AGent: A Novel Pipeline for Automatically Creating Unanswerable Questions

Anonymous ACL submission

Abstract

The development of large high-quality datasets and high-performing models has led to significant advancements in the domain of Extractive Question Answering (EQA). This progress has sparked considerable interest in exploring unanswerable questions within the EQA domain. Training EQA models with unanswerable questions helps them avoid extracting misleading or incorrect answers for queries that lack valid responses. However, manually annotating unanswerable questions is labor-intensive. To address this, we propose *AGent*, a novel pipeline that automatically creates new unanswerable questions by re-matching a question with a new context that lacks the necessary information for a correct answer. In this paper, we demonstrate the usefulness of this *AGent* pipeline by creating two sets of unanswerable questions from answerable questions in SQuAD and HotpotQA. These created question sets exhibit low error rates. Additionally, models fine-tuned on *AGent* unanswerable questions show comparable performance with those fine-tuned on the SQuAD 2.0 dataset on multiple EQA benchmarks.

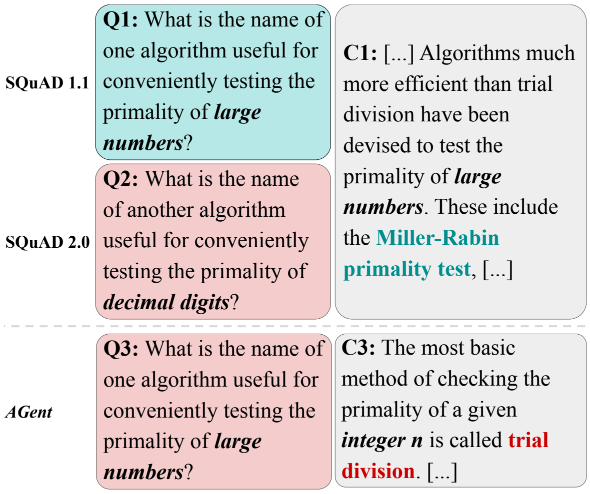

Figure 1: Examples of an answerable question $Q1$ from SQuAD 1.1, and two unanswerable questions $Q2$ from SQuAD 2.0 and $Q3$ from SQuAD *AGent*. In SQuAD 2.0, crowdworkers create unanswerable questions by replacing "large numbers" with "decimal digits." On the other hand, our automated *AGent* pipeline matches the original question $Q1$, now $Q3$, with a new context $C3$. The pair $C3 - Q3$ is unanswerable as context $C3$ does not indicate whether the **trial division** can **conveniently** test the primality of **large** numbers.

1 Introduction

Extractive Question Answering (EQA) is an important task of Machine Reading Comprehension (MRC), which has emerged as a prominent area of research in natural language understanding. Research in EQA has made significant gains thanks to the availability of many challenging, diverse, and large-scale datasets (Rajpurkar et al., 2016, 2018; Kwiatkowski et al., 2019; Yang et al., 2018; Trivedi et al., 2022). Moreover, recent advancements in datasets also lead to the development of multiple systems in EQA (Huang et al., 2018; Zaheer et al., 2020) that have achieved remarkable performance, approaching or even surpassing human-level performance across various benchmark datasets.

Matching the rapid progress in EQA, the subfield of unanswerable questions has emerged as a new research area. Unanswerable questions are those that cannot be answered based only on the information provided in the corresponding context. Unanswerable questions are a critical resource in training EQA models because they allow the models to learn how to avoid extracting misleading answers when confronted with queries that lack valid responses. Incorporating unanswerable questions in the training set of EQA models enhances the overall reliability of these models for real-world applications (Tran et al., 2023).

Nevertheless, the manual annotation of unanswerable questions in EQA tasks can be prohibitively labor-intensive. Consequently, we present a novel pipeline to automate the creation of high-quality unanswerable questions given a dataset comprising answerable questions. This pipeline uses a retriever to re-match questions with paragraphs that lack the necessary information to answer them. Additionally, it incorporates the concept of adversarial filtering for identifying challenging unanswerable questions. The key contributions of our work can be summarized as follows:

1. We propose ***AGent*** which is a novel pipeline for automatically creating unanswerable questions. In order to prove the utility of *AGent*, we apply our pipeline on two datasets with different characteristics, SQuAD and HotpotQA, to create two different sets of unanswerable questions. In our study, we show that the two unanswerable question sets created using *AGent* pipeline exhibit a low error rate.

2. Our experiments show that the two unanswerable question sets created using our proposed pipeline are challenging for models fine-tuned using human annotated unanswerable questions from SQuAD 2.0. Furthermore, our experiments show that models fine-tuned using our automatically created unanswerable questions show comparable performance to those fine-tuned using the SQuAD 2.0 dataset on various EQA benchmarks, such as SQuAD 1.1, HotpotQA, and Natural Questions.

2 Related Work

2.1 Unanswerable Questions

In the early research on unanswerable questions, Levy et al. (2017) re-defined the BiDAF model (Seo et al., 2017) to allow it to output whether the given question is unanswerable. Their primary objective was to utilize MRC as indirect supervision for relation extraction in zero-shot scenarios.

Subsequently, Rajpurkar et al. (2018) introduced a crowdsourcing process to annotate unanswerable questions, resulting in the creation of the SQuAD 2.0 dataset. This dataset later inspired similar works in other languages, such as French (Heinrich et al., 2022) and Vietnamese (Nguyen et al., 2022). However, recent research has indicated that models trained on SQuAD 2.0 exhibit poor performance on out-of-domain samples (Sulem et al., 2021).

Furthermore, apart from the adversarially-crafted unanswerable questions introduced by Rajpurkar et al. (2018), Natural Question (Kwiatkowski et al., 2019) and Tydi QA (Clark et al., 2020) present more naturally constructed unanswerable questions. While recent language models surpass human performances on adversarial unanswerable questions of SQuAD 2.0, natural unanswerable questions in Natural Question and Tidy QA remain a challenging task (Asai and Choi, 2021).

In a prior work, Zhu et al. (2019) introduce a pair-to-sequence model for generating unanswerable questions. However, this model requires a substantial number of high-quality unanswerable questions from SQuAD 2.0 during the training phase to generate its own high-quality unanswerable questions. Therefore, the model introduced by Zhu et al. (2019) cannot be applied on the HotpotQA dataset for generating high-quality unanswerable questions. In contrast, although our *AGent* pipeline cannot generate questions from scratch, it distinguishes itself by its ability to create high-quality unanswerable questions without any preexisting sets of unanswerable questions.

2.2 Robustness of MRC Models

The evaluation of Machine Reading Comprehension (MRC) model robustness typically involves assessing their performance against adversarial attacks and distribution shifts. The research on adversarial attacks in MRC encompasses various forms of perturbations (Si et al., 2021). These attacks include replacing words with WordNet antonyms (Jia and Liang, 2017), replacing words with words having similar representations in vector space (Jia and Liang, 2017), substituting entity names with other names (Yan et al., 2022), paraphrasing question (Gan and Ng, 2019; Ribeiro et al., 2018), or injecting distractors into sentences (Jia and Liang, 2017; Zhou et al., 2020). Recently, multiple innovative studies have focused on enhancing the robustness of MRC models against adversarial attacks (Chen et al., 2022; Zhang et al., 2023; Tran et al., 2023).

On the other hand, in the research line of robustness under distribution shift, researchers study the robustness of models in out-of-domains settings using test datasets different from training dataset (Miller et al., 2020; Fisch et al., 2019; Sen and Saffari, 2020).

3 Tasks and Models

In the task of EQA, models are trained to extract a list of prospective outputs (answers), each accom-

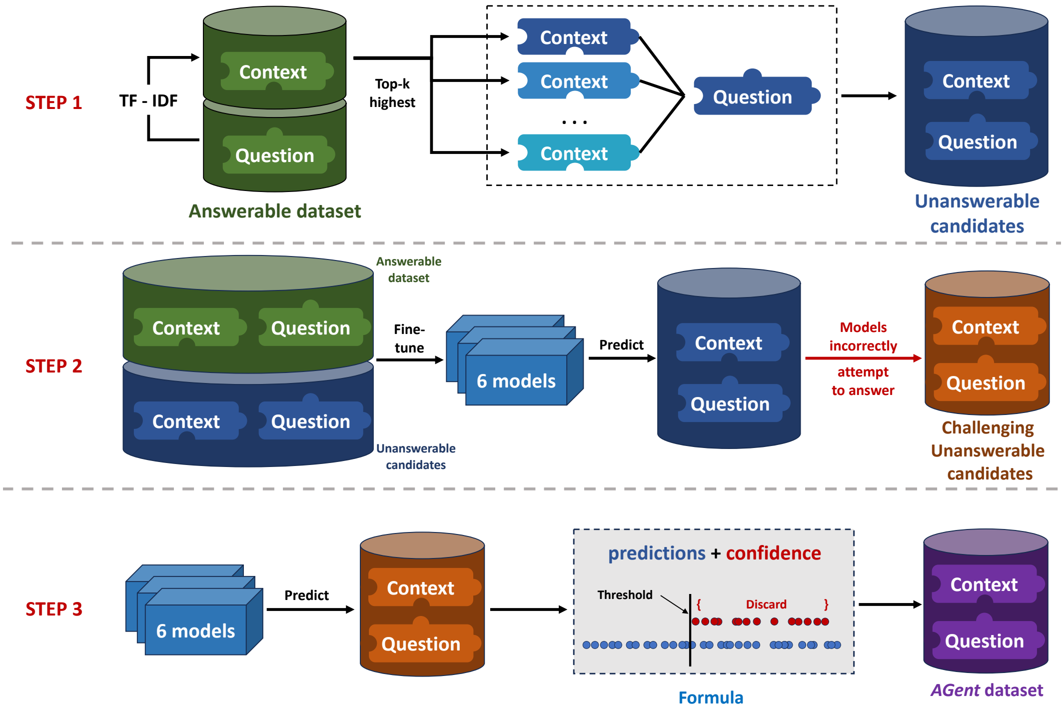

Figure 2: The *AGent* pipeline for generating challenging high-quality unanswerable questions in Extractive Question Answering given a dataset with answerable questions. In step 3 of the pipeline, the blue dots represent the calculated values (using formula discussed in §4.2) for unanswerable questions, while the red dots represent the calculated values for answerable questions. The threshold for discarding questions from the final extracted set of unanswerable questions is determined by finding the minimum value among all answerable questions. Any question with a calculated value greater than the threshold will not be included in our final extracted set.

panied by a probability (output of softmax function) that represents the machine's confidence in the answer's accuracy. When the dataset includes unanswerable questions, a valid response in the extracted list can be an "empty" response indicating that the question is unanswerable. The evaluation metric commonly used to assess the performance of the EQA system is the F1-score, which measures the average overlap between the model's predictions and the correct answers (gold answers) in the dataset. For more detailed information, please refer to the work by Rajpurkar et al. (2016).

3.1 Datasets

In our work, we utilize three datasets: SQuAD (Rajpurkar et al., 2016, 2018), HotpotQA (Yang et al., 2018), and Natural Questions (Kwiatkowski et al., 2019). In the SQuAD dataset, each question is associated with a short paragraph from Wikipedia. HotpotQA is a dataset designed for multi-hop reasoning question answering where each question requires reasoning over multiple supporting paragraphs. Additionally, the Natural Questions dataset comprises real queries from the Google search engine, and each question is associated with a Wikipedia page.

3.2 Models

We employ three transformer-based models in our work: BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and SpanBERT (Joshi et al., 2020). **BERT** is considered the pioneering application of the Transformer model architecture (Vaswani et al., 2017). BERT is trained on a combination of English Wikipedia and BookCorpus using masked language modeling and next-sentence prediction as pre-training tasks. Later, a replication study by Liu et al. (2019) found that BERT was significantly under-trained. Liu et al. (2019) built **RoBERTa** from BERT by extending the pre-training time and increasing the size of the pre-training data. Joshi et al. (2020) developed **SpanBERT** by enhancing BERT's ability to represent and predict text spans by masking random contiguous spans and replacing NSP with a span boundary objective.

Each of these three models has two versions: base and large. Our study uses all six of these models.

4 Automatically Creating Unanswerable Questions

4.1 Criteria

In order to guarantee the quality of our automatically created unanswerable questions, we design our pipeline to adhere to the following criteria:

Relevance. The created unanswerable questions should be closely related to the subject matter discussed in the corresponding paragraph. This criterion ensures that the unanswerability of the question is not easily recognizable by simple heuristic methods and that the created question "makes sense" regarding the provided context.

Plausibility. Our pipeline also ensures that the created unanswerable questions have at least one plausible answer. For instance, when considering a question like "What is the name of one algorithm useful for conveniently testing the primality of large numbers?", there should exist a plausible answer in the form of the name of an algorithm in Mathematics that is closely linked to the primality within the corresponding context. See Figure 1 for an example showcasing an unanswerable question with strong plausible answer(s).

Fidelity. Our pipeline adds an additional step to ensure a minimal rate of error or noise in the set of automatically created unanswerable questions. It is important that the newly created questions are genuinely unanswerable. This quality control measure improves the reliability of the pipeline. The effectiveness of this step will be verified in the study in Section 4.3.

4.2 *AGent* Pipeline

Figure 2 summarizes all the steps in the *AGent* pipeline for automatically creating unanswerable questions corresponding to each dataset of answerable questions. Our proposed *AGent* pipeline consists of three steps which align with the three criteria discussed in Section 4.1:

Step 1

Matching questions with new contexts. In the EQA task, the input consists of a question and a corresponding context. By matching the question with a new context that differs from the original context, we create a new question-context pair that is highly likely to be unanswerable. This step prioritizes the criterion of **relevance**. We employ the term frequency–inverse document frequency (TF-IDF) method to retrieve the k most relevant paragraphs from the large corpus containing all contexts from the original dataset (while obviously discarding the context that was originally matched with this question). The outcome of this step is a set of **unanswerable candidates**. It's important to note that the unanswerable candidates created in this step may includes some answerable questions, and these answerable questions will be filtered out in step 3 of the pipeline.

Step 2

Identifying hard unanswerable questions. In this step, we give priority to both the **relevance** and **plausibility** criteria. We aim to identify unanswerable questions with a highly relevant corresponding context and at least one strong plausible answer. To achieve this, we leverage the concept of adversarial filtering where the adversarial model(s) is applied to filter out easy examples (Yang et al., 2018; Zellers et al., 2018; Zhang et al., 2018).

We first fine-tune six models using a dataset comprising answerable questions from the original dataset and randomly selected unanswerable candidates. We acknowledge that some unanswerable questions in this training set may be answerable. Nevertheless, the percentage of answerable questions among the unanswerable candidates is minimal and within an acceptable range (Appendix A.2). To ensure training integrity, we then exclude all unanswerable questions utilized for training these six models from the set of unanswerable candidates. Then, we employ the six fine-tuned models to evaluate the difficulty of each sample in the set of unanswerable candidates. If at least two of the six models predict that a given question is answerable, we consider it to be a challenging unanswerable question and include it in our set of **challenging unanswerable candidates**. [1]

Step 3

Filtering out answerable questions. The set of challenging unanswerable candidates consists of questions that at least two out of the six models predict as answerable. Consequently, there may be a considerable percentage of questions that are indeed answerable. Therefore, this specific step in our pipeline aims to ensure the **fidelity** of the *AGent* pipeline, ensuring that most questions created by our pipeline are genuinely unanswerable. We leverage the predicted answers and confidence scores

[1] An ablation study on the adversarial threshold is presented in Appendix 6.

from the six deployed models in the previous step to achieve this. Subsequently, we devise a filtering model with four inputs: c_a, representing the cumulative confidence scores of the models attempting to answer (or predicting as answerable); c_u, representing the cumulative confidence scores of the models not providing an answer (or predicting as unanswerable); n_a, denoting the number of models attempting to answer; and n_u, denoting the number of models not providing an answer. The output of this filtering model is a value $V(q)$ for each question q. The filtering models must be developed independently for different datasets.

In order to determine the filtering threshold and develop the filtering model, we manually annotate 200 challenging unanswerable candidates from each dataset. The filtering threshold is established by identifying the minimum value $V(q_a)$ where q_a represents an answerable question from our annotated set. This approach ensures a precision of 100% in identifying unanswerable questions on the annotated 200 questions. The filtering model then acts to minimize the number of false positives (number of unanswerable candidates that are answerable) at the expense of tossing out some candidate questions that are unanswerable. However, as the filtering model is applied on unseen challenging unanswerable candidates, the precision of the filtering model in this step would not be 100% as on the 200 muanually annotated samples. Therefore, in next section, we use human experts to evaluate the precision exhibited by the filtering model.

Further details for the *AGent* pipeline are outlined in Appendix A.

4.3 Human Reviewing

This section presents our methodology for evaluating the data quality of unanswerable questions automatically created by *AGent*.

		Phase 1	Phase 2
SQuAD	Fleiss' Kappa	0.76	0.95
AGent	Data Error	0.10	**0.06**
HotpotQA	Fleiss' Kappa	0.83	0.97
AGent	Data Error	0.09	**0.05**

Table 1: The Fleiss' Kappa score and *AGent* data error for the annotations collected from human experts after two distinct phases.

We use three experts to validate 100 random unanswerable questions from each development set of SQuAD *AGent* and HotpotQA *AGent*. In order to prevent an overwhelming majority of unanswerable questions in our annotation set, which could potentially undermine the integrity of the annotation, we randomly incorporate 20 questions we already manually labeled as answerable and that are not included in the final *AGent* datasets. Consequently, we provide a total of 120 questions to each expert for each set.

The process of expert evaluation involves two distinct phases. During the first phase, each of the three experts independently assesses whether a given question is answerable and provides the reasoning behind their annotation. In the second phase, all three experts are presented with the reasons provided by the other experts for any conflicting samples. They have the opportunity to review and potentially modify their final set of annotations based on the reasons from their peers.

Our three experts provided high-quality annotations. Table 1 presents the Fleiss' Kappa score (Fleiss, 1971) for our three experts after the completion of both phases, as well as the error rate of the *AGent* development set. Notably, the Fleiss' Kappa score, which measures the level of agreement among experts, in phase 1 is remarkably high (0.76 on SQuAD AGent and 0.83 on HotpotQA AGent). This strong agreement between experts after the first phase shows their expertise in the task and suggests that the annotations obtained through this process are reliable.

As demonstrated in Table 1, the high-quality annotations provided by three experts indicate an exceptionally low error rate for the unanswerable questions created using *AGent* (6% for SQuAD and 5% for HotpotQA). For comparison, these error rates are slightly lower than that of SQuAD 2.0, a dataset annotated by humans.

5 Experiments and Analysis

We now shift our attention from the *AGent* pipeline to examining the effectiveness of our *AGent* questions in training and benchmarking EQA models.

5.1 Training Sets

The models in our experiments are trained using SQuAD 2.0, SQuAD *AGent*, and HotpotQA *AGent*. **SQuAD *AGent*** includes all answerable questions from SQuAD 1.1 and *AGent* unanswerable questions. To create the SQuAD *AGent* unanswerable questions, we feed answerable questions from

Test → / *Train* ↓	SQuAD answerable	SQuAD unanswerable	SQuAD *AGent*	HotpotQA answerable	HotpotQA unanswerable	HotpotQA *AGent*	Natural Questions answerable	Natural Questions unanswerable
SQuAD 2.0	$84.55_{\pm 3.43}$	$\mathbf{79.16}_{\pm 5.16}$	$49.38_{\pm 5.21}$	$51.05_{\pm 5.15}$	$86.28_{\pm 2.68}$	$58.98_{\pm 4.64}$	$\mathbf{44.30}_{\pm 6.36}$	$60.55_{\pm 12.95}$
SQuAD *AGent*	$\mathbf{86.96}_{\pm 1.86}$	$29.63_{\pm 3.97}$	$81.38_{\pm 4.52}$	$63.26_{\pm 2.88}$	$90.01_{\pm 2.40}$	$50.61_{\pm 5.56}$	$41.05_{\pm 6.81}$	$78.66_{\pm 13.22}$
HotpotQA *AGent*	$59.06_{\pm 6.26}$	$46.13_{\pm 3.46}$	$\mathbf{87.61}_{\pm 2.72}$	$\mathbf{77.75}_{\pm 1.92}$	$\mathbf{99.70}_{\pm 0.06}$	$\mathbf{95.94}_{\pm 2.13}$	$24.11_{\pm 7.04}$	$\mathbf{84.20}_{\pm 11.37}$

Table 2: Performance of 6 models fine-tuned on SQuAD 2.0, SQuAD *AGent*, and HotpotQA *AGent* datasets evaluated on SQuAD, HotpotQA, and Natural Questions. Each entry in the table is the mean and standard deviation of the F1 scores of the six MRC models. *AGent* (test sets) refers to the unanswerable questions created using the *AGent* pipeline. For a more detailed version of this table, refer to Table 12.

SQuAD 1.1 into the *AGent* pipeline. Similarly, we also use the *AGent* pipeline to create HotpotQA *AGent* unanswerable questions from the original dataset HotpotQA. The **HotpotQA *AGent*** train set includes HotpotQA *AGent* unanswerable questions and original HotpotQA answerable questions.

5.2 Testing Sets

In our experiments, we use eight sets of EQA questions as summarized in Table 2. In addition to two sets of *AGent* unanswerable questions, we also incorporate the following six types of questions.

SQuAD. We use all **answerable** questions from SQuAD 1.1. We use all **unanswerable** questions from SQuAD 2.0.

HotpotQA. In preprocessing **answerable** questions in HotpotQA, we adopt the same approach outlined in MRQA 2019 (Fisch et al., 2019) to convert each dataset to the standardized EQA format. Specifically, we include only two supporting paragraphs in our answerable questions and exclude distractor paragraphs.

In preprocessing **unanswerable** questions in HotpotQA, we randomly select two distractor paragraphs provided in the original HotpotQA dataset, which are then used as the context for the corresponding question.

Natural Questions (NQ). In preprocessing **answerable** questions in NQ, we again adopt the same approach outlined in MRQA 2019 to convert each dataset to the standardized EQA format. This format entails having a single context, limited in length. Specifically, we select examples with short answers as our answerable questions and use the corresponding long answer as the context.

For **unanswerable** questions in NQ, we select questions with no answer and utilize the entire Wikipedia page, which is the input of original task of NQ, as the corresponding context. However, in line with the data collection process of MRQA 2019, we truncate the Wikipedia page, limiting it to the first 800 tokens.

5.3 Main Results

Table 2 presents the results of our experiments. Firstly, our findings demonstrate that unanswerable questions created by *AGent* pose significant challenges for models fine-tuned on SQuAD 2.0, a dataset with human-annotated unanswerable questions. The average performance of the six models fine-tuned on SQuAD 2.0 and tested on SQuAD *AGent* is 49.38; the average score for testing these models on HotpotQA *AGent* data is 58.98. Notably, unanswerable questions from HotpotQA *AGent* are considerably more challenging compared to their unanswerable counterparts from HotpotQA.

Secondly, models fine-tuned using two *AGent* datasets exhibit comparable performance to models fine-tuned using SQuAD 2.0 on 7 out of 8 testing domains. On unanswerable questions from HotpotQA and NQ, models fine-tuned on *AGent* datasets significantly outperform those fine-tuned on SQuAD 2.0. On answerable questions from SQuAD and HotpotQA, models fine-tuned on SQuAD *AGent* also demonstrate significant improvement over those fine-tuned on SQuAD 2.0 ($86.96 - 84.55$ on SQuAD and $63.26 - 51.05$ on HotpotQA). This finding highlights the applicability of models fine-tuned on *AGent* datasets to various question types.

However, on answerable questions from NQ and unanswerable questions from SQuAD 2.0, models fine-tuned on *AGent* datasets exhibit lower performance than those fine-tuned on SQuAD 2.0. On the one hand, the lower performance on unanswerable questions from SQuAD 2.0 of models fine-tuned on *AGent* datasets is due to the unfair comparison as models fine-tuned on *AGent* datasets are tested with out-of-domain samples, and models fine-tuned with SQuAD 2.0 are tested with in-domain sam-

		SQuAD 2.0 %	SQuAD *AGent* %
Insufficient context for question	Murray survives and , in front of the RGS trustees , accuses Fawcett of abandoning him in the jungle . Fawcett elects to resign from the society rather than apologize . World War I breaks out in Europe , and Fawcett goes to France to fight . Manley dies in the trenches at the Battle of the Somme , and Fawcett is temporarily blinded in a chlorine gas attack . Jack , Fawcett 's eldest son – who had long accused Fawcett of abandoning the family – reconciles with his father as he recovers . **Question**: who dies in the lost city of z?	54	63

Table 3: Example of an answerable question in Natural Questions that is predicted as unanswerable by models fine-tuned on **SQuAD 2.0** and **SQuAD *AGent*** due to insufficient context from the provided context.

ples. On the other hand, in the next section, we provide a comprehensive explanation for the lower performance on NQ answerable questions of models fine-tuned on *AGent* datasets.

5.4 Analysis on Natural Questions

To delve deeper into the underperformance of models fine-tuned on *AGent* dataset on answerable questions of NQ, we analyze two sets of answerable questions from NQ. The first set is 100 answerable questions that models fine-tuned on SQuAD *AGent* predict as unanswerable; the second one is 100 answerable questions that models fine-tuned on SQuAD 2.0 predict as unanswerable. For the sake of simplicity, we limit our reporting in this section to the analysis of models RoBERTa-base. Our analysis uncovers an issue that can arise when evaluating models with answerable questions from the NQ dataset. [2]

A considerable difference between the original NQ dataset and the NQ used in the EQA task following a prevailing approach in the research community is the difference in the provided context. While original NQ task supplies an entire Wikipedia page as the context for a given question, NQ in the EQA task uses the long answer as the context (Fisch et al., 2019). This difference presents a potential problem of inadequate context for answering the question. For instance, in Table 3, we observe that the long answer associated with the question "Who dies in the lost city of z?" fails to mention "the lost city of z". Using a long answer as the context causes this question unanswerable due to the insufficient context provided. We find that most answerable questions predicted as unanswerable by models fine-tuned on SQuAD 2.0 and SQuAD *AGent* belong to this specific question type (54% and 63% respectively). This finding highlights the potential unreliability when comparing models using the NQ dataset in the same way as it is commonly done in multiple EQA studies.

[2] We discuss another minor issue in Appendix B.

6 Ablation Study: Adversarial Threshold

In step 2 of *AGent* pipeline, we consider a question to be a challenging unanswerable candidate if at least **two** adversarial models predict that question as answerable. We denote this number of adversarial models predicting answerable as ***adversarial threshold***. In this section, we study how this threshold affects the quality of our final *AGent* dataset.

SQuAD	*Adversarial Threshold*	Train	Test	Data Error
AGent A1	1	34,908	4,611	0.04
AGent	2	48,016	2,217	0.06
AGent A3	3	11,501	1,619	0.13

Table 4: Data statistics of SQuAD *AGent* datasets with different adversarial thresholds. The *AGent* data error are collected through the same human reviewing process as in Section 4.3.

		AGent A1	*AGent*	*AGent A3*
BERT	base	50.1	43.6	38.6
	large	54.3	46.5	37.4
RoBERTa	base	64.6	54.1	44.6
	large	66.2	57.1	48.7
SpanBERT	base	53.9	45.9	37.4
	large	59.0	49.1	40.0
Average		$58.0_{\pm 6.4}$	$49.4_{\pm 5.2}$	$41.1_{\pm 4.6}$

Table 5: Performance of 6 models fine-tuned on SQuAD 2.0 evaluated on test set of SQuAD *AGent* with different adversarial thresholds.

For the purpose of simplicity, we focus our report on SQuAD *AGent*. We follow the same *AGent* pipeline outlined in Section 4.2, but with different adversarial threshold: 1, 2 (current AGent), and

Test → *Train* ↓	*Adversarial Threshold*	**SQuAD** answerable	**SQuAD** unanswerable	**HotpotQA** answerable	**HotpotQA** unanswerable	**HotpotQA** *AGent*	**NQ** answerable	**NQ** unanswerable
SQuAD AGent A1	1	**87.6**$_{\pm 2.4}$	29.4$_{\pm 3.1}$	59.0$_{\pm 3.9}$	**92.3**$_{\pm 0.9}$	**52.3**$_{\pm 4.0}$	38.8$_{\pm 7.9}$	**81.6**$_{\pm 2.9}$
SQuAD AGent	2	87.0$_{\pm 1.9}$	**29.6**$_{\pm 4.0}$	**63.3**$_{\pm 2.9}$	90.1$_{\pm 2.4}$	50.6$_{\pm 5.6}$	41.1$_{\pm 6.8}$	78.7$_{\pm 13.2}$
SQuAD AGent A3	3	88.4$_{\pm 1.9}$	27.0$_{\pm 5.6}$	60.0$_{\pm 2.4}$	86.0$_{\pm 4.6}$	46.4$_{\pm 8.4}$	**45.1**$_{\pm 10.5}$	72.5$_{\pm 24.4}$

Table 6: Performance of 6 models fine-tuned on SQuAD *AGent A1* , SQuAD *AGent*, and SQuAD *AGent A3* datasets evaluated on SQuAD, HotpotQA, and Natural Questions. Each entry in the table is the mean and standard deviation of the F1 scores of the six MRC models. *AGent* (test sets) refers to the unanswerable questions created using the *AGent* pipeline. For a more detailed version of this table, refer to Table 12.

3. Our study focus on three criteria for assessing the dataset: Data Error (as discussed in Section 4.3), Test set Difficulty, and Usefulness of Train set (Section 5).

Data Error

Table 4 reports the number of unanswerable questions and data error rates *AGent* datasets using adversarial thresholds of 1, 2, and 3. *AGent A1*, *AGent* and *AGent A3* correspond to datasets with adversarial thresholds set at 1, 2, and 3, respectively. We observe that increasing the adversarial threshold to 3 would significantly decrease the number of unanswerable questions created by the *Agent* pipeline ($48,016 - 11,501$ on Train set and $2,217 - 1,619$ on Test set) and increase the data error rate to 13%, which is significantly higher than that of SQuAD 2.0. On the other hand, *AGent* datasets using adversarial thresholds of 1 and 2 have the data error rate lower than that of SQuAD 2.0.

Test Set Difficulty

Table 5 presents the performance of models fine-tuned on SQuAD 2.0 when evaluated on the test sets of *Agent* datasets with varying adversarial thresholds. We observe that as we increase the adversarial threshold, *AGent* unanswerable questions become more challenging, and we see a corresponding decline in the performance of models fine-tuned on SQuAD 2.0 tested on *AGent* unanswerable questions.

Train Set Usefulness

To evaluate the usefulness of *AGent* train sets with different adversarial thresholds, we fine-tune 6 models on each train set and evaluate on testing sets described in Section 5. Table 6 reports our experimental results. Our findings reveal that models fine-tuned on SQuAD *AGent A3* exhibit significantly lower performance compared to models fine-tuned on the other two train sets. This performance gap can be attributed to the notably high data error rates and a limited number of unanswerable questions in the SQuAD *AGent A3* dataset. On the other hand, models fine-tuned on the current SQuAD *AGent* and SQuAD *AGent A1* show similar performance.

7 Conclusion and Future Works

In this work, we propose *AGent*, a novel pipeline creates unanswerable questions from datasets of answerable questions. We systematically apply *AGent* on SQuAD and HotpotQA to create unanswerable questions. Through a two-stage process of human reviewing, we demonstrate that *AGent* unanswerable questions exhibit a low error rate.

Our experimental results indicate that unanswerable questions created using AGent pipeline present significant challenges for EQA models fine-tuned on SQuAD 2.0. We also demonstrate that models fine-tuned using *AGent* unanswerable questions exhibit competitive performance compared to models fine-tuned on human-annotated unanswerable questions from SQuAD 2.0 on multiple test domains. The good performance of models finetuned on two *AGent* datasets with different characteristics, SQuAD *AGent* and HotpotQA *AGent*, demonstrate the utility of *AGent* in creating high-quality unanswerable questions. Furthermore, our analysis sheds light on a potential issue when utilizing the NQ dataset in the task of EQA. Specifically, we identify the problems of insufficient provided context, which can cause EQA to predict an answerable question as unanswerable.

Our work also provides a comprehensive ablation study on the adversarial threshold in step 2 of the *AGent* pipeline. We hope that our efforts can shed light on the broader application of *AGent* pipeline in EQA in future research.

Limitations

We acknowledge certain limitations in our work. Firstly, our study primarily focuses on evaluating the pipeline using multiple pre-trained transformers-based models in English, which can be prohibitively expensive to create, especially for languages with limited resources. Furthermore, given the empirical nature of our study, there is no guarantee that all other transformer-based models or other deep neural networks would demonstrate the same level of effectiveness when applied in the *AGent* pipeline. Consequently, the impact of the AGent pipeline on low-resource languages may be challenged due to this limitation. Potential future research could complement our findings by investigating the effectiveness of implementing *AGent* pipeline in other languages.

Secondly, our analysis does not encompass a comprehensive examination of the models' robustness against various types of adversarial attacks in EQA when fine-tuned on *AGent* datasets. We believe that such an analysis is crucial in determining the effectiveness of the *AGent* pipeline in real-world applications, and its absence deserves further research.

Finally, our study has not discussed underlying factors for the observed phenomenon: a model fine-tuned on SQuAD AGent is less robust against TextBugger attack than its peer model fine-tuned on SQuAD 2.0 (in Appendix B). The study in this direction requires remarkably intricate investigation, which we believe beyond the scope of our present research. We leave this for our future work where we will propose our hypotheses that may shed light on this phenomenon and potential solutions to improve the robustness of EQA models against TextBugger.

References

Akari Asai and Eunsol Choi. 2021. Challenges in information-seeking QA: Unanswerable questions and paragraph retrieval. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1492–1504, Online. Association for Computational Linguistics.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Howard Chen, Jacqueline He, Karthik Narasimhan, and Danqi Chen. 2022. Can rationalization improve robustness? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3792–3805, Seattle, United States. Association for Computational Linguistics.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 1–13, Hong Kong, China. Association for Computational Linguistics.

Joseph Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.

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, Florence, Italy. Association for Computational Linguistics.

Quentin Heinrich, Gautier Viaud, and Wacim Belblidia. 2022. FQuAD2.0: French question answering and learning when you don't know. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2205–2214, Marseille, France. European Language Resources Association.

Hsin-Yuan Huang, Chenguang Zhu, Yelong Shen, and Weizhu Chen. 2018. Fusionnet: Fusing via fully-aware attention with application to machine comprehension. In *International Conference on Learning Representations*.

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, Copenhagen, Denmark. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, Vancouver, Canada. Association for Computational Linguistics.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2019. TextBugger: Generating adversarial text against real-world applications. In *Proceedings 2019 Network and Distributed System Security Symposium*. Internet Society.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

John Miller, Karl Krauth, Benjamin Recht, and Ludwig Schmidt. 2020. The effect of natural distribution shift on question answering models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 6905–6916. PMLR.

Kiet Van Nguyen, Son Quoc Tran, Luan Thanh Nguyen, Tin Van Huynh, Son T. Luu, and Ngan Luu-Thuy Nguyen. 2022. VLSP 2021 - ViMRC challenge: Vietnamese machine reading comprehension.

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, Melbourne, Australia. Association for Computational Linguistics.

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, Austin, Texas. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging NLP models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865, Melbourne, Australia. Association for Computational Linguistics.

Priyanka Sen and Amir Saffari. 2020. What do models learn from question answering datasets? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2429–2438, Online. Association for Computational Linguistics.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *International Conference on Learning Representations*.

Chenglei Si, Ziqing Yang, Yiming Cui, Wentao Ma, Ting Liu, and Shijin Wang. 2021. Benchmarking robustness of machine reading comprehension models. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 634–644, Online. Association for Computational Linguistics.

Elior Sulem, Jamaal Hay, and Dan Roth. 2021. Do we know what we don't know? studying unanswerable questions beyond SQuAD 2.0. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4543–4548, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Son Quoc Tran, Phong Nguyen-Thuan Do, Uyen Le, and Matt Kretchmar. 2023. The impacts of unanswerable questions on the robustness of machine reading comprehension models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1543–1557, Dubrovnik, Croatia. Association for Computational Linguistics.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Jun Yan, Yang Xiao, Sagnik Mukherjee, Bill Yuchen Lin, Robin Jia, and Xiang Ren. 2022. On the robustness of reading comprehension models to entity renaming. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 508–520, Seattle, United States. Association for Computational Linguistics.

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, Brussels, Belgium. Association for Computational Linguistics.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, volume 33, pages 17283–17297. Curran Associates, Inc.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium. Association for Computational Linguistics.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. Record: Bridging the gap between human and machine commonsense reading comprehension.

Yiming Zhang, Yangqiaoyu Zhou, Samuel Carton, and Chenhao Tan. 2023. Learning to ignore adversarial attacks. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2970–2984, Dubrovnik, Croatia. Association for Computational Linguistics.

Xiaorui Zhou, Senlin Luo, and Yunfang Wu. 2020. Co-attention hierarchical network: Generating coherent long distractors for reading comprehension. In *Proceedings of AAAI Conference on Artificial Intelligence*, volume 34, page 9725–9732. AAAI Press.

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, Florence, Italy. Association for Computational Linguistics.

A *AGent* on SQuAD and HotpotQA

	SQuAD *AGent*	HotpotQA *AGent*
Unanswerable Candidates	$975,520$	$1,800,550$
Challenging Candidates	$89,432$	$41,755$
AGent	$50,404$	$27,840$

Table 7: Statistics of SQuAD *AGent* and HotpotQA *AGent* after each step of the *AGent* pipeline.

A.1 Create Unanswerable Candidates

SQuAD. In order to create unanswerable candidates from questions in SQuAD 1.1, we employ bigram TF-IDF, using the question as the query (Chen et al., 2017), to retrieve the top-10 highest contexts from dataset SQuAD 1.1. Additionally, our algorithm includes a step to ensure that the set of top-10 highest TF-IDF scored contexts does not include the original context corresponding to the question. As a result, *AGent* creates $975,520$ unanswerable candidates from SQuAD 1.1.

HotpotQA. In constructing benchmark settings for HotpotQA, Yang et al. (2018) employ bigram TF-IDF, using the question as the query, to retrieve eight paragraphs from Wikipedia as distractors. Yang et al. (2018) then mix these distractors with the two gold paragraphs (the ones used to collect the question and answer). We then create unanswerable candidates from questions in HotpotQA by combining every two distractors from HotpotQA. Consequently, *AGent* creates $1,800,550$ unanswerable candidates from HotpotQA.

A.2 Identifying Challenging Unanswerable Candidates

Before using unanswerable candidates for fine-tuning the six adversarial models, we manually annotate 100 unanswerable candidates from each set of HotpotQA and SQuAD. After the manual annotation, we have 1 answerable question from the set of SQuAD and 2 from the set of HotpotQA. As the error rate from SQuAD 2.0 is 7%, we consider the error rate in unanswerable candidates is within the acceptable range for fine-tuning the six adversarial models.

In order to fine-tune adversarial models for identifying challenging unanswerable candidates, we randomly select a set of unanswerable questions from the set of unanswerable candidates from the previous step. Here, we adopt the ratio of answerable over unanswerable of SQuAD 2.0. As a result, the training set in this step for SQuAD consists of $87,599$ answerable and $43,799$ unanswerable questions; that for HotpotQA consists of $58,525$ answerable and $29,262$ unanswerable questions.

After step 2 of *AGent*, we have $89,432$ and $41,755$ challenging candidates on SQuAD and HotpotQA, respectively.

A.3 Filtering Model

We employ a model with the following formula to classify questions as answerable or unanswerable:

$$V(q) = c_a \cdot \alpha^{n_a} - c_u \cdot \beta^{n_u}$$

In our model, we have four inputs and two adjustable parameters. Firstly, c_a and c_u represent the total confidence scores of the models attempting to answer (or predict as answerable) and the models not providing an answer (or predict as unanswerable), respectively. Additionally, n_a and n_u denote the number of models attempting to answer and the number of models not providing an answer, respectively. The parameters α and β are tunable parameters.

In order to tune the filtering model, we manually annotate 200 questions from each set challenging unanswerable candidates. We define the difficulty level for a particular question as the number of models predicting it as answerable. Consequently, our sets of challenging unanswerable candidates encompass five difficulty levels (from 2 to 6). From each level, we randomly choose 40 questions for manual annotation.

Next, we employ grid search with the step size of 0.01 to tune for the parameters α and β within the range of $(0, 2]$ with the objective of maximizing the recall of unanswerable questions, aiming to include as many unanswerable questions as possible in our final dataset. As a result, on SQuAD, we have $\alpha = 0.64$ and $\beta = 0.69$; on HotpotQA, we have $\alpha = 0.52$ and $\beta = 0.94$. After going through the filtering model, SQuAD AGent has $50,404$ unanswerable questions; HotpotQA AGent has $27,840$.

B Minor Issue in Natural Questions

In analyzing the two sets of answerable questions predicted as unanswerable in Section 5.4, we discover another minor issue. The questions in the

		SQuAD 2.0 %	SQuAD *AGent* %
typographical errors of key words	Gimme Gimme Gimme has broadcast three series and 19 episodes in total . The first series premiered on BBC Two on 8 January 1999 and lasted for six episodes , concluding on 12 February 1999 . [..] **Question**: when did gim me gim me gim me start?	3	6

Table 8: Example of an answerable question in Natural Questions that is predicted as unanswerable due to typographical errors by models fine-tuned on **SQuAD 2.0** and **SQuAD** ***AGent***.

NQ dataset are sourced from real users who submitted information-seeking queries to the Google search engine under natural conditions. As a result, a small portion of these questions may inevitably contain typographical errors or misspellings. In our analysis, we observe that models fine-tuned on our *AGent* training set tend to predict the questions of this type as unanswerable more frequently. Nevertheless, due to the relatively small proportion of questions with typographical errors in our randomly surveyed sets, we refrain from drawing a definitive conclusion at this point. Therefore, in the subsequent section, we will delve further into this matter by adopting an adversarial attack on the EQA task. This approach aims to simulate and thoroughly examine the potential impact of syntactic deviations (i.e., typographical errors) on model performance.

B.1 TextBugger

In this section, we apply the adversarial attack technique TextBugger into EQA.

Our adversarial attack in this section is inspired by the TextBugger attack (Li et al., 2019). We use black-box TextBugger in this section, which means that the attack algorithm does not have access to the gradients of the model. TextBugger generates attack samples that closely resemble the typographical errors commonly made by real users. We perform adversarial attacks on questions from the SQuAD 1.1 dataset.

Original	Insert	Delete	Swap	Substitute Character
South	Sou th	Souh	Souht	S0uth
What **Souh** African law **recongized** two **typ es** of schools?				

Table 9: Examples of how TextBugger generates bugs in a given token "South" and a full question after the TextBugger attack. The attacked tokens are highlighted in red.

Algorithm 1 provides the pseudocode outlining the process of generating attacked questions. Table 9 provides examples of how TextBugger generates bugs in a given token.

B.2 Robustness against TextBugger

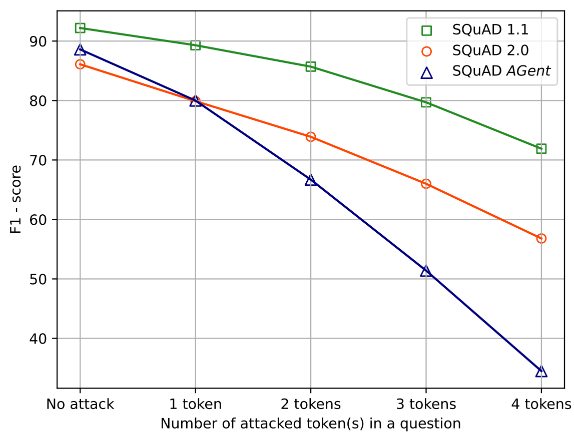

Figure 3: Robustness of RoBERTa-base trained on SQuAD 1.1, SQuAD 2.0, SQuAD *AGent* against TextBugger.

We investigate the impact of TextBugger attacks on models fine-tuned using different datasets, namely SQuAD 1.1, SQuAD 2.0, and SQuAD *AGent*. To accomplish this, we generate attacked questions by modifying 1, 2, 3, and 4 tokens in the questions from the SQuAD 1.1 dataset.

Figure 3 reports the performance of three models RoBERTa-base fine-tuned on SQuAD 1.1, SQuAD 2.0, and SQuAD *AGent*. Firstly, we see that the performance of the model fine-tuned on SQuAD 1.1 show small decreases (from 92.2 to 71.9). Adversarial attack TextBugger does not present a significant challenge to the EQA model when the model is designed only to handle answerable questions.

Secondly, we can observe that the model fine-tuned on unanswerable questions from SQuAD 2.0 demonstrates significantly better robustness compared to the model fine-tuned on SQuAD *AGent* ($86.1 - 56.8$ compared to $88.6 - 34.5$). This finding confirms our initial hypothesis that the lower per-

Algorithm 1: TextBugger EQA Attack

```
Function TextBugger(question, numAttack):
    attackPositions ← randomly select indices of tokens in question;
    forall pos ∈ attackPositions do
        question[pos] ← GenerateBug(question[pos]);
    end
Function GenerateBug(token):
    newToken ← token
    while newToken ≠ token do
        bugType ← randomly select Bug type;
        newToken ← Bug(newToken, bugType);
    end
    return newToken
```

formance of models fine-tuned on *AGent* datasets for answering questions in the NQ dataset is partly attributable to misspelled keywords in the questions from the NQ dataset.

C Details for Models Training

The input of a question-context pair into the pre-trained model is in the form of *<Question>[SEP]<Context>*, with *[SEP]* as a special token of pre-trained tokenizer accompanying the pre-trained model. After getting embeddings for each token, we feed its final embedding into a start and end token classifier. After taking the dot product between the output embeddings and the classifier's weights, we apply the softmax activation to produce a probability distribution over all words. The word with the highest probability after the start classifier will be predicted as the start of the answer span.

	total samples	# unanswerable
SQuAD *Adversarial*	130,319	43,439
HotpotQA *Adversarial*	87,787	29,262
SQuAD *AGent*	135,615	48,016
HotpotQA *AGent*	83,589	25,064
SQuAD 2.0	130,319	43,498

Table 10: Data statistics of all training sets used in this paper. Adversarial datasets refer to training sets for the adversarial models in Step 2.

Table 10 provides the statistics for all training sets in this paper. Table 11 provides the statistics for all testing sets in this paper.

	SQuAD	**HotpotQA**	**NQ**
Answerable	11,873	5,901	12,836
Unanswerable	5,945	5,918	2,331
AGent	2,217	2,776	–

Table 11: Data statistics of all testing sets used in this paper. *AGent* refers to the unanswerable questions created using the *AGent* pipeline.

We train all models with batch size of 8 for 2 epochs. The maximum sequence length is set to 384 tokens. We use the AdamW optimizer (Loshchilov and Hutter, 2019) with an initial learning rate of $2 \cdot 10^{-5}$, and $\beta_1 = 0.9$, $\beta_2 = 0.999$. We use a single NVIDIA GeForce RTX 3080 for training and evaluating models.

D Detailed Results of Main Experiments

Table 12 presents a detailed version of our experiments with training six models on SQuAD 2.0, SQuAD *AGent*, and HotpotQA *AGent* and evaluating on SQuAD, HotpotQA, and Natural Questions.

E Unanswerable Examples

Table 13 and 14 present some notable examples of unanswerable questions created using *AGent*.

			SQuAD			HotpotQA			NQ	
			ans	unans	*AGent*	ans	unans	*AGent*	ans	unans
SQuAD 2.0	BERT	base	78.2	70.9	43.6	42.7	84.2	58.2	34.7	53.2
		large	84.5	77.2	46.5	50.1	85.8	61.5	38.7	53.4
	RoBERTa	base	84.5	82.5	54.1	50.0	88.5	59.6	45.1	78.7
		large	85.7	84.6	57.1	50.4	89.5	64.9	46.7	64.7
	SpanBERT	base	85.9	76.8	45.9	56.7	82.4	50.9	50.9	70.0
		large	88.5	83.0	49.1	56.4	87.3	58.8	49.7	43.3
SQuAD *AGent*	BERT	base	83.6	23.6	77.0	58.1	86.6	42.0	30.0	81.2
		large	86.8	28.2	82.0	62.8	91.0	51.6	36.3	68.2
	RoBERTa	base	87.6	29.2	86.2	63.8	91.6	53.8	41.9	90.7
		large	87.3	34.6	86.5	64.9	92.4	56.5	47.8	57.3
	SpanBERT	base	87.2	28.7	75.6	63.3	87.4	45.8	43.2	89.3
		large	89.3	33.5	81.0	66.7	91.1	54.0	47.1	85.3
HotpotQA *AGent*	BERT	base	48.2	45.1	86.3	74.4	99.6	92.2	14.2	98.1
		large	56.6	45.2	87.9	77.1	99.7	96.0	20.0	98.6
	RoBERTa	base	62.8	40.6	82.9	77.7	99.7	97.2	24.8	99.5
		large	62.4	49.2	89.9	79.0	99.7	98.3	35.0	71.0
	SpanBERT	base	58.5	50.4	90.3	78.3	99.7	95.0	23.0	99.2
		large	65.9	46.3	88.4	80.0	99.8	96.8	27.7	98.8
SQuAD *AGent A1* *(Ablation)*	BERT	base	83.3	25.3	–	54.6	91.0	46.3	27.6	78.7
		large	86.3	28.6	–	56.0	91.6	50.4	29.9	81.3
	RoBERTa	base	89.0	28.6	–	61.0	92.3	53.2	41.5	85.4
		large	89.9	33.0	–	65.6	92.4	53.4	44.0	79.4
	SpanBERT	base	87.9	27.4	–	58.7	92.5	52.1	45.5	79.8
		large	88.9	33.2	–	58.3	93.7	58.6	44.0	85.0
SQuAD *AGent A3* *(Ablation)*	BERT	base	85.0	18.8	–	57.2	80.2	36.7	33.5	78.3
		large	87.4	24.8	–	57.1	82.7	41.1	33.8	82.9
	RoBERTa	base	89.1	29.4	–	61.4	87.9	48.1	45.1	89.1
		large	89.7	34.6	–	62.4	91.5	56.9	61.3	23.3
	SpanBERT	base	88.8	24.2	–	62.2	83.3	40.5	47.5	77.7
		large	90.2	30.4	–	59.5	90.3	55.3	49.5	83.5

Table 12: Performance of 6 models fine-tuned on SQuAD 2.0, SQuAD *AGent* and HotpotQA *AGent* evaluated on SQuAD, HotpotQA, and NQ. The term *AGent* (test sets) refers to the unanswerable questions that are created using the *AGent* pipeline. the terms ans and unans stand for answerable and unanswerable, respectively

Unanswerable questions	Reasons
Question: What is the most critical resource measured to in assessing the determination of a Turing machine's ability to solve any given set of problems? **Context**: Many types of Turing machines are used to define complexity classes, such as deterministic Turing machines, probabilistic Turing machines, non-deterministic Turing machines, quantum Turing machines, symmetric Turing machines and alternating Turing machines. They are all equally powerful in principle, but when resources (such as **time or space**) are bounded, some of these may be more powerful than others.	The context provide examples for critical resources but does not specify whether these resources are most critical or not.
Question: What are the specific divisors of all even numbers larger than 2? **Context**: Many questions regarding prime numbers remain open, such as Goldbach's conjecture (that every even integer greater than 2 can be expressed as the sum of **two primes**), and the twin prime conjecture (that there are infinitely many pairs of primes whose difference is 2). [...]	The context provides insights into even numbers and primes, but it does not directly specify the divisors of all even numbers larger than 2.
Question: What is the atomic number for oxygen? **Context**: [...] Dalton assumed that water's formula was HO, giving the atomic mass of oxygen as **8** times that of hydrogen, instead of the modern value of about **16**. [...],	The context only mentions the atomic mass ratio between oxygen and hydrogen. It does not provide information about the atomic number of oxygen.
Question: When did Tesla make these claims? **Context**: [...] In **February 1912**, an article "Nikola Tesla, Dreamer" by Allan L. Benson was published in World Today, in which an artist's illustration appears showing the entire earth cracking in half with the caption, "Tesla claims that in a few weeks he could set the earth's crust into such a state of vibration that it would rise and fall hundreds of feet and practically destroy civilization. A continuation of this process would, he says, eventually split the earth in two.	The context only refers to an article published in February 1912 by Allan L. Benson, which discusses Tesla's claims about setting the earth's crust into vibration. However, it does not explicitly mention when Tesla made the claims.

Table 13: Examples unanswerable questions in SQuAD *AGent*. The spans in **red** are strong plausible answers for the corresponding questions.

Unanswerable questions	Reasons
Question: Keene is an unincorporated community in Wabaunsee County, Kansas, in what federal republic composed of 50 states? **Context**: The **United Mexican States** (Spanish: “Estados Unidos Mexicanos”) is a federal republic composed of 31 states and the capital, Mexico City, an autonomous entity on par with the states. Newbury is an unincorporated community in Wabaunsee County, Kansas, in the United States.	The context mentions the United Mexican States, which is a federal republic composed of 31 states and Mexico City. However, it does not provide any information about a federal republic composed of 50 states.
Question: What was the last date the creator of the NOI was seen by Elijah Muhammad? **Context**: Tynnetta Muhammad [...] wrote articles and columns for the Nation of Islam (NOI) newspaper “Muhammad Speaks”. Having worked as a secretary to Elijah Muhammad, she made it known after his death in 1975 that she was one of his widows. Elijah Muhammad [...] led the Nation of Islam (NOI) from **1934 until his death in 1975.** [...].	The context mentions that Elijah Muhammad led the Nation of Islam from 1934 until his death in 1975, but it does not specify the exact date of the last encounter between the creator of the NOI and Elijah Muhammad.
Question: Polk County Florida’s second most populated city is home to which mall? **Context**: **Lakeland Square Mall** is a shopping mall located on the northern side of Lakeland, Florida in the United States. [...] It is owned and managed by Rouse Properties, one of the largest mall owners in the United States. [...]	The context specifically mentions Lakeland Square Mall, which is located in Lakeland, Florida, but it does not state that Lakeland is the second most populated city in Polk County.
Question: What podcast was the cheif executive officer of Nerdist Industries a guest on? **Context**: Nerdist News [...] was founded and operated by Nerdist Industries’ CEO, Peter Levin, and its CCO, Chris Hardwick. [...] Nerdist Industries was founded as a sole podcast **(The Nerdist Podcast)** created by Chris Hardwick but later spread to include a network of podcasts. [...]	The context mentions the Nerdist Industries CEO, Peter Levin. However, the context does not provide information about a specific podcast where the CEO of Nerdist Industries was a guest.
Question: What book provided the foundation for Masters and Johnson’s research team? **Context**: **Sheep** is a horror novel by British author Simon Maginn, originally published in 1994 and reissued in 1997. [...] William Howell Masters (December 27, 1915 - February 16, 2001) was an American gynecologist, best known as the senior member of the Masters and Johnson sexuality research team. [...]	The context mentions William Howell Masters, who was a prominent member of the Masters and Johnson sexuality research team. However, it does not specify the book that served as the foundation for their research.

Table 14: Examples unanswerable questions in Hotpot *AGent*. The spans in **red** are strong plausible answers for the corresponding questions.