Towards Robust Extractive Question Answering Models: Rethinking the Training Methodology

Anonymous ACL submission

Abstract

This paper proposes a novel training method to improve the robustness of Extractive Question Answering (EQA) models. Previous research has shown that existing models, when trained on EQA datasets that include unanswerable questions, demonstrate a significant lack of robustness against distribution shifts and adversarial attacks. Despite this, the inclusion of unanswerable questions in EQA training datasets is essential for ensuring real-world reliability. Our proposed training method includes a novel loss function for the EQA problem and challenges an implicit assumption present in numerous EQA datasets. Models trained with our method maintain in-domain performance while achieving a notable improvement on out-of-domain datasets. This results in an overall F1 score improvement of 5.7 across all testing sets. Furthermore, our models exhibit significantly enhanced robustness against two types of adversarial attacks, with a performance decrease of only about a third compared to the default models.

1 Introduction

Unanswerable questions are a valuable part in the training datasets of Extractive Question Answering (EQA) models. By learning from these questions, models can can develop the ability to avoid extracting misleading responses, ultimately improving their reliability in real-world applications.

Currently, there are two lines of research on unanswerable questions in EQA. Firstly, Rajpurkar et al. (2018) introduced the SQuAD 2.0 dataset by adding ***adversarial unanswerable questions*** into SQuAD 1.1 (Rajpurkar et al., 2016). This work later inspired similar benchmarks in other languages such as French (Heinrich et al., 2021) and Vietnamese (Nguyen et al., 2022). In the crowdsourcing process for adversarial unanswerable questions, human annotators are typically presented with a triple of context, an answerable question, and its corresponding answer(s). They are then asked to write unanswerable questions that exhibit an adversarial similarity to the presented answerable ones.

In addition to the adversarially-written unanswerable questions, Natural Question (Kwiatkowski et al., 2019), Tydi QA (Clark et al., 2020b), and SQuAD *AGent* (Tran et al., 2023b) propose more naturally constructed unanswerable questions. This category of unanswerable questions is also known as ***information-seeking unanswerable questions***, emerging within the realm of information retrieval. These questions are initially independent of any context. The contexts are then paired with the questions as a result of the attempt to locate answers for the given questions within a large database containing multiple contexts.

The distinct characteristics of these two types of unanswerable questions pose a challenge for models. Models trained with one type of unanswerable questions often struggle when encountering the other type (Sulem et al., 2021; Tran et al., 2023a), defined in Machine Learning as a lack of robustness under distribution shift in the inputs. Additionally, models trained on unanswerable questions also demonstrate a lack of robustness against adversarial attack (Tran et al., 2023b). Notably, models trained on adversarial unanswerable questions in SQuAD 2.0 tend to output an "empty" response upon detecting any sign of contradiction between the attack sentence and the given question.

We hypothesize that the observed lack of robustness in EQA models can be attributed to two primary factors. First, the current EQA training loss objective (Devlin et al., 2019) inaccurately treats unanswerable questions as if they have an answer span. This span is designated to start and end at the special classification token `[CLS]` of the pre-trained model which is also the first token in the input sequence. This approach potentially misguides the model's understanding of unanswerable questions.

Second, the assumption that a given question can only have a single answer or no answer introduces a learning shortcut, making EQA models vulnerable to adversarial attacks.

In this work, we propose a new training method for EQA models to address the two problems discussed above. First, we design new training loss function that naturally treats unanswerable questions as lacking any answer. Second, to overcome the single-answer assumption in most EQA datasets, we create a new "synthetic" answer span in a number of answerable questions. Our empirical findings are summarized as follows:

1. We test our newly proposed training method on three language models. While the new method does not reduce the in-domain performance of models, models fine-tuned with our training method show a 13 F1-score improvement on out-of-domain testing sets. Furthermore, our models exhibit significantly enhanced robustness against two types of adversarial attacks, with a performance decrease of only 13.2 in F1-score compared to a 40.7 decrease of default models.

2. We also investigate the independent contributions of new loss function and "synthetic" answers in our training method. Our analysis reveals that the new loss function helps enhance the robustness against distribution shifts from adversarial unanswerable questions in the training set to information-seeking unanswerable questions in the testing set. On the other hand, eliminating the single-answer assumption by creating "synthetic" answer significantly enhances the robustness of models against adversarial attacks.

2 Related Work

There are two key research areas on improving the robustness of natural language processing (NLP) models: robustness against adversarial attacks and against distribution shift (Wang et al., 2022). Adversarial attacks involve editing a test sample to create a more challenging example for trained models without causing additional difficulty for humans. These attacks can be classified based on whether the attack process has access to the models' parameters (white-box attacks, (Blohm et al., 2018; Neekhara et al., 2019; Alzantot et al., 2018; Wallace et al., 2019; Ebrahimi et al., 2018)) or not (black-box attacks, (Jia and Liang, 2017; Ribeiro et al., 2018; Wang and Bansal, 2018; Blohm et al., 2018; Iyyer et al., 2018)). On the other hand, robustness against distribution shift is measured using test samples that exhibit linguistic differences from the samples encountered by models during the training phase (Miller et al., 2020).

Findings of limited robustness in NLP models have spurred significant efforts to improve their resilience. From a data-driven perspective, adversarial attacks can be employed during the training phase to enhance model robustness. Augmented training data can be created by heuristically editing (Wang and Bansal, 2018) or through neural-based generation (Iyyer et al., 2018; Khashabi et al., 2020a; Bartolo et al., 2021; Fu et al., 2023). Additionally, increasing the diversity of training data has proven to be an effective strategy for improving model robustness (Fisch et al., 2019; Khashabi et al., 2020b).

In addition to data-driven approaches, model-based approaches are also effective in improving model robustness. Following the success of BERT, various studies have shown that the pretraining process, which involves a self-supervised objective and the use of large amounts of diverse pretraining data, significantly enhances the generalization of language models in downstream tasks (Hendrycks et al., 2020; Tu et al., 2020).

Another research direction involves using a biased model during the training phase to force the target model to discard some spurious patterns in the training set. These biased models can be designed with a specific targeted type of bias (Clark et al., 2019; Schuster et al., 2019; He et al., 2019; Utama et al., 2020a; Karimi Mahabadi et al., 2020), or without prior knowledge about the biases present in the training dataset (Clark et al., 2020a; Utama et al., 2020b; Ghaddar et al., 2021; Sanh et al., 2021).

3 Models and Tasks

In Extractive Question Answering (EQA), models are trained to identify the answer (a text span in the context) to the given question. The dataset may include unanswerable questions, for which a valid prediction is an "empty" answer. A common metric to evaluate MRC systems is F1-score. It measures the average overlap between the words in the predicted answer and the human-annotated gold answer.

3.1 Models

In this work, we evaluate our newly proposed training method using the base version of three pre-trained models BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and SpanBERT (Joshi et al., 2020)).

3.2 Extractive Question Answering

An EQA problem is given by a test set $\mathcal{D}$ of triplets (q, c, a) where q is a question posed to models, c is the corresponding context (usually a short paragraph of text), and a is the expected answer (or set of "gold" answers). The performance of the EQA model f is measured by

$$Per(f, \mathcal{D}) = \frac{1}{\mid \mathcal{D} \mid} \sum_{(c,q,a)\in\mathcal{D}} m(a, f(c, q))$$

where m, in this paper, is the F1-score metric.

In our experiments, we evaluate models on both answerable and unanswerable questions from different domains as outlined in the next section. To compare the performance of models accross all tested domains, we assume that (1) the number of answerable questions is equal to the number of answerable questions, and that (2) the importance of different domains is the same.

$$Per(f) = \frac{Per_{has-ans}(f) + Per_{no-ans}(f)}{2}$$

where $Per_{has-ans}(f)$ and $Per_{no-ans}(f)$ are the average performance of model f on all domains of answerable and unanswerable questions, respectively. Specifically, we can calculate $Per_{has-ans}(f)$ as follows:

$$Per_{has-ans}(f) = \frac{1}{\mid \mathcal{S}^{has-ans} \mid} \sum_{\mathcal{D}\in\mathcal{S}^{has-ans}} Per(f, \mathcal{D})$$

, where $\mathcal{S}^{has-ans}$ is the set of all testing set with answerable questions.

3.3 Datasets

In our experiments, we fine-tune our EQA models by conducting additional training on SQuAD 2.0 (Rajpurkar et al., 2018) (for Sections 6 or 7) and SQuAD *AGent* (Tran et al., 2023a) (for Section 7). While both datasets share the same answerable questions, SQuAD 2.0 includes adversarially written unanswerable questions, whereas SQuAD *AGent* utilizes information-seeking unanswerable questions.

We test the performance of our models on

- **SQuAD 2.0**: We test our models on both ***answerable*** (*has-ans*) and ***unanswerable*** (*no-ans*) questions of this dataset. The unanswerable questions in SQuAD 2.0 are adversarially written.

- **SQuAD *AGent***: We only test models on ***unanswerable*** questions (*AGent*) of this dataset because the answerable questions in this dataset are the same as ones in SQuAD 2.0. The unanswerable questions from this dataset are information-seeking.

- **ACE-whQA** (Sulem et al., 2021): We test models on ***answerable*** (*has-ans*) questions and ***two types of unanswerable*** questions: competitive (*no-ans competitive*), where the passage contains an entity of the same type as the expected answer, and non-competitive (*no-ans non-com*), where the passage does not contain any entity of the same type as the expected answer.

The diversity of testing domains enables us to measure the robustness of models against distribution shifts, which occur when encountering testing data that differs from the training data.

4 Adversarial Attacks

In addition to evaluating models' robustness against distribution shift, we also measure the robustness against adversarial attacks.

4.1 Robustness Evaluation

An attack algorithm $\mathcal{A}$ transforms triplets (q, c, a) in $\mathcal{D}$ into adversarial test samples (q', c', a') in the adversarial test set $\mathcal{D}^{\mathcal{A}}_{attacked}$, where c', q', and a' are the modified (attacked) versions of c, q, and a. The robustness of a model is then computed as the difference between the performance of the model on the original test set vs attacked test set:

$$\Delta^{\mathcal{A}} = Per(f, \mathcal{D}) - Per(f, \mathcal{D}^{\mathcal{A}}_{attacked})$$

When there are more than one attack algorithm, we measure the overall robustness by

$$\Delta = \frac{1}{\mid \mathcal{T} \mid} \sum_{\mathcal{A}\in\mathcal{T}} \Delta^{\mathcal{A}}$$

where $\mathcal{T}$ is the set of all tested types of adversarial attacks.

Attack Types	Question	Attacked Context	Ground Truth Answer
AddOneSent *AOS* (Jia and Liang, 2017)	What is the name of the water body that is found to the east?	To the east is the Colorado Desert and the **Colorado River** at the border with Arizona, and the Mojave Desert at the border with the state of Nevada. To the south is the Mexico –United States border. **Sea is the name of the water body that is found to the west.**	**Colorado River**
Negation (Tran et al., 2023b)	What is the name of the water body that is found to the east?	To the east is the Colorado Desert and the **Colorado River** at the border with Arizona, and the Mojave Desert at the border with the state of Nevada. To the south is the Mexico –United States border. **Sea is the name of the water body that is found to the not east.**	**Colorado River**

Table 1: Examples of AddOneSent (*AOS*) and Negation Attacks on answerable questions. The adversarial sentence is highlighted in red color.

4.2 Algorithms for Attack Construction

In this paper, we test the experimented models on two types of adversarial attacks.

4.2.1 AddOneSent Attacks

Table 1 gives an example of AddOneSent (*AOS*) attack (Jia and Liang, 2017). The *AddOneSent attack* strategy creates the attack sentence from a modified question and a fake answer. To construct the modified question, nouns and adjectives in the original question are substituted with their antonyms sourced from WordNet (Fellbaum, 1998). Meanwhile, the fake answer is nearest word to the original gold answer in the vector space of GloVe (Pennington et al., 2014).

4.2.2 Negation Attacks

The Negation Attack, shown in Table 1, is designed to mislead models into giving incorrect "empty" predictions. This method involves the crafting of an attack statement that has significant lexical overlap with the original question yet is easy to identify as contradictory by simply inserting "not" in front of the first adjective within the question. The fake answer is created similarly to the AddOneSent attack.

The questions and answers are unchanged in both types of attacks ($q' = q$ and $a' = a$).

5 Extractive Question Answering Loss Functions

EQA models are typically fed a question q and a context c as input. State-of-the-art EQA models, often employing BERT-style language models at their core, process q and c together as a sequence input $<$ `[CLS]`q`[SEP]`$c >$, with `[CLS]` and `[SEP]` as special tokens of pre-trained tokenizer accompanying the pre-trained model.

Given an input sequence (pair of question-context) with n tokens $seq = (t_1, t_2, ..., t_n)$, we have

$$\mathcal{M}(seq) = (\vec{v_1}, \vec{v_2}, ..., \vec{v_n})$$

where $\mathcal{M}$ is a pre-trained language model that takes sequence seq as the input and output n contextualized vectors $(\vec{v_1}, \vec{v_2}, ..., \vec{v_n})$, each corresponds to one of the input tokens, encoding its contextual information. Note that the dimension d_v of each vector $\vec{v_k}$ is predetermined by the specifications of pre-trained language model $\mathcal{M}$.

We then employ two single-layer feed-forward neural networks, denoted as S and E for predicting the start and end positions, respectively. Both networks are designed to receive input vectors $\vec{v_k}$ of dimension d_v and produce a scalar output of dimension 1. We then have that

$$s_k = S(\vec{v_k}), \quad e_k = E(\vec{v_k})$$

for every $\vec{v_k}$ in $(\vec{v_1}, \vec{v_2}, ..., \vec{v_n})$.

5.1 Default Loss Function

Devlin et al. (2019) use the Cross Entropy loss function for training BERT on SQuAD 2.0.

$$L_{Default} = -\Sigma_{k=1}^{n} \log \frac{\exp(s_k)}{\Sigma_{i=1}^{n} \exp(s_i)} y_k^s - \Sigma_{k=1}^{n} \log \frac{\exp(e_k)}{\Sigma_{i=1}^{n} \exp(e_i)} y_k^e$$

where y_k^s and y_k^e are the labels of whether k^{th} token in the input sequence is the start or end of a gold

Train Set: SQuAD 2.0		SQuAD			ACE-whQA			Average		Overall
		has-ans	no-ans	*AGent*	has-ans	no-ans non-com	no-ans competitive	has-ans	no-ans	
BERT	Default	**78.8**	71.1	44.2	67.6	52.3	**38.7**	**73.2**	51.6	62.4
	Ours	73.7	**75.7**	**63.2**	**69.9**	**59.1**	36.6	71.8	**58.7**	**65.3**
RoBERTa	Default	**85.0**	81.2	51.8	66.0	77.1	57.8	**75.5**	67.0	71.3
	Ours	81.3	**85.6**	**67.9**	**67.4**	**85.3**	**66.3**	74.4	**76.3**	**75.4**
SpanBERT	Default	**86.0**	76.0	46.0	**66.0**	53.1	24.2	**76.0**	49.8	62.9
	Ours	80.2	**81.9**	**66.1**	61.5	**90.5**	**60.4**	70.9	**74.7**	**72.8**
Average	Default	**83.3**	76.1	47.3	**66.5**	60.8	40.2	**74.9**	56.1	65.5
	Ours	78.4	**81.1**	**65.7**	66.3	**78.3**	**54.4**	72.4	**69.9**	**71.2**

Table 2: Performance of models fine-tuned on SQuAD 2.0 using Default training method and our proposed training method, each averaged over five runs with random initialization. The performance on in-domain samples are highlighted in gray cells.

answer identified by human annotators. Unanswerable questions are treated as having an answer span with start and end at the [CLS] token, which means y_0^s and y_0^e are 1s.

As of the time of writing this paper, the training methodology utilizing this particular loss function remains widely adopted in most EQA models. We term this training methodology the "default" approach.

5.2 Our Loss Function

QA Loss

This component (L_{QA}) of the newly proposed loss function is similar to the Cross Entropy loss function used in work by Devlin et al. (2019). However, a key difference lies in how we handle unanswerable sequences. In our approach, all tokens within an unanswerable sequence are assigned the same label, represented as $y_k^s = y_k^e = \frac{1}{n}$, where n denotes the sequence length.

Sequence Tagging Loss

$$L_{Tag} =$$
$$- \Sigma_{k=1}^{n} (y_k^s \log \sigma(s_k) + (1 - y_k^s) \log(1 - \sigma(s_k)))$$
$$- \Sigma_{k=1}^{n} (y_k^e \log \sigma(e_k) + (1 - y_k^e) \log(1 - \sigma(e_k)))$$

where $\sigma(x) = \frac{1}{1+\exp(-x)}$, the labels for the gold start tokens are assigned $y_k^s = 1$, and labels for all other tokens are set to $y_k^s = 0$. This logic extends to the labels for end tokens. Consequently, all y_k^s and y_k^e in unanswerable sequences are zeros.

Overall Loss

$$L_{Ours} = \lambda_{QA} \cdot L_{QA} + \lambda_{Tag} \cdot L_{Tag}$$

where λ_{QA} and λ_{Tag} denote weights for their corresponding losses. In this paper, we set $\lambda_{QA} = 2$ and $\lambda_{Tag} = 1$. Appendix A discusses the selection of these weights in more detail.

5.3 Inference Pipeline

In both model types, the score for a candidate span ranging from position i to position j is given by $s_i + e_j$, The span with the highest score, where $j \geq i$, is selected for prediction.

Models trained with the default training loss function indicate an unanswerable question by outputting an "empty" string when the highest scoring span is $(0, 0)$, which corresponds to the [CLS] token.

Conversely, models trained with our method indicate an "empty" string response when the maximum span score of $s_i + e_j$ is negative.

6 Experiments

6.1 Experiment Design

In the experiments in this section, we train our models using the SQuAD 2.0 dataset. For models trained with the default loss function, the original SQuAD 2.0 dataset is used without modifications. However, for models trained using our proposed method in this section, we introduce modifications to the SQuAD 2.0 dataset to eliminate the single-answer assumption during the training phase. We augment approximately 20% of the answerable questions in the original dataset with an additional "synthetic" answer, resulting in these questions having two answers. Further details are presented in Appendix B.

6.2 Results

Table 2 shows performances of models trained on default and our training methods. Firstly, models trained with our method (new loss function and additional synthetic answers) achieve almost the same performance as those trained using default approach on SQuAD 2.0, the in-domain testing set. Specifically, models trained with the default loss function achieve an average F1 score of 79.7 (across both answerable and unanswerable questions $\frac{83.3+76.1}{2}$) on SQuAD 2.0, while our models achieve an average F1 score of 79.8.

On the other hand, our models consistently outperform default model on out-of-domain unanswerable questions, including those from SQuAD *AGent* and both competitive and noncompetitive unanswerable questions from ACE-whQA. On information-seeking unanswerable questions from SQuAD *AGent*, our models outperform default models by a large margin of 18.4 F1 score on average. Furthermore, on the unanswerable questions in ACE-whQA, our models outperform default ones by 17.5 F1 for noncompetitive unanswerable questions and 14.2 F1 for competitive ones. This enhanced robustness against distribution shifts enables our models to attain a higher overall performance of 71.2, compared to the 65.5 achieved by default models across all evaluated answerable and unanswerable questions.

We then analyze the performance gap of each model on unanswerable questions between SQuAD 2.0 and SQuAD *AGent* over three training epochs. Figure 1 presents the dynamics of this performance gap for RoBERTa models trained with the default method and our proposed method on SQuAD 2.0.

Notably, models using the default loss function exhibit an increasing performance gap throughout the training process. This indicates that as models better perform on adversarial unanswerable questions within SQuAD 2.0, their performance on information-seeking unanswerable questions in SQuAD *AGent* decreases significantly. Conversely, models trained with our proposed loss function demonstrate a stable robustness against such shifts across three training epochs.

In addition to evaluating the generalization of our models, we also evaluate their robustness against adversarial attacks. The results, presented in Table 3, demonstrate the improved robustness of models trained with our method compared to those trained with the default approach. Specifically, under the AddOneSent attacks, the performance of default models drops by 27.4, whereas our models exhibit a much smaller decrease of 9.9 F1 score. Similarly, for the Negation attack, while default models experience a performance decrease of 56.3, our models see a reduction of only 16.4 on F1. These results highlight the significantly improved robustness of our models, with our training method mitigating 67.6% of the performance drop due to adversarial attacks, reducing from 40.7 to 13.2 on F1-score metric.

Train Set: SQuAD 2.0		Original	**Adversarial Attack**		$\Delta\downarrow$
			AOS	Negation	
BERT	Default	**78.8**	52.2	27.5	38.9
	Ours	73.7	**64.0**	**49.5**	**16.9**
RoBERTa	Default	**85.0**	56.1	30.9	41.5
	Ours	81.3	**71.9**	**65.8**	**12.4**
SpanBERT	Default	**86.0**	57.9	30.7	41.7
	Ours	80.2	**69.5**	**70.6**	**10.1**
Average	Default	**83.3**	55.4	29.7	40.7
	Ours	78.4	**68.5**	**62.0**	**13.2**

Table 3: Robustness against adversarial attacks of models fine-tuned on SQuAD 2.0 using Default training method and our proposed training method.

7 Further Analysis

7.1 Experiment Design

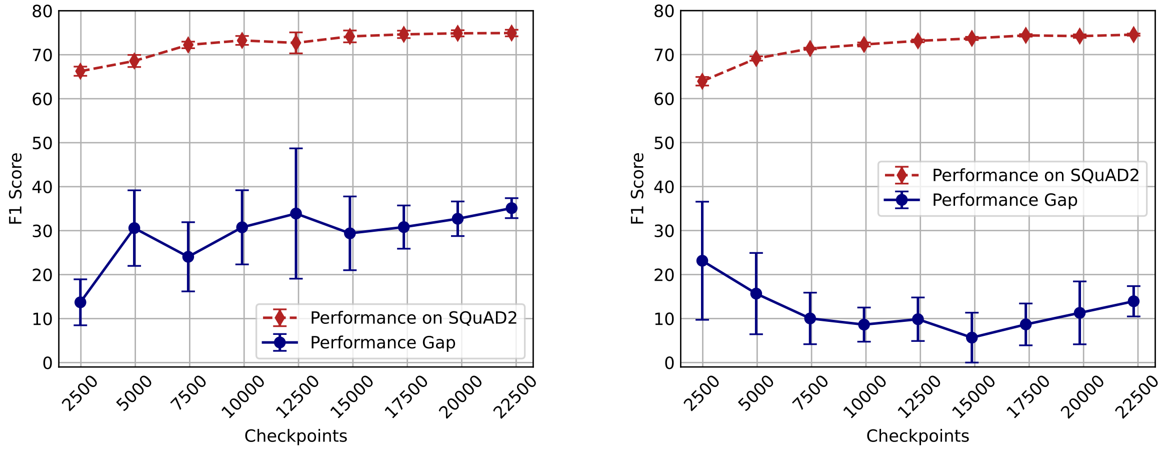
To evaluate the effectiveness of our proposed training method under different scenarios, we design two experiments.

1. We train models on SQuAD 2.0 using our proposed loss function without introducing "synthetic" answers. We then compare these models (referred to as "*no synthetic*") with those trained using the default loss function, also trained on SQuAD 2.0. This experiment is designed to study the independent contributions of the newly proposed loss function and the augmented "synthetic" answers to the robustness of our models.

2. We train models on the information-seeking, unanswerable question dataset SQuAD *AGent* using our proposed training method (including new loss function and "synthetic" answers). We then compare these models with those trained using the default method, also trained on SQuAD *AGent*. This experiment investigates the effectiveness of our proposed method on datasets with information-seeking unanswerable questions.

7.2 Robustness against Distribution Shift

We now evaluate the performance of models trained on SQuAD 2.0 using our proposed loss function,



(a) Default

(b) Ours

Figure 1: The training dynamics of RoBERTa models trained using the Devlin method versus our proposed method on SQuAD 2.0. We analyze the performance gap on unanswerable questions between SQuAD 2.0 and SQuAD *AGent* across three training epochs. The error bars represent the standard deviations of five runs.

Train Set:* *SQuAD 2.0		SQuAD has-ans	SQuAD no-ans	SQuAD *AGent*
BERT	Default	**78.8**	71.1	44.2
	no synthetic	76.4	**74.8**	**60.4**
RoBERTa	Default	**85.0**	81.2	51.8
	no synthetic	83.5	**83.4**	**63.1**
SpanBERT	Default	**86.0**	76.0	46.0
	no synthetic	82.2	**80.8**	**61.5**
Average	Default	**83.3**	76.1	47.3
	no synthetic	80.7	**79.7**	**61.7**

Table 4: Performance of models fine-tuned on SQuAD 2.0 using Default training method and our proposed training method but without augmented synthetic answers, each averaged over five runs with random initialization. The performance on in-domain samples are highlighted in gray cells.

Train Set:* *SQuAD AGent		SQuAD has-ans	SQuAD no-ans	SQuAD *AGent*
BERT	Default	**83.7**	23.4	75.6
	Ours	80.3	**30.1**	**81.2**
RoBERTa	Default	**87.7**	30.2	84.4
	Ours	85.7	**35.7**	**88.8**
SpanBERT	Default	**87.3**	28.6	76.5
	Ours	83.6	**36.6**	**86.0**
Average	Default	**86.2**	27.4	78.8
	Ours	83.2	**34.1**	**85.3**

Table 5: Performance of models fine-tuned on SQuAD *AGent* using Default training method and our proposed training method, each averaged over five runs with random initialization. The performance on in-domain samples are highlighted in gray cells.

while excluding synthetic answers. The experimental results, in Table 4, highlight that even in the absence of synthetic answers, our models better generalize to information-seeking unanswerable questions. The "*No synthetic*" outperform default models by a large margin of 18.4 on F1 when tested on *AGent* unanswerable questions. This finding shows that the robustness of our models can be mainly attributed to the incorporation of the new loss function.

Having established the successful generalization of our models from adversarial to information-seeking unanswerable questions, we now investigate the effectiveness of our loss function in achieving the reverse (generalizing from SQuAD *AGent* to SQuAD 2.0).

Table 5 shows the performance of models trained on SQuAD *AGent* using default and our training methods. We observe that models trained with our method do not exhibit improved robustness against distribution shift to unanswerable questions in SQuAD 2.0, compared to those trained with the default method. This result indicates that our loss function mainly benefits the generalization of models to information-seeking unanswerable questions, such as those in SQuAD *AGent*.

7.3 Robustness against Adversarial Attacks

While models trained with our method on SQuAD *AGent* do not exhibit improved robustness against distribution shifts to SQuAD 2.0, they demonstrate significant improvements when encountering adversarial attacks.

Train Set: SQuAD AGent		Orig	Adversarial Attack		Δ ↓
			AOS	Negation	
BERT	Default	**83.7**	61.0	44.5	30.7
	Ours	80.3	**67.0**	**57.1**	**18.3**
RoBERTa	Default	**87.7**	68.6	46.4	30.2
	Ours	85.7	**75.4**	**64.4**	**15.8**
SpanBERT	Default	**87.3**	66.8	37.4	35.2
	Ours	83.6	**72.2**	**65.9**	**14.6**
Average	Default	**86.2**	65.5	42.8	30.0
	Ours	83.2	**71.5**	**62.5**	**16.2**

Table 6: Robustness of models fine-tuned on SQuAD *AGent* using Default training method and our proposed training method.

The experimental results in Table 6 show that when using SQuAD *AGent* as the training set, models trained with default approach exhibit a significant reduction in performance of 30.0 F1 points. Conversely, models trained with our method (new loss function and the synthetic answers) experience a much smaller performance drop of 16.2 F1 points. Our findings conclusively demonstrate that our training method notably enhances the robustness of models trained on both SQuAD 2.0 and SQuAD *AGent* against adversarial attacks.

Train Set: SQuAD 2.0		Orig	Adversarial Attack		Δ ↓
			AOS	Negation	
BERT	Default	**78.8**	**52.2**	**27.5**	38.9
	no synthetic	76.4	49.6	26.3	**38.4**
RoBERTa	Default	**85.0**	**56.1**	**30.9**	41.5
	no synthetic	83.5	55.0	30.1	**40.9**
SpanBERT	Default	**86.0**	**57.9**	**30.7**	**41.7**
	no synthetic	82.2	53.0	22.5	44.4
Average	Default	**83.3**	**55.4**	**29.7**	**40.7**
	no synthetic	80.7	52.5	26.3	41.3

Table 7: Robustness of models fine-tuned on SQuAD 2.0 using Default training method and our proposed training method.

With this significant improvement established, we then shift our focus to identifying the primary factor behind this increased robustness. We hypothesize that our models' robustness against adversarial attacks might be mainly thanks to the augmented "synthetic" answers, which eliminate the single-answer assumption in the SQuAD dataset.

Therefore, we examine the robustness against adversarial attacks of "*no synthetic*" models trained on SQuAD 2.0 using our proposed loss function, while omitting synthetic answers. The experimental results, in Table 4, indicate that without the synthetic answers, our models are no longer robust against adversarial attacks. The performance gap Δ of our models without synthetic answers is even higher than that of default models (41.3 compared to 40.7). This finding strongly supports our hypothesis that the inclusion of "synthetic" answers in our training method is a key factor in the improved robustness against adversarial attacks of our models.

8 Conclusion

In this paper, we introduce a novel training methodology for EQA models aimed at enhancing their robustness against distribution shifts and adversarial attacks. Our new training method is characterized by a novel training loss for the EQA problem, as well as challenging the single-answer assumption by creating a new "synthetic" answer span in a number of answerable questions. Our experimental findings demonstrate that models trained using our approach exhibit significant improvement on out-of-domain testing datasets. Furthermore, the robustness of these models against two tested types adversarial attacks is also significantly better than that of the default models.

In Section 7, we also study the independent contributions of our new loss function and the augmented "synthetic" answers to the robustness of our models. Our analysis reveals that the new loss function specifically benefits the performance on information-seeking unanswerable questions. This improved performance of information-seeking unanswerable questions contribute to the robustness against distribution shifts of models trained on SQuAD 2.0 with our method.

On the other hand, our training method challenges the single-answer assumption of many existing EQA datasets by creating "synthetic" answers for a number of answerable questions. Our experiments indicate that these "synthetic" answers significantly contribute to the robustness of models trained with our method on both SQuAD 2.0 and SQuAD *AGent* against adversarial attacks. This finding strongly corroborates our initial hypothesis, suggesting that the longstanding single-answer assumption of many EQA training datasets is a learning shortcut for models that can significantly compromise their robustness. We believe this work highlights the importance of future Question Answering datasets that incorporate the possibility of multiple, non-contiguous answer spans, similar to the MultiSpanQA dataset (Li et al., 2022).

Limitations

We acknowledge certain limitations in our work. Our study primarily focuses on evaluating the proposed training methodology using multiple pre-trained transformers-based models in English. This does not guarantee that our method will maintain its effectiveness when applied to other languages.

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.

Max Bartolo, Tristan Thrush, Robin Jia, Sebastian Riedel, Pontus Stenetorp, and Douwe Kiela. 2021. Improving question answering model robustness with synthetic adversarial data generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8830–8848, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Matthias Blohm, Glorianna Jagfeld, Ekta Sood, Xiang Yu, and Ngoc Thang Vu. 2018. Comparing attention-based convolutional and recurrent neural networks: Success and limitations in machine reading comprehension. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 108–118, Brussels, Belgium. Association for Computational Linguistics.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2019. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. 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 4069–4082, Hong Kong, China. Association for Computational Linguistics.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2020a. Learning to model and ignore dataset bias with mixed capacity ensembles. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3031–3045, Online. Association for Computational Linguistics.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020b. 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.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press.

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.

Deqing Fu, Ameya Godbole, and Robin Jia. 2023. SCENE: Self-labeled counterfactuals for extrapolating to negative examples. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7832–7848, Singapore. Association for Computational Linguistics.

Abbas Ghaddar, Phillippe Langlais, Mehdi Rezagholizadeh, and Ahmad Rashid. 2021. End-to-end self-debiasing framework for robust NLU training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1923–1929, Online. Association for Computational Linguistics.

He He, Sheng Zha, and Haohan Wang. 2019. Unlearn dataset bias in natural language inference by fitting the residual. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 132–142, Hong Kong, China. Association for Computational Linguistics.

Quentin Heinrich, Gautier Viaud, and Wacim Belblidia. 2021. Fquad2.0: French question answering and knowing that you know nothing.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. 2020. Pretrained transformers improve out-of-distribution robustness. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2744–2751, Online. Association for Computational Linguistics.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. 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 1875–1885, New Orleans, Louisiana. Association for Computational Linguistics.

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.

Rabeeh Karimi Mahabadi, Yonatan Belinkov, and James Henderson. 2020. End-to-end bias mitigation by modelling biases in corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8706–8716, Online. Association for Computational Linguistics.

Daniel Khashabi, Tushar Khot, and Ashish Sabharwal. 2020a. More bang for your buck: Natural perturbation for robust question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 163–170, Online. Association for Computational Linguistics.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020b. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, Online. Association for Computational Linguistics.

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.

Haonan Li, Martin Tomko, Maria Vasardani, and Timothy Baldwin. 2022. MultiSpanQA: A dataset for multi-span question answering. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1250–1260, Seattle, United States. Association for Computational Linguistics.

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.

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.

Paarth Neekhara, Shehzeen Hussain, Shlomo Dubnov, and Farinaz Koushanfar. 2019. Adversarial reprogramming of text classification neural networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5216–5225, Hong Kong, China. Association for Computational Linguistics.

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.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

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.

Victor Sanh, Thomas Wolf, Yonatan Belinkov, and Alexander M Rush. 2021. Learning from others' mistakes: Avoiding dataset biases without modeling them. In *International Conference on Learning Representations*.

Tal Schuster, Darsh Shah, Yun Jie Serene Yeo, Daniel Roberto Filizzola Ortiz, Enrico Santus, and Regina Barzilay. 2019. Towards debiasing fact verification models. 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 3419–3425, Hong Kong, China. 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, Gia-Huy Do, Phong Nguyen-Thuan Do, Matt Kretchmar, and Xinya Du. 2023a. Agent: A novel pipeline for automatically creating unanswerable questions.

Son Quoc Tran, Phong Nguyen-Thuan Do, Uyen Le, and Matt Kretchmar. 2023b. 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.

Lifu Tu, Garima Lalwani, Spandana Gella, and He He. 2020. An empirical study on robustness to spurious correlations using pre-trained language models. *Transactions of the Association for Computational Linguistics*, 8:621–633.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. 2020a. Mind the trade-off: Debiasing NLU models without degrading the in-distribution performance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8717–8729, Online. Association for Computational Linguistics.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. 2020b. Towards debiasing NLU models from unknown biases. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7597–7610, Online. Association for Computational Linguistics.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. 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 2153–2162, Hong Kong, China. Association for Computational Linguistics.

Xuezhi Wang, Haohan Wang, and Diyi Yang. 2022. Measure and improve robustness in NLP models: A survey. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4569–4586, Seattle, United States. Association for Computational Linguistics.

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.

A Derivation on Unanswerable Sequence

Let us consider the k^{th} token in an ***unanswerable*** sequence. Our objective is to ensure that the logit s_k generally decreases if $s_k \geq 0$ after each training batch. To achieve this, we need the partial derivative of L_{Ours} with respect to the start score s_k of the k^{th} token, i.e. $\frac{\lambda_{Tag} \partial L_{Tag}}{\partial s_k} + \frac{\lambda_{QA} \partial L_{QA}}{\partial s_k}$, remains positive whenever $s_k \geq 0$.

It is established that the partial derivative of the tagging loss L_{Tag} with respect to the score s_k, $\frac{\partial L_{Tag}}{\partial s_k}$, is positive. Nonetheless, there is no assurance that the partial derivative of the question-answering loss L_{QA} with respect to s_k, $\frac{\partial L_{QA}}{\partial s_k}$, will also be positive.

Firstly, we assume that both Tagging weight λ_{Tag} and Question Answering weight λ_{QA} are positive. We then have that

$$
\begin{aligned}
&\lambda_{Tag} \frac{\partial L_{Tag}}{\partial s_k} \\
&= -\lambda_{Tag} \frac{d}{ds_k} \left[\log(1 - \frac{1}{1 + \exp(-s_k)}) \right] \\
&= -\lambda_{Tag} \frac{\frac{d}{ds_k} \left[1 - \frac{1}{1+\exp(-s_k)} \right]}{1 - \frac{1}{1+\exp(-s_k)}} \\
&= -\lambda_{Tag} \frac{\frac{d}{ds_k} [1 + \exp(-s_k)]}{(1 + \exp(-s_k))^2 (1 - \frac{1}{1+\exp(-s_k)})} \\
&= \lambda_{Tag} \frac{\exp(-s_k)}{(1 + \exp(-s_k))^2 - (1 + \exp(-s_k))} \\
&= \lambda_{Tag} \frac{1}{1 + \exp(-s_k)} = \lambda_{Tag} (\frac{\exp(s_k)}{1 + \exp(s_k)})
\end{aligned}
$$

$$
\begin{aligned}
&\lambda_{QA} \frac{\partial L_{QA}}{\partial s_k} \\
&= \lambda_{QA} \frac{\partial}{\partial s_k} \left[-\Sigma_{k=1}^{n} \log \frac{\exp(s_k)}{\Sigma_{i=1}^{n} \exp(s_i)} y_k^s \right] \\
&= \lambda_{QA} \frac{\partial}{\partial s_k} \left[-\Sigma_{k=1}^{n} \log \frac{\exp(s_k)}{\Sigma_{i=1}^{n} \exp(s_i)} \frac{1}{n} \right] \\
&= \frac{\lambda_{QA}}{n} \left(\frac{(n-1)\exp(s_k)}{\Sigma_{i=1}^{n} \exp(s_i)} \right. \\
&\qquad \left. - \frac{\Sigma_{i=1}^{n} \exp(s_i) - \exp(s_k)}{\Sigma_{i=1}^{n} \exp(s_i)} \right) \\
&= \frac{\lambda_{QA}}{n} \left(\frac{n \exp(s_k)}{\Sigma_{i=1}^{n} \exp(s_i)} - 1 \right) \\
&= \lambda_{QA} \left(-\frac{1}{n} + \frac{\exp(s_k)}{\Sigma_{i=1}^{n} \exp(s_i)} \right) > -\frac{\lambda_{QA}}{n}
\end{aligned}
$$

Because $s_k \geq 0$, we know that $\frac{\exp(s_k)}{1+\exp(s_k)} \geq \frac{1}{2}$. Therefore, we can derive that

$$
\begin{aligned}
&\lambda_{Tag} \frac{\partial L_{Tag}}{\partial s_k} + \lambda_{QA} \frac{\partial L_{QA}}{\partial s_k} \\
&\qquad > \lambda_{Tag} (\frac{\exp(s_k)}{1 + \exp(s_k)}) - \frac{\lambda_{QA}}{n} \\
&\qquad \geq \frac{\lambda_{Tag}}{2} - \frac{\lambda_{QA}}{n}
\end{aligned}
$$

Consequently, the partial derivative of the overall loss (L_{Ours}) with respect to the score s_k, $\frac{\partial L_{Ours}}{\partial s_k}$, will be positive whenever $s_k \geq 0$ if the ratio of $\frac{\lambda_{Tag}}{\lambda_{QA}} > \frac{2}{n}$. In our experiments, the number of tokens in a question-context sequence is set to $n = 384$. We set $\lambda_{Tag} = 1$ and $\lambda_{QA} = 2$. Therefore, $\frac{\lambda_{Tag}}{\lambda_{QA}} = \frac{1}{2} > \frac{2}{384}$.

B Synthetic Answers

Table 8 illustrates the incorporation of Synthetic answers into the context 20% of the answerable questions within the training set, serving as an example of our augmentation approach.

Incorporating "synthetic" answers into contexts of answerable questions involves three steps:

1. Creating fake answers that differ from the ground truth answers annotated by human crowdsource workers.

   (a) We re-match each answerable question with 10 new contexts.

   (b) We train 10 models on SQuAD 2.0 and obtain their predictions on the re-matched question-context pairs.

   (c) For each answerable question, we extract the answer span that is most frequently predicted by the models.

   In this step, we ensure that the extracted spans are different from the corresponding ground truth answers, with F1 score lower than 0.2. Through this method, we can extract relevant and plausible answers that can serve as "synthetic" answers for the corresponding questions.

2. Given the fake answer and the original question, we use ChatGPT-turbo3.5 to convert them into a natural statement. We use the

Types	Question	Attacked Context	Ground Truth Answer
Original	In 1948, what general assembly resolution established genocide as a prosecutable act?	[...] Lemkin successfully campaigned for the universal acceptance of international laws defining and forbidding genocides. In 1948, the UN General Assembly adopted the ***Convention on the Prevention and Punishment of the Crime of Genocide (CPPCG)*** which defined the crime of genocide for the first time. [...]	***Convention on the Prevention and Punishment of the Crime of Genocide (CPPCG)***
With "synthetic" answer	In 1948, what general assembly resolution established genocide as a prosecutable act?	[...] Lemkin successfully campaigned for the universal acceptance of international laws defining and forbidding genocides. **In 1948, Resolution 46/3 established genocide as a prosecutable act**. In 1948, the UN General Assembly adopted the ***Convention on the Prevention and Punishment of the Crime of Genocide (CPPCG)*** which defined the crime of genocide for the first time. [...]	***Convention on the Prevention and Punishment of the Crime of Genocide (CPPCG)*** **Resolution 46/3**

Table 8: An example of "synthetic" answers.

prompt:

```
Given the question and its answer,
write a statement:
Example:
  <example1>
  <example2>
Question: <question>
Answer: <answer>
Statement: ...
```

3. We then insert the newly created statement into the original context at a random position between existing sentences. We utilize SpaCy's pipeline [1] to perform sentence boundary detection on original contexts.

C Details for Models Training

The input of a question-context pair into the pre-trained model is in the form of [CLS]<*Question*>[SEP]<*Context*>, with [CLS] and [SEP] as special tokens 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 classifiers.

We train all models with batch size of 8 for 3 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.

[1] https://github.com/explosion/spaCy