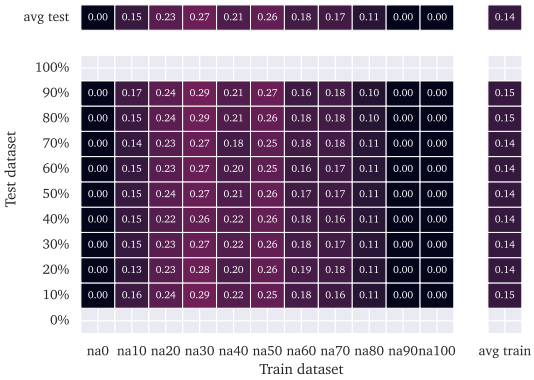

Figure 37: Youden's J statistic: 4 options per question (A, B, C, D)

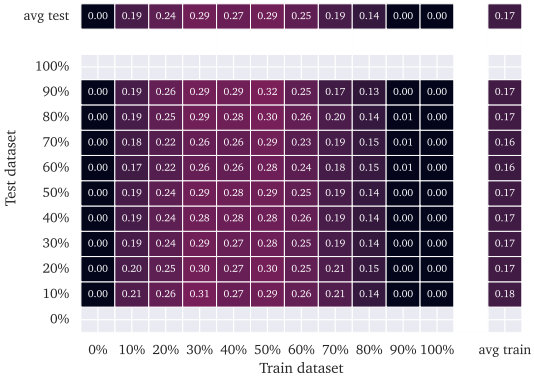

Figure 38: Youden's J statistic: 3 options per question (A, B, C)

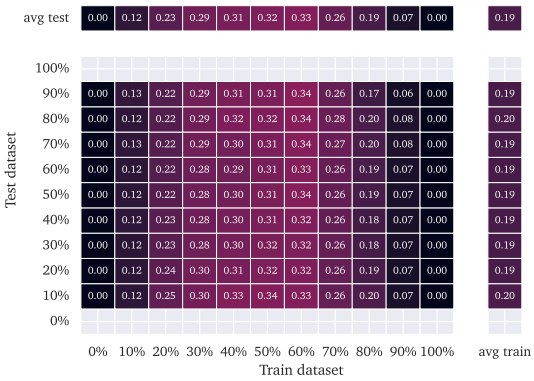

Figure 39: Youden's J statistic: 2 options per question (A, B)