TASA: Deceiving Question Answering Models by Twin Answer Sentences Attack

Yu Cao[1*], Dianqi Li[2], Meng Fang[3], Tianyi Zhou[4], Jun Gao[5], Yibing Zhan[6], Dacheng Tao[1]

[1]School of Computer Science, The University of Sydney
[2]University of Washington [3]University of Liverpool [4]University of Maryland
[5]Harbin Institute of Technology, Shenzhen [6]JD Explore Academy
ycao8647@uni.sydney.edu.au, dianqili@uw.edu
Meng.Fang@liverpool.ac.uk, zhou@umiacs.umd.edu
jgao95@stu.hit.edu.cn, zhanyibing@jd.com, dacheng.tao@gmail.com

Abstract

We present **T**win **A**nswer **S**entences **A**ttack (TASA), an adversarial attack method for question answering (QA) models that produces fluent and grammatical adversarial contexts while maintaining gold answers. Despite phenomenal progress on general adversarial attacks, few works have investigated the vulnerability and attack specifically for QA models. In this work, we first explore the biases in the existing models and discover that they mainly rely on keyword matching between the question and context, and ignore the relevant contextual relations for answer prediction. Based on two biases above, TASA attacks the target model in two folds: (1) lowering the model's confidence on the gold answer with a *perturbed answer sentence*; (2) misguiding the model towards a wrong answer with a *distracting answer sentence*. Equipped with designed beam search and filtering methods, TASA can generate more effective attacks than existing textual attack methods while sustaining the quality of contexts, in extensive experiments on five QA datasets and human evaluations.

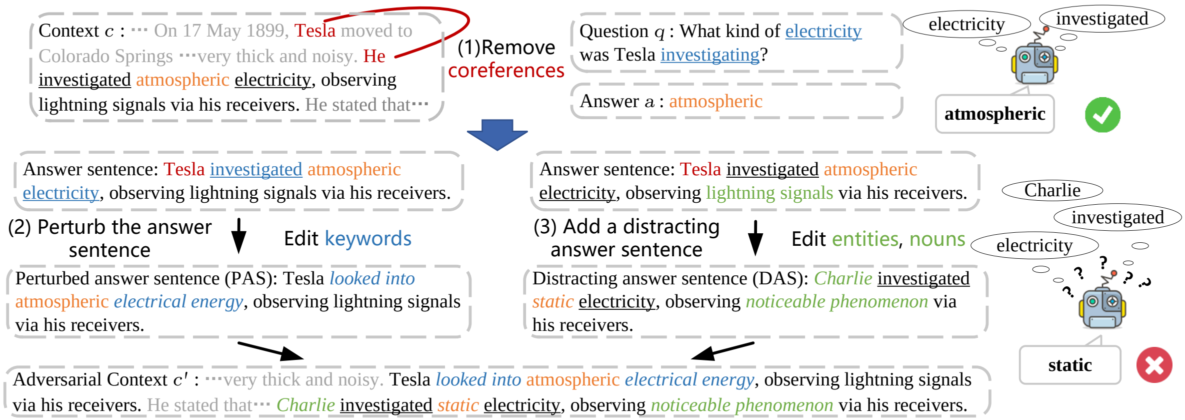
1 Introduction

Question Answering (QA) is the cornerstone of various NLP tasks. In extractive QA (the most common setting), given a question and an associated context, a QA model needs to comprehend on the context and predict the answer (Rajpurkar et al., 2016). While most works keep improving the answer correctness on benchmarks (Devlin et al., 2019; Yu et al., 2018), few studies investigate the robustness of QA models, e.g., is the performance achieved by sound contextual comprehension or via shortcuts like keyword matching? Although adversarial attacks attract growing interests in computer vision (Goodfellow et al., 2014; Zhao et al., 2018) and recently in NLP (Ren et al., 2019; Li et al., 2021), most of them study general tasks without taking into account the properties of QA. The vulnerability and biases of models can lead to catastrophic failures outside the benchmark datasets. An effective way to study them is through adversarial attacks specifically designed for QA tasks.

Generating adversarial textual examples is challenging due to the discrete syntactic restriction, especially on QA, where the additional relationship between question and context should be further considered. Existing works such as AddSent and Human-in-the-loop (Jia and Liang, 2017; Wallace et al., 2019b) heavily rely on human annotators to create effective adversarial QA examples, which are costly and hard to scale. A few studies (Gan and Ng, 2019; Wang et al., 2020; Wallace et al., 2019a) can generate adversarial samples automatically. But they only perturb either the context or the question separately, and thus ignore the consistency between them. Moreover, the major pitfalls of QA models' detailed comprehension process are not fully investigated, confining producing more powerful adversarial attacks.

In this paper, we develop an adversarial attack specifically targeting two biases of mainstream QA models discussed in §2: (1) making prediction via keywords matching in the answer sentence of contexts; and (2) ignorance of the entities shared between the question and context. Our method, **T**win **A**nswer **S**entences **A**ttack (TASA), automatically produces black-box adversarial attacks (Papernot et al., 2017) perturbing a context without hurting its fluency or changing the gold answer. TASA firstly allocates the answer sentence in the context that is decisive for answering (Chen and Durrett, 2019) and then modify it into two sentences targeting the two biases above: one sentence preserves the gold answer and the meaning but replaces the keywords that are shared with the question with their synonyms; while the other leaves the keywords and

[*] Work was done when Yu Cao was an intern at JD Explore Academy.



Context c : ⋯ On 17 May 1899, Tesla moved to Colorado Springs ⋯very thick and noisy. He investigated atmospheric electricity, observing lightning signals via his receivers. He stated that⋯

Question q : What kind of electricity was Tesla investigating?

Answer a : atmospheric

Answer sentence: Tesla investigated atmospheric electricity, observing lightning signals via his receivers.

Answer sentence: Tesla investigated atmospheric electricity, observing lightning signals via his receivers.

Perturbed answer sentence (PAS): Tesla *looked into* atmospheric *electrical energy*, observing lightning signals via his receivers.

Distracting answer sentence (DAS): *Charlie* investigated *static* electricity, observing *noticeable phenomenon* via his receivers.

Adversarial Context c' : ⋯very thick and noisy. Tesla *looked into* atmospheric *electrical energy*, observing lightning signals via his receivers. He stated that⋯ *Charlie* investigated *static* electricity, observing *noticeable phenomenon* via his receivers.

Figure 1: An example of TASA generating adversarial context C'. Underlined parts indicate keywords. Orange indicates gold answer or *pseudo answer*. Other colors indicate tokens for perturbation, distracting, or coreferences.

the syntactic structure intact but changes the entities (subjects/objects) associated with the answer. Thereby, the former is a *perturbed answer sentence* (PAS) lowering the focus of the model on the gold answer, while the latter generates a *distracting answer sentence* (DAS) as Jia and Liang (2017) to further misguide the model towards a wrong answer with respect to irrelevant entities. Thus, the adversarial context can substantially distort the QA model without changing the answer for humans. To address the challenge of efficiency and textual fluency, we further propose beam search and filtering techniques empowered by pretrained models.

In experiments, we evaluate TASA and other adversarial attack baselines on attacking three popular contextualized QA models, BERT (Devlin et al., 2019), SpanBERT (Joshi et al., 2020), and BiDAF (Seo et al., 2017), on five extractive QA datasets, i.e., SQuAD 1.1, NewsQA, NaturalQuestions, HotpotQA, and TriviaQA. Experimental results and human evaluations consistently show that TASA achieves better attack capability than other baselines and meanwhile preserves the textual quality and gold answers identifiable by humans.

Our contributions are three-fold:

- We propose a novel adversarial attack method "TASA" specifically designed to fool extractive QA models while retain the gold answers for humans.
- We study the biases and vulnerability of QA models that motivate TASA, and demonstrate that those models mainly rely on keyword matching, while may ignore the contextual relation.
- Experiments on five QA benchmark datasets and three types of victim models demonstrate that TASA outperforms existing baselines on attack performance, as well as the comparable capability to preserve textual quality and answers.

We release our code at `https://github.com/caoyu-noob/TASA`

2 Predicting Bias in Question Answering

Recent works show that state-of-the-art natural language inference models often overly rely on certain keywords as shortcuts for prediction (Wallace et al., 2019a; Sinha et al., 2021). In the empirical study of this section, we illustrate that current QA models consistently exhibit such bias on the sensitive words without leveraging the contextual relationship for predicting answers.

We analyze two mainstream QA models with contextualized comprehension capabilities, BERT (Devlin et al., 2019) and BiDAF (Seo et al., 2017), trained on the original training set of SQuAD1.1 (Rajpurkar et al., 2016) and tested on samples modified from its validation set. We define the sentence in the context that contains the gold answer as the **answer sentence**, which is the key for predicting answers (Chen and Durrett, 2019). We first compare the performance of models on the original sample with only answer sentence as the context ("*Only answer sent.*"). Besides, to investigate the bias on sensitive words, we further examine models on samples with various types of sensitive words in the answer sentence being (1) either removed ("*Remove*") or (2) only retained ("*Only*"). There are three types of sensitive words to be considered:

(1) Entities. The same named entities shared between the answer sentence and the question.

(2) Lexical words (lexical.). with lexical meanings (excluding all named entities) shared between the answer sentence and question. They cover the words with POS tags of *NOUN*, *VERB*, *ADJ*, etc.

(3) Function words (func.). Words that do not

Answer sentence: The annual NFL Experience was held at the Moscone Center located in San Francisco.
Question: In what city is the Moscone Center located?

***Remove* entities**	The annual NFL Experience was held at the located in San Francisco.
***Only* entities**	Moscone Center San Francisco.
***Remove* lexical.**	The annual NFL Experience held at the Moscone Center in San Francisco.
***Only* lexical.**	Was located San Francisco.
***Remove* func.**	The annual NFL Experience was held at Moscone Center located San Francisco.
***Only* func.**	The in San Francisco.

Figure 2: The illustration of removing or only retaining (*Only*) different types of sensitive words, the answer is underlined and kept.

Model	BERT		BiDAF	
	EM	F1	EM	F1
Original	80.91	88.23	65.72	75.97
Only answer sent.	+2.79	+2.87	+3.27	+4.37
Remove entities	-5.39	-4.17	-4.84	-6.27
Only entities	-23.42	-15.90	-26.75	-18.03
Remove lexical.	-18.62	-16.71	-24.43	-24.46
Only lexical.	-5.27	-1.81	+0.86	+4.28
Only lexical. (shuffle)	+4.07	+2.94	+10.68	+10.46
Remove func.	-5.20	-3.18	-5.42	-3.55
Only func.	-24.08	-22.34	-22.24	-22.72

Table 1: EM and F1 scores of BERT and BiDAF models on different modified samples compared to results on the original samples. Shuffle means the best results among texts whose tokens are random-ordered.

have lexical meaning but are shared between the answer sentence and the question. They include words with POS tags of *DET*, *ADP*, *PRON*, etc.

When modifying the answer sentence, we only remove or retain these three types of sensitive words, except the gold answer words, and also keep the rest context intact. As shown in Figure 2, the modified texts are unreadable and difficult to infer their true meaning from the human perspective. In addition, we follow UNLI (Sinha et al., 2021) to **Shuffle** tokens in the answer sentence for *Only* lexical. conditions, verifying the possibility of models to achieve even better performance, given the texts are totally ungrammatical but contain sensitive words.

Table 1 compares the evaluation results on different modifications. Given the answer-sentence-only context, the performance of both BERT and BiDAF are improved, indicating that they mainly rely on the answer sentence and almost ignore the rest of the context. While removing entities or function words causes a slight difference in metrics, removing lexical words leads to a larger performance drop. In addition, both models perform surprisingly satisfactory when keeping only lexical words in answer sentences, compared to the 30% $\sim$ 60% drop when keeping other words. Moreover, shuffling tokens under the lexical-only conditions even possibly benefit the model despite the answer sentence being merely discrete tokens and hard to read. This suggests that both models can answer questions solely relying on the shared lexical words (not contextual), i.e., *keywords* in the answer sentence, regardless of the word order and other contextual information like entities.

Inspired by this observation, we question that whether we can utilize the discovered pitfall to design an efficient adversarial attack method specifically for QA? Can we lower the model's attention on the gold answers and then misguide it to incorrect answers by manipulating the existing sensitive keywords in the context and adding some new misleading ones? The answer is affirmative: we show that the predictions can be shifted to crafted wrong answers in §4.4.

3 Methodology

We propose an adversarial attack method for extractive QA models, **T**win **A**nswer **S**entences **A**ttack (TASA), which automatically produces black-box attacks solely based on the final output of the **victim QA model** $F(\cdot)$. Given a typical QA sample composed of a **context** c, a **question** q, and an **answer** a (i.e., a positional span in c), we study how to **perturb the context** c as c' that can deceive $F(\cdot)$ towards producing an incorrect answer $F(c', q) \neq a$, while c' retains the correct answer a that can be identified by humans. We keep the question q intact to ensure the answer a valid, as editing the short q with simple syntactic structure easily alters its meaning.

TASA can be summarized as three main steps: (1) Remove coreferences in the context to facilitate the following edits; (2) Perturb the *answer sentence* by replacing *keywords* (overlapped sensitive lexical words in §2) with synonyms to produce a *perturbed answer sentence* (PAS), lowering the model's focus on the gold answer; (3) Add a *distracting answer sentence* (DAS) that keeps the *keywords* intact but changes the associated subjects/objects to misguide the model for producing a wrong answer, which can be proven in Table 5. How these the three steps are applied is illustrated in Figure 1. And Algorithm 1 gives the complete procedure of TASA.

3.1 Remove Coreferences

Coreference relations across sentences commonly exist in texts (Hobbs, 1979) and also bring extra challenges to adversarial attacks during making substitutions on target words. For example, in a sentence "*His patented AC induction motor were licensed*", "*His*" refers to "*Nikola Tesla's*" according to the whole context. However, given the single sentence, it is hard to precisely allocate candidates for substitution "*his*" as it is a pronoun. Instead, we remove the coreference by replacing such pronouns with the entity names they refer to, e.g., specific persons or locations, so we can edit them directly without considering a complicated coreference.

3.2 Perturb the Answer Sentence

According to the former analysis, the *answer sentence* is the most important part of context c for QA tasks, and QA models usually predict answers according to keyword matching (Chen and Durrett, 2019). Hence, we first study how to obtain a perturbed answer sentence (PAS) by only perturbing those sensitive *keywords* instead of changing the whole context. Given the gold answer a, we first allocate the *answer sentence* s_a in c. In TASA, we use the text matching to search for s_a that contains text a.

Determine the keywords to perturb. As discussed in §2, QA models normally rely on *keywords* to make predictions. Hence, we directly perturb those keywords rather than randomly-selected tokens as previous works (Ren et al., 2019; Jin et al., 2020) to produce more effective attacks. We adopt three criteria to select words of s_a into the keyword set $\mathcal{X}$: (1) they are not included in the answer span a so the gold answer will retain; and (2) each of them shares the same lemma with a token in the question q; and (3) each keyword's POS tag belongs to a POS tag set for lexical words, e.g., *NOUN*, *ADJ*, etc.

Rank keywords by importance. Following previous works (Jin et al., 2020), we rank keywords in $\mathcal{X}$ according to their importance scores in the descending order. Given the original context c and answer a, the importance score I_i of $x_i \in \mathcal{X}$ is

$$I_i = p_F(a|c, q) - p_F(a|mask(c, x_i), q), \qquad (1)$$

where $p_F(a|\cdot)$ denotes the probability of the original span position of gold answer a predicted by the victim model F, $mask(c, x_i)$ means c is modified by replacing a token x_i with a special mask symbol, e.g., given $c = ..x_{i-1}x_ix_{i+1}..$, $mask(c, x_i) = ..x_{i-1} < mask > x_{i+1}...$ Finally, we obtain a set $\mathcal{X}$ of keywords ranked by their importance.

Generate perturbed answer sentence (PAS). Following the order in $\mathcal{X}$, we edit each keyword $x_i \in \mathcal{X}$ one after another. Specifically, we replace x_i with its synonym r_j from a synonym set and transform the inflection of r_j as the same as x_i, e.g., we change "*Tesla investigated...*" to "*Tesla looked into...*" where "*investigated*" is a keyword and "*look into*" is one of its synonyms.

Thereby, multiple PASs are obtained during editing each keyword if more than one synonym exists. We retain the top few of them via beam search and filtering strategy (as elaborated in §3.4) to promote the effectiveness as well as efficiency, resulting in a set of PASs $\mathcal{P}$, which will be the initial texts of the next perturbation turn. While PASs do not change the meaning of texts as they replace words with their synonyms, they will distract the model, which relies on keyword matching, away from PAS containing the answer.

3.3 Add a Distracting Answer Sentence

To further deceive the model, we also add a distracting answer sentence (DAS) at the end of the context. In particular, DAS is modified from the *answer sentence* s_a as well: it changes the subjects/objects and the answer, but keeps the keywords intact which can draw models' attention. Collaborating with PAS, DAS misguides models to predict incorrect answers regarding wrong subjects/objects due to the pitfall studied in §2, which will be verified in Table 5. Our method differs from previous works (Jia and Liang, 2017; Wallace et al., 2019a) as our distractors are added automatically and suits more general conditions.

Determine the tokens to edit. Similar to PAS, the first step of generating DAS is to select a set $\mathcal{Y}$ of tokens from the s_a as the candidates of subjects/objects that will be edited. In TASA, each selected token $y \in \mathcal{Y}$ needs to meet all the following criteria: (1) $y \notin \mathcal{X}$ so the original keywords are preserved; (2) $y \notin a$ (as we will process the answer tokens separately); (3) y is a named entity or its POS tag is NOUN. The goal of (3) is to extract and change the subjects/objects of s_a to produce a pseudo answer sentence that contains incorrect answers. We do not use a syntactic parser to locate the subjects/objects, as we find it less accurate and effective than POS tags empirically.

Generate distracting answer sentence (DAS). Similar to PAS, we edit each $y_i \in \mathcal{Y}$ to obtain a DAS. Specifically, we replace each y_i with a token/phrase of the same entity/noun type, e.g., "*Tesla investigated...*" can be modified to "*Charlie investigated...*" since both "*Tesla*" and "*Charlie*" are persons. In principle, (1) if y_i is a named entity, we randomly sample N different entities with the same NER tag from the whole corpus as the candidates to replace y_i; (2) otherwise, we randomly sample N nouns with the same hypernym as y_i from the corpus for substitution. Hence, multiple DASs can be generated, and we also use the beam search strategy to only choose the top few of them, resulting in a set of DASs $\mathcal{D}$.

Change the answer in DAS. Since the main purpose of DAS is to misguide the model to predict a wrong answer, we entirely replace the text span of the original answer in DAS with a pseudo answer, which helps to remove the ambiguity of the answer from humans' perspective. Specifically, we replace every lexical token of a in DAS with one of pseudo answer token candidates that share the same NER tags or POS tags, which are randomly sampled from the whole corpus. Likewise, this procedure results in multiple results and thus a beam search is also necessary for the efficiency and attack success purpose as well.

Algorithm 1 TASA

```
Definition: Beam size $M$, importance score $I_i$ given in Eq. 1,
    effect score $E_n$ given in Eq. 2, threshold $T_E$ for $E_n$
Input: QA sample $(c, q, a)$, victim model $F(\cdot)$
Output: An adversarial context $c'$ to fool $F(\cdot)$
Remove coreferences in $c$;
Extract answer sentence $s_a$ from $c$;
$\mathcal{X} \leftarrow$ keywords in $s_a$ and rank them by $I_i$;
Initialize the PAS set: $\mathcal{P} \leftarrow \{s_a\}$
for $1 \leq i \leq |\mathcal{X}|$ do
    $\mathcal{P} \leftarrow$ perturb $x_i$ for each item in $\mathcal{P}$;
    $\mathcal{P} \leftarrow M$ items in $\mathcal{P}$ with the highest $E_n$;
    if $T_E \leq$ minimum $E_n$ in $\mathcal{P}$ then break;
    end if
end for
$\mathcal{P} \leftarrow$ filter $\mathcal{P}$ based on answerable and quality in §3.4;
Initialize the DAS set: $\mathcal{D} \leftarrow \{(s_j, c_j)\}$, each DAS $s_j = s_a$, paired with context $c_j$ modified by each PAS in $\mathcal{P}$;
$\mathcal{Y} \leftarrow$ a set of tokens in $s_a$ to be edited for DAS;
for $1 \leq i \leq |\mathcal{Y}|$ do
    $\mathcal{D} \leftarrow$ edit $y_i$ for each DAS $s_j$ in $\mathcal{D}$;
    $\mathcal{D} \leftarrow M$ items in $\mathcal{D}$ with the highest $E_n$;
end for
$\mathcal{D} \leftarrow$ edit answer tokens for each DAS $s_j$ in $\mathcal{D}$;
$(s_b, c_b) \leftarrow$ The item in $\mathcal{D}$ with the highest $E_n$;
$c' \leftarrow$ append DAS $s_b$ to the end of context $c_b$;
return $c'$;
```

3.4 Beam Search and Filtering

Beam search. When editing each word in generating the PAS and DAS, there usually exist multiple replacement candidates, resulting in multiple perturbed sentences. In order to obtain the one that has the greatest potential leading to a successful attack, and to improve the attack's efficiency, we apply a beam search strategy defined based on the effect score E_n for each perturbed sentence s_n.

$$E_n = p_F(a|c, q) - p_F(a|edit(c, s_n), q), \qquad (2)$$

where $edit$ denotes that the original context c is modified by s_n: (1) if s_n is a PAS, it replaces the original answer sentence s_a in c; (2) if s_n is a DAS, it is appended to the end of c. These edited texts will be ranked by E_n in the descending order, and only the top M (M is beam size) are retained for the next edit step. Beam search will stop if (1) no additional edit is needed for the current sample, or (2) the minimum effect score among the result is higher than a threshold T_E that can ensure sufficient performance drop.

TASA runs beam search for PAS to obtain a PAS set $\mathcal{P}$, then obtain a DAS set $\mathcal{D}$ sequentially, and finally generate the adversarial context c'. Note that we obtain a DAS based on a series of contexts that are already perturbed by $\mathcal{P}$. So each item in $\mathcal{D}$ is a pair of a DAS s_j and a corresponding perturbed context c_j, and the initial $\mathcal{D}$ contains all possible contexts edited by each PAS in $\mathcal{P}$.

Filtering by textual quality. To ensure high textual quality and answer preservation of the generated adversarial contexts, TASA applies a filtering procedure on the M (beam size) PASs achieved after the final beam search for generating PAS. We skip it for DASs as they have no effect on the gold answer. In particular, we firstly use a model to justify whether the question q is still answerable given the perturbed context $edit(c, s_n)$. Such a model can be a large-scale pretrain model fine-tuned on both answerable and unanswerable samples (refer to Appendix A.2 for details). Only those contexts classified as *answerable* will remain. In addition, we further constrain the remained contexts' textual quality in terms of semantic similarity and fluency:

$$U_n = USE(s_n, s_a) - PPL(s_n)/PPL(s_a), \quad (3)$$

where USE denotes the USE similarity (Cer et al., 2018) between two sentences and PPL denotes the perplexity computed by a GPT2 model (Radford et al., 2019). Only s_n fulfilling $U_n \geq T_U$ (T_U as a threshold) are retained for the next step.

4 Experiments

We evaluate TASA on extractive QA tasks. We begin by details of setup (§4.1), then introduce the main results in §4.2, followed by ablation studies in §4.3 and additional analysis in §4.4 to better illustrate each module in our method.

4.1 Setup

Datasets. We evaluate the QA adversarial attacks generated by TASA using 5 extractive QA datasets: SQuAD 1.1 (Rajpurkar et al., 2016), NewsQA (Trischler et al., 2017), Natural Questions (NQ) (Kwiatkowski et al., 2019), HotpotQA (Yang et al., 2018), and TriviaQA (Joshi et al., 2017). We use the settings from MRQA (Fisch et al., 2019) for the latter four datasets, more details are given in Appendix A.4. We report results on their dev sets, as not all their test sets are publicly available.

Victim models. We attack three QA models, i.e., BERT (Devlin et al., 2019), SpanBERT (Joshi et al., 2020), and BiDAF (Seo et al., 2017), in our experiments. The former two are on the top of pretrained BERT$_{\text{base}}$ and SpanBERT$_{\text{large}}$ respectively. Both of them benefit from huge corpora, where SpanBERT can also be regarded as one of the SOTA models for general extractive QA tasks. The latter BiDAF is an end2end model based on LSTM and bidirectional attention specially designed for extractive QA (Related results are provided in Appendix B.1 as it is not a SOTA model). All models output the start and end positions of the answer span in the context as the prediction.

Implementation. Given a dataset, we firstly train each kind of models on its training set to get a model achieving satisfactory performance on the dev set. The trained model is then used as a victim model $F(\cdot)$ and we perform an adversarial attack using all samples from the **whole dev** set. We use a beam size $M = 5$ for TASA. The synonym set used for PAS is obtained by unionizing two sources, i.e., (1) **WordNet** synonym dictionary (Fellbaum, 2010) and (2) **PPDB 2.0** dataset containing token-level paraphrase pairs (Mrkšic et al., 2016). More details about TASA can be found in Appendix A.2.

Baselines. We consider the following 2 strong baselines[1] besides the original dev set (Original).

- **TextFooler** (Jin et al., 2020): A general token-level attack method using synonyms derived from counter-fitting word embeddings. We directly apply it to the context c to make perturbations and use the model's prediction $F_a(\cdot)$ on the gold answer to determine whether to stop attacking.
- **T3** (Wang et al., 2020): A tree-autoencoder-based method to obtain perturbed sentences for attacking. It can be directly applied to QA by adding a distracting sentence to the context.

Both of them and our TASA are **black-box** attack methods without using the internal parameters of victim models. We also include human-annotated **AddSent** adversarial data (Jia and Liang, 2017) for SQuAD 1.1, as they share the same contexts.

Evaluation metrics. Following the former works (Rajpurkar et al., 2016; Wang et al., 2020; Li et al., 2021), we evaluate attack methods using the following metrics: 1) **EM**, the exact match ratio of predicted answers; 2) **F1**, the F1 score between the predicted answers and the gold answers. Lower EM and F1 mean better attack effectiveness; 3) **Grammar error (GErr)**, the context grammatical error numbers given by LanguageTool[2] following Zang et al. (2020), we use the average value per 100 tokens due to various context lengths among datasets; 4) **PPL**, the average perplexity of all adversarial contexts given by a small sized GPT2 model (Radford et al., 2019) to measure their fluency (Kann et al., 2018). Lower values of **GErr** and **PPL** indicate better textual quality.

4.2 Main Results

The main experimental results on BERT and SpanBERT are summarized in Table 2. TASA achieves the overall best performance among all methods. In particular, it shows the best capability to deceive models than others on 3 datasets and the comparable best results on NewsQA and TriviaQA, where it causes more drops on EM and F1 metrics compared to baselines. It means the combination of PAS and DAS is more efficient than solely editing tokens or adding distracting text. Noticeably, all methods have fair attack effect for datasets with longer contexts, e.g., NewsQA and TriviaQA, because limited numbers of token-level perturbations or adding a single sentence causes fewer impacts on long texts. Besides, SpanBERT is more robust with slight accuracy declines due to its larger scale and superior pre-training strategy.

In terms of textual quality, TASA achieves the overall lowest **PPL** and comparable low values

[1] We run the codes provided by the original papers to get results. We use the *black-box* and *targeted* config for T3.

[2] https://languagetool.org/

Victim model		**BERT-base**					**SpanBERT-large**				
Dataset	**method**	**EM↓**	**F1↓**	**GErr↓**	**PPL↓**	**Num**	**EM↓**	**F1↓**	**GErr↓**	**PPL↓**	**Num**
SQuAD 1.1	Original	80.91	88.23	2.39	33.25	10,570	88.25	94.00	2.39	33.25	10,570
	AddSent*	57.78	64.58	2.47	33.98	3,560	73.88	79.77	2.47	33.98	3,560
	TextFooler	67.18	78.18	**2.95**	44.84	7,919	80.00	88.08	**3.03**	44.57	7,746
	T3	71.63	78.86	3.48	44.45	9,622	76.76	82.13	3.66	42.30	9,761
	OURS	**40.06**	**50.87**	2.98	**41.15**	9,559	**54.18**	**65.50**	3.09	**40.31**	9,580
NewsQA	Original	51.57	65.57	1.98	22.50	4,212	58.78	73.81	1.98	22.50	4,212
	TextFooler	43.31	58.34	**2.14**	24.33	3,727	52.13	68.25	**2.16**	24.80	3,685
	T3	**39.54**	53.49	2.33	**22.86**	3,865	51.29	66.40	2.23	22.89	3,875
	OURS	39.62	**53.46**	2.16	**22.86**	2,860	**49.96**	**64.93**	2.18	**22.77**	2,872
NQ	Original	67.39	79.28	20.48	49.74	12,836	71.74	83.12	20.48	49.74	12,836
	TextFooler	48.31	63.08	20.46	49.02	7,158	55.47	69.49	**20.37**	45.67	7,252
	T3	60.06	71.20	20.93	60.90	10,439	57.92	70.03	20.78	63.21	10,446
	OURS	**43.23**	**55.32**	**20.42**	**44.30**	8,809	**51.08**	**64.84**	20.40	**15.24**	8,829
HotpotQA	Original	56.89	75.70	3.73	17.01	5,901	63.29	81.60	3.73	17.01	5,901
	TextFooler	33.59	47.76	4.01	20.52	5,397	60.49	78.94	4.04	20.96	5,369
	T3	30.45	42.08	4.81	21.17	5,669	53.89	70.55	4.80	20.96	5,583
	OURS	**27.01**	**39.10**	**3.99**	**17.29**	5,345	**44.18**	**60.50**	**3.99**	**17.24**	5,355
TriviaQA	Original	58.61	65.42	3.74	24.42	7,785	67.51	74.38	3.74	24.42	7,785
	TextFooler	52.51	57.39	4.30	25.85	7,307	**63.62**	69.95	**3.81**	25.78	7,358
	T3	51.85	56.06	4.06	**24.49**	7,543	64.12	**69.62**	4.07	**24.50**	7,549
	OURS	**51.50**	**54.23**	**3.81**	24.69	7,092	63.86	69.97	3.85	24.65	7,103

Table 2: Main results on 5 QA datasets. The best results are in **bold**. **Num** is the sample number of a dataset or generated adversarial samples from the whole dataset by a method. ↓ means that the lower value is the better. *: samples are annotated by humans.

on **GErr**. TextFooler usually has the lowest GErr values, as it makes pure token-level perturbation that generates fewer sentence-level unnatural errors. While T3 always generates sentence-level distractors that are meaningless without a complete syntactic structure, resulting in worse performance on GErr and PPL. TASA fulfills trade-off attacks on both token and sentence levels, avoiding significant textual quality loss.

It is also worth mentioning that TASA is better than AddSent at fooling models. Despite having a better textual quality by adding human-annotated distracting texts, samples in AddSent does not perturb the influential part of the original context, limiting its effects on making attacks.

Human evaluation. We randomly sample 150 sets of adversarial samples, each containing 3 samples generated by TextFooler, T3, and TASA originated respectively from the same sample in SQuAD 1.1, using BERT as the victim model. Each set is evaluated in two aspects: (1) Answer preservation, whether the gold answer of a sample remains unchanged; (2) Textual quality, ranking the quality ($1 \sim 3$) of the context based on the fluency and grammaticality. Totally 63 non-expert annotators are involved, and related results are summarized in Table 3. Although TASA is weaker than T3 in answer preservation as T3 always retains the original part of the context, it is equivalent to Textfooler and both of them have a significantly better textual quality than T3 due to the reason we have concluded before. Such a comparable sample quality is sufficient to verify the superiority of TASA, considering its much stronger capability to deceive models (Refer to Appendix C for qualitative adversarial samples by TASA).

Methods	TextFooler	T3	TASA
Answer preservation	79.9±4.5	85.9±3.3	79.1±4.7
Avg. quality rank	1.52±0.06	2.64±0.07	1.83±0.06

Table 3: Human evaluation results on SQuAD 1.1 (answer preservations are in percentage). ± indicates the confidence intervals with a 95% confidence level.

4.3 Ablation Studies

We verify the effectiveness of each key module in TASA by: 1) *w/o remove coref.*: without removing coreferences; 2) *w/o PAS*: without applying perturbed answer sentence; 3) *w/o DAS*: without adding distracting answer sentence. The upper part of Table 4 proves their contributions. It can be found that *remove coref.* slightly benefits the quantity of suitable attack samples, while both PAS and DAS make vital contributions to successful attacks and feasible numbers of adversarial samples.

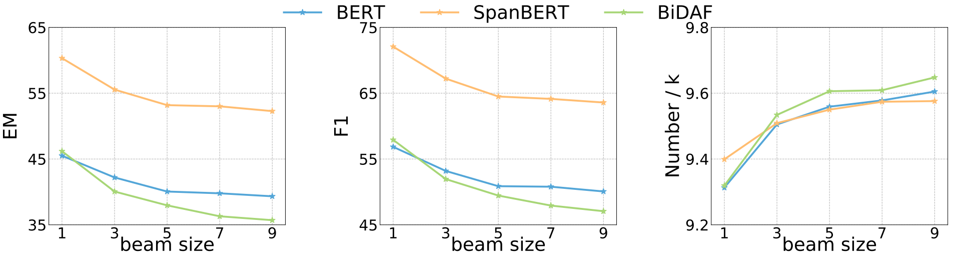


Figure 3: The EM, F1 and quantities of adversarial samples using different beam size on three victim models.

Modules	**EM↓**	**F1↓**	**GErr↓**	**PPL↓**	**Num**
TASA	40.06	50.87	2.98	41.15	9,559
w/o remove coref.	39.95	50.39	2.96	41.13	9,374
w/o PAS	59.63	70.91	2.73	35.89	8,709
w/o DAS	54.13	67.68	3.03	53.39	5,646
w/o importance	41.44	52.32	3.01	41.94	9,564
w/o quality	38.70	49.18	3.36	44.46	9,654
Only use WordNet	43.19	54.12	3.00	41.15	9,262
Only use PPDB	45.08	56.35	2.91	37.19	9,482
w/o edit answer	57.63	68.91	2.86	37.00	9,559
Only NEs	40.79	51.88	3.10	42.95	8,822
Only nouns	43.95	55.45	3.34	45.46	7,426

Table 4: Results of TASA ablation studies on SQuAD 1.1 dataset using BERT as the victim model.

We then do ablations on PAS, including: 1) *w/o importance*: without ranking keywords and edit them randomly; 2) *w/o quality*: without filtering perturbed texts using quality index U_n; 3) *Only use WordNet* as the synonym source; and 4) *Only use PPDB* as the synonym source. Based on the middle part of Table 4, *w/o importance* slightly lower the overall performance. Despite *w/o* quality can promote the attack success rate, it introduces extra textual quality degeneration. Besides, more synonym sources mean a larger search space, so we introduce both WordNet and PPDB into TASA.

Ablations on DAS are finally conducted, 1) *w/o pseudo answer*: do not change answers in DASs; 2) *Only NE* and 3) *Only nouns*: only edit named entities/nouns. Related results are given in the lower Table 4. The obvious change on *w/o pseudo answer* illustrates that changing the original answer in DASs is crucial for attacking, also proving DAS can shift models' focus from the original answer sentence as they can still derive the gold answer from DASs. Moreover, involving various editing types, including both NE and nouns, benefit the attack effectiveness and generated sample quantity.

4.4 More Analysis

Effect of beam size. We vary the beam size during generating PASs and DASs to investigate its influence. Figure 3 reports the cahnges of EM, F1, and quantities of adversarial samples. Clearly, a larger beam size leads to better performance and more diverse adversarial samples. Naturally, the larger the beam size also means the slower speed. Thus, we use $M = 5$ for a trading off of performance and efficiency, as we see limited performance gains from beam sizes larger than 5.

Datasets	SQuAD 1.1	NewsQA	NQ	Hotpot	Trivia
BERT	39.19	20.95	36.22	36.15	24.78
SpanBERT	33.20	20.92	32.14	38.09	27.71

Table 5: F1 score of predicted answers and pseudo answers, on adversarial samples from TASA with DASs.

Shift to the pseudo answers. Since DAS aims to misguide the attention of models from the original answer sentences to them, we expect QA models to output the pseudo answers contained in DASs. Table 5 shows the F1 scores between the predicted answers and the pseudo answers on all adversarial samples that include DAS from 5 datasets. The results demonstrate that there are high overlaps between incorrect predictions by victim models and pseudo answers, as these values are close to the performance drops caused by adversarial samples, confirming that DASs can draw attention from models to make incorrect predictions.

Adversarial training. To verify the effectiveness of TASA in improving the robustness of QA models, we randomly replace training data in SQuAD 1.1 with corresponding adversarial samples generated by TASA in varied ratios, and then use the new training data to fine-tune a BERT model. The performance on the original dev set, the adversarial dev set generated by TASA, and samples from AddSent, is shown in Figure 4, where different mixing ratios are used, Noticeably, with a suitable mixture ratio, adversarial samples from TASA can make models more robust under adversarial attacks without significant performance loss on the original data. Interestingly, this defense capability can also be transferable to other adversarial

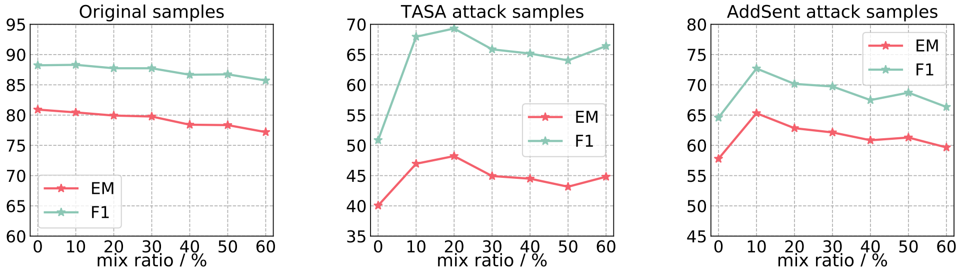


Figure 4: The performance of BERT model fine-tuned on the original SQuAD data **mixed with adversarial samples from TASA** in different ratios, evaluated on the original dev samples, adversarial samples from TASA and AddSent. We expect a slight influence on original ones, while promotions on the latter two kinds of samples.

data, e.g. AddSent. Such results verify the potential of TASA to enhance the current QA models.

5 Related Work

Question answering. Extractive QA is the most common QA task, where the answer is a text span in the supporting context. There are various datasets, e.g., SQuAD, NewsQA, and NaturalQuestions (Rajpurkar et al., 2016, 2018; Trischler et al., 2017; Kwiatkowski et al., 2019), motivating more works on QA models, such as end2end models like BiDAF, R-Net, QANet and so on (Seo et al., 2017; Wang et al., 2017; Yu et al., 2018). Pretrained models are widely applied recently, such as BERT, RoBerta, and SpanBert (Devlin et al., 2019; Liu et al., 2019; Joshi et al., 2020). They realize remarkable promotions benefited from huge corpora, meanwhile they can also be used as backbones to solve more complex QA tasks (Cao et al., 2019; Huang et al., 2021). Nevertheless, there are more concerns (Sinha et al., 2021; Ettinger, 2020; Wallace et al., 2019a) whether models can really capture contextual information rather than using token-level knowledge simply.

Textual adversarial attack. Textual adversarial attack has been widely investigated in general tasks like text classification and natural language inference (NLI). Some works use character-level misspelled tokens to attack models, but are easy to be defended (Liang et al., 2018; Ebrahimi et al., 2018; Li et al., 2019; Pruthi et al., 2019). More studies use more sophisticated toke-level perturbations (Ren et al., 2019; Alzantot et al., 2018; Zang et al., 2020; Li et al., 2021) or phrase/sentence-level editing (Iyyer et al., 2018; Chen et al., 2021; Lei et al., 2022) to produce adversarial texts, with some filtering strategies to guarantee the text meaning and quality. However, none of them shows their effectiveness on QA tasks.

There are some efforts on attacking QA models. AddSent (Jia and Liang, 2017) contains adversarial samples with distracting sentences added by human annotators. Wallace et al. employ human testers to interact with models and realize dynamic attacks. Despite showing their effectiveness, these approaches are not extensible and limited in scale. There are also automatic methods. T3 (Wang et al., 2020) utilizes a Tree LSTM to obtain a distracting sentence based on the skeleton of the question. Universal Trigger (Wallace et al., 2019a) find input-agnostic texts that deceive models for a specific question type via gradient-guided search. Our TASA differs from them as it bridges contexts and questions to attack more efficiently and suits more general conditions.

6 Conclusion

We present TASA, an automatic adversarial attack method for QA models. It generates twin answer sentences, perturbed answer sentence (PAS), and distracting answer sentence (DAS), to construct a new adversarial context in a QA sample. It can deceive models and misguide them to an incorrect answer based on their pitfalls that overly rely on matching sensitive keywords during predicting answers. In experiments, TASA achieves remarkable attack performance on five datasets and three victim models with satisfactory sample quality. Our additional analysis also proves that it is possible to get more robust QA models via TASA in the future.

Acknowledgements

This work is supported in part by Australian Research Council FL-170100117, IC-190100031, and LE-200100049.

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 EMNLP 2018*, pages 2890–2896.

Yu Cao, Meng Fang, and Dacheng Tao. 2019. BAG: Bi-directional attention entity graph convolutional network for multi-hop reasoning question answering. In *Proceedings of NAACL-HLT 2019*.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Céspedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Jifan Chen and Greg Durrett. 2019. Understanding dataset design choices for multi-hop reasoning. In *Proceedings of NAACL-HLT 2019*, pages 4026–4032.

Yangyi Chen, Jin Su, and Wei Wei. 2021. Multi-granularity textual adversarial attack with behavior cloning. In *Proceedings of the EMNLP 2021*, pages 4511–4526.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT 2019*, pages 4171–4186.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. Hotflip: White-box adversarial examples for text classification. In *Proceedings of ACL 2018*, pages 31–36.

Allyson Ettinger. 2020. What bert is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Christiane Fellbaum. 2010. Wordnet. In *Theory and applications of ontology: computer applications*, pages 231–243. Springer.

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.

Wee Chung Gan and Hwee Tou Ng. 2019. Improving the robustness of question answering systems to question paraphrasing. In *Proceedings of ACL 2019*, pages 6065–6075.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Jerry R Hobbs. 1979. Coherence and coreference. *Cognitive science*, 3(1):67–90.

Yinya Huang, Meng Fang, Yu Cao, Liwei Wang, and Xiaodan Liang. 2021. DAGN: Discourse-aware graph network for logical reasoning. In *Proceedings of NAACL-HLT 2021*.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the NAACL-HLT 2018*, pages 1875–1885.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of EMNLP 2017*, pages 2021–2031.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of AAAI 2020*, pages 8018–8025.

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.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of ACL 2017*, pages 1601–1611.

Katharina Kann, Sascha Rothe, and Katja Filippova. 2018. Sentence-level fluency evaluation: References help, but can be spared! In *Proceedings of the CCNLL 2018*, pages 313–323.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Yibin Lei, Yu Cao, Dianqi Li, Tianyi Zhou, Meng Fang, and Mykola Pechenizkiy. 2022. Phrase-level textual adversarial attack with label preservation. In *Findings of NAACL-HLT 2022*.

Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and William B Dolan. 2021. Contextualized perturbation for textual adversarial attack. In *Proceedings of NAACL-HLT 2021*, pages 5053–5069.

J Li, S Ji, T Du, B Li, and T Wang. 2019. Textbugger: Generating adversarial text against real-world applications. In *Proceedings of 26th Annual Network and Distributed System Security Symposium*.

Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2018. Deep text classification can be fooled. In *Proceedings of IJCAI 2018*, pages 4208–4215.

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. *arXiv preprint arXiv:1907.11692*.

Nikola Mrkšic, Diarmuid OSéaghdha, Blaise Thomson, Milica Gašic, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. In *Proceedings of NAACL-HLT 2016*, pages 142–148.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. 2017. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP 2014*, pages 1532–1543.

Danish Pruthi, Bhuwan Dhingra, and Zachary C Lipton. 2019. Combating adversarial misspellings with robust word recognition. In *Proceedings of ACL 2019*, pages 5582–5591.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *Proceedings of ACL 2018*, pages 784–789.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of EMNLP 2016*, pages 2383–2392.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of ACL 2019*, pages 1085–1097.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. *Proceedings of ICLR 2017*.

Koustuv Sinha, Prasanna Parthasarathi, Joelle Pineau, and Adina Williams. 2021. Unnatural language inference. In *Proceedings of NAACL 2021*, pages 7329–7346.

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

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019a. Universal adversarial triggers for attacking and analyzing nlp. In *Proceedings of EMNLP-IJCNLP 2019*.

Eric Wallace, Pedro Rodriguez, Shi Feng, Ikuya Yamada, and Jordan Boyd-Graber. 2019b. Trick me if you can: Human-in-the-loop generation of adversarial examples for question answering. *Transactions of the Association for Computational Linguistics*, 7:387–401.

Boxin Wang, Hengzhi Pei, Boyuan Pan, Qian Chen, Shuohang Wang, and Bo Li. 2020. T3: Tree-autoencoder regularized adversarial text generation for targeted attack. In *Proceedings of EMNLP 2020*, pages 6134–6150.

Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of ACL 2017*, pages 189–198.

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 EMNLP 2018*, pages 2369–2380.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. In *Proceedings of ICLR 2018*.

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2020. Word-level textual adversarial attacking as combinatorial optimization. In *Proceedings of ACL 2020*, pages 6066–6080.

Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2018. Generating natural adversarial examples. In *Proceedings of ICLR 2018*.

A Implementation Details

A.1 Training Victim Models

BERT We use the huggingface-transformers[3] to implement the model and the *bert-base-uncased* version of BERT model[4] to initialize the model weights. It contains 12 layers with a hidden size of 768. A linear layer is added to predict the start and end positions of the answer span.

During fine-tuning BERT on different QA datasets, we set the maximum input sequence length as 384, using an Adam optimizer whose initial learning rate is 6.25e$-$5 with the batch size 32. The epoch number is 3 and the final model after all epochs will be saved as the victim model.

SpanBERT We also use the huggingfance-transformers to implement the model, along with *spanbert-large-cased* version[5] to initialize the weights. It contains 24 layers with a hidden size of 1024. A linear layer is added to predict the start and end positions of the answer span.

During finetuning, we set the maximum input sequence length as 512, using an Adam optimizer whose initial learning is rate 2e$-$5 with the batch size 32. The epoch number is 3 and the final model after all epochs will be saved as the victim model.

BiDAF We use the model implementation provided by AllenNLP[6]. The 6B 100d version of GLoVe (Pennington et al., 2014) is used to initialize the token embedding layer of BiDAF.

During training, we set the maximum input context length as 800, using an Adam optimizer with an initial learning rate 1e-3 and batch size 40 to train BiDAF for 20 epochs. All other settings are in default. We will save the model with the best performance on the dev set as the victim model.

A.2 TASA

Remove coreferences. We use NeuralCoref[7] combined with SpaCy[8] to find out the coreferences in contexts.

Perturbation on answer sentences. To select the answer sentence s_a, we use the answer span position given by each label in datasets, where the sentence containing this span is regarded as s_a. If the answer spans are not unique, we use the answer span that is chosen most times by annotators or the first span in the context. The lemmas and POS tags of different are obtained via SpaCy. The POS tag set used to get keywords is {*"VERB", "NOUN", "ADJ", "ADV"*}. When perturbing a token with its synonyms, we use pyinflect[9] to recover the lemmas of replacements into the same inflections of the original token.

Adding distracting answer sentences. We construct a NER dictionary and a word dictionary (except named entities) for each target dataset by parsing all contexts in both the train and dev sets via SpaCy. During generating DAS or changing answers in DAS, we randomly sample named entities with the same NER tag or words with the same POS tag from the dictionaries we built before. Each time, we sample $N = 20$ from them and ensure there is no overlap with the original entity/token we want to replace. Pyinflect is also used during replacement.

Beam Search. During beam search, we apply an early-stop strategy on the filtered results after each time of a search. We also restrict the maximum perturbation number to 5 for both PAS and DAS. If one of the following 3 criteria is satisfied: 1) the minimum effect score E_n among them satisfies $min(E_n) \geq T_E$, where T_E is a threshold and $T_E = 0.2$; 2) all possible token/entities have been replaced, the beam search will stop, and the final M sentences will proceed to the next step.

Quality filtering. During filtering, we use the official USE model[10] to get USE similarity and a small size GPT2 model[11] to get the PPL.

Model to determine whether a question is answerable for modified context. We use a RoBerta$_{\text{base}}$ model fine-tuned on the original SQuAD 2.0 dataset[12] as the answerable judgment model for SQuAD 1.1, because these two datasets share the same corpus and model trained on SQuAD 2.0 has the capability to predict whether a question is answerable. If the model outputs the highest answer possibility on the special "<s>" token at the beginning of the input, then the current sample is regarded as unanswerable.

For the rest four datasets, we use other Roberta models fine-tuned on our newly constructed training sets. More specially, each set includes the original samples whose label is answerable and

[3] https://github.com/huggingface/transformers

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

[5] https://huggingface.co/SpanBERT/spanbert-large-cased

[6] https://github.com/allenai/allennlp

[7] https://github.com/huggingface/neuralcoref

[8] https://spacy.io

[9] https://spacy.io/universe/project/pyinflect/

[10] https://tfhub.dev/google/universal-sentence-encoder/4

[11] https://huggingface.co/gpt2

[12] https://huggingface.co/deepset/roberta-base-squad2

Datasets	Has EM	Has F1	No EM	EM	F1
SQuAD 2.0*	77.94	84.03	81.80	79.87	82.91
NewsQA	52.68	65.64	98.43	75.56	82.04
NQ	67.72	79.26	99.21	83.46	89.24
HotpotQA	59.43	77.31	99.83	79.63	88.57
TriviaQA	48.45	53.10	99.88	74.17	76.49

Table 6: The performance of our RoBerta models to determine whether a question is answerable on our newly constructed dataset containing answerable and unanswerable samples. “Has” means samples has an answer, “No” means samples without an answer. *: we directly use the SQuAD 2,0 dataset to train the model for SQuAD 1.1 as they share the same corpus.

Hyperparameters	Value
effect score threshold T_E	0.2
quality score threshold T_U	-2
beam search size M	5
random sampling size for DAS N	20

Table 7: Values of hyperparameters used in TASA.

negative samples (unanswerable samples) whose quantity is the same as answerable samples. Here, each negative sample has a question obtained by randomly sampling from the whole dataset that does not belong to the given context. We follow the same training pattern as SQuAD 2.0 to finetune RoBerta models, where the model needs to have the capability of both answering answerable samples and output *"unanswerable"* label for unanswerable samples. We list the performance of all these models used in our experiments in Table 6. Our constructed data are less challenging for models because the questions of negative samples are randomly sampled from the whole corpus, which may be quite different from the context and easy to be distinguished.

We list all hyperparameter values used by TASA method in Table 7, which are obtained by empirical tuning based on the trade-off between attack effectiveness and textual quality. We conduct all our experiments on a single NVIDIA V100 GPU. We also publish our code anonymously at `https://anonymous.4open.science/r/TASA/`.

The possible limitations of our method: TASA is only appliable to extractive QA tasks, and the question or answer is not perturbed to achieve a better deception on models ,which we leave for the future work.

A.3 Baselines

We run the official code provided by the authors of original baseline papers to derive the relevant results in our experiments. We have tried our best to reproduce the results reported in papers, but their configurations are quite different from ours.

Datasets	$\vert Q\vert$	$\vert C\vert$	Train size	Dev size
SQuAD 1.1	11	137	87,599	10,570
NewsQA	8	599	74,160	4,212
Natural Questions (NQ)	9	153	104,071	12,836
HotpotQA	22	232	72,928	5,901
TriviaQA	16	784	61,688	7,785

Table 8: The statistics of 5 datasets used in our experiments. |C| is the average length of context, |Q| is the average length of questions, both in token level.

TextFooler Since this method is not designed for QA tasks, we made some modifications to it. 1) We only use the context as the targeted attack text and mask tokens within it to get their importance scores; 2) in order to avoid changing the answer, we do not involve answer tokens as the perturbation targets; 3) we also use the prediction possibility on the gold answer to get the evaluation on each time attack and determine when to stop the attack. We implement our attack based on the official code and keep other settings as the default.

T3 We apply its official code directly as it already contains the function to attack QA dataset in SQuAD format. To make a fair comparison, we use its **black-box** configuration without accessing the internal parameters of models. Besides, we use its **target** configuration, which aims to specially misguide the model predictions to the pseudo answer in the distracting sentence and shows a better performance.

A.4 Datasets

We provide some statistics about 5 datasets we used in Table 8. We use the official release version of SQuAD 1.1, while the MRQA version [13] for other 4 datasets, where we transform them into the same format as SQuAD 1.1 for the convenience of our experiments.

B Additional Results

B.1 Using BiDAF as the Victim Model

We also include BiDAF as one kind of victim model in our experiments, as it is a representative End2end RNN-based model. The related results are not provided in the main part due to the page limitation and its current fair performance compared to SOTA models. The attack results on five dataset same as §4 are shown in Table 9. Similarly,

[13] https://github.com/mrqa/MRQA-Shared-Task-2019

Dataset	method	EM↓	F1↓	GErr↓	PPL↓	Num
SQuAD 1.1	Original	65.72	75.97	2.39	33.25	10,570
	AddSent*	40.87	49.19	2.47	33.98	3,560
	TextFooler	42.65	56.96	**2.56**	**37.95**	7,228
	T3	52.74	61.69	4.44	44.20	9,681
	OURS	**37.96**	**49.44**	2.89	41.08	9,606
News QA	Original	43.99	57.64	1.98	22.50	4,212
	TextFooler	**32.03**	**46.69**	**2.11**	23.92	3,662
	T3	39.21	51.89	2.56	22.99	3,775
	OURS	33.76	47.23	2.19	**22.83**	2,903
NQ	Original	56.77	68.83	20.48	49.74	12,836
	TextFooler	39.65	53.91	**20.50**	47.31	7,111
	T3	41.98	52.27	20.72	65.61	10,460
	OURS	**37.86**	**49.56**	20.58	**43.25**	8,955
Hotpot QA	Original	46.38	63.88	3.73	17.01	5,901
	TextFooler	36.75	55.40	**3.87**	18.49	4,974
	T3	41.38	58.41	5.16	20.78	5,186
	OURS	**34.12**	**49.15**	4.00	**15.31**	5,050
Trivia QA	Original	45.19	52.85	3.74	24.42	7,785
	TextFooler	**38.05**	**45.25**	**3.82**	25.63	7,227
	T3	43.22	50.07	4.20	25.50	7,434
	OURS	40.21	46.47	3.87	**24.72**	7,110

Table 9: Attack results on 5 QA datasets using BiDAF as the victim model. The best results are **bold**. **Num** is the sample number of a dataset or generated from the whole dataset by a method. ↓ represents that the lower the better. *: annotated by humans.

our TASA achieves the best attack effectiveness in 3 datasets among 5 according to the declining scale of EM and F1, while remaining comparable in the other 2 datasets. In addition, TASA also achieves the overall lower PPL among all conditions and a close performance to TextFooler in terms of grammar error. These observations again demonstrate the superiority of our method. Moreover, it is noticeable that BiDAF is less vulnerable than BERT as the performance degeneration is slighter, especially on datasets with long contexts, e.g., NewsQA and TriviaQA.

B.2 Shift to Pseudo Answers

Since PASs aim to attract models' focus from original answer sentences and misguide models to make predictions on pseudo answers. We have conducted related experiments in §4.4 to prove their validity. Here, we provide more results about this experiment, including not only the F1 scores between predictions by 3 models on TASA adversarial samples and the pseudo answers contained in the corresponding PASs, but also F1 scores between pseudo answers and the models' predictions on the original samples, making a further comparison to eliminate the possible influence of the existing prediction overlap. Results are given in Table 10. Obviously, all models under all conditions tend to predict answers that have more overlaps with pseudo answers given TASA adversarial samples, proving the misguiding effect of DASs. Besides, the F1 score difference between predictions on TASA samples and the original samples will be reduced on datasets where the attack capability of TASA is consistently weaker, such as NewsQA and TriviaQA. This proves that the efficiency of DASs drawing models' attention affects the attack performance remarkably when combined with PAS.

Datasets	SQuAD 1.1	NewsQA	NQ	Hotpot	Trivia
BERT	39.19	20.95	36.22	36.15	24.78
BERT(Ori)	15.08	14.40	21.21	20.38	20.30
SpanBERT	33.20	20.92	32.14	38.09	27.71
SpanBERT(Ori)	16.06	16.54	22.49	21.71	23.54
BiDAF	26.34	16.69	29.43	27.80	20.08
BiDAF(Ori)	12.68	12.80	18.51	16.76	16.42

Table 10: F1 score between predicted answers on TASA adversarial samples with DASs and pseudo answers from corresponding DASs, or between predicted answers on the original samples and pseudo answers (Ori), using 3 victim models on 5 datasets.

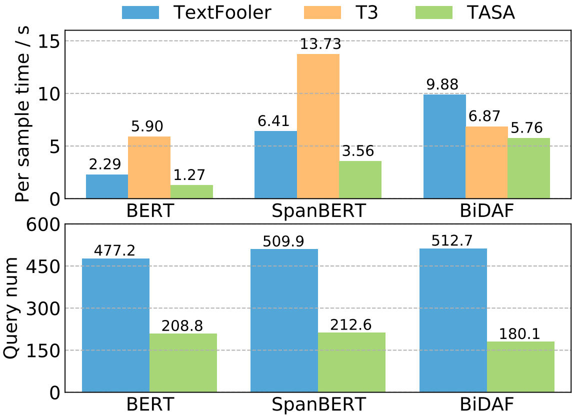


Figure 5: The per sample time to generate adversarial samples (in second) and average query number to victim models of TextFooler, T3 and TASA, using all kinds of victim models on SQuAD 1.1 dataset.

B.3 Analysis of Computational Complexity

We illustrate the per sample attack time and query number to the victim models of our TASA and two baselines, TextFooler and T3, on SQuAD 1.1 dataset and all 3 types of models, in Figure 5. Note that T3 has a constant query number to victim models, so it is not involved in this part. All results are obtained on a single NVIDIA V100 GPU. It can be seen that our TASA is the fastest attack method compared to other baselines, and it also makes fewer queries to the victim model before obtaining an adversarial sample. Although T3 has a

Dataset	**source**	BERT			SpanBERT			BiDAF		
		ratio / %	EM	F1	ratio /‘%	EM	F1	ratio / %	EM	F1
SQuAD 1.1	PAS+DAS	50.2	22.75	32.98	53.2	41.67	53.31	54.4	26.47	36.46
	PAS	8.9	51.06	63.48	10.2	63.90	78.29	9.3	37.46	50.08
	DAS	40.9	58.86	70.03	36.6	70.36	81.12	36.3	55.28	68.69
NewsQA	PAS+DAS	40.8	32.42	45.77	42.7	43.47	59.25	44.0	25.84	39.38
	PAS	19.9	34.23	47.80	20.7	44.67	60.67	21.1	26.49	40.89
	DAS	39.3	50.01	64.52	36.6	62.85	75.29	34.9	40.95	53.78
NQ	PAS+DAS	54.1	33.09	45.27	54.5	44.12	58.50	54.9	30.29	42.25
	PAS	19.5	47.55	62.97	19.9	53.21	69.24	19.9	41.24	55.81
	DAS	26.4	60.80	70.25	25.6	64.26	74.94	25.2	51.72	60.59
HotpotQA	PAS+DAS	61.1	21.60	32.63	63.5	38.55	54.86	60.8	29.91	44.85
	PAS	14.8	33.10	46.91	14.9	51.52	68.81	14.9	34.65	50.73
	DAS	24.1	39.53	52.78	21.6	56.55	72.53	24.3	44.34	59.02
TriviaQA	PAS+DAS	62.9	48.07	51.73	63.3	60.68	66.59	63.5	37.10	43.13
	PAS	13.2	51.86	54.42	13.3	63.16	69.37	12.3	40.61	47.29
	DAS	23.9	60.01	64.72	23.4	71.41	77.31	24.2	48.70	55.65

Table 11: The ratio and performance of QA models on different compositions of adversarial samples generated by TASA, on all 5 datasets and 3 victim models. PAS+DAS: both PAS and DAS are applicable in the current sample; PAS: only PAS is applicable in the current sample; DAS: only DAS is applicable in the current sample.

constant query number to the victim model, its complexity depends on the scale of the target model's embedding.

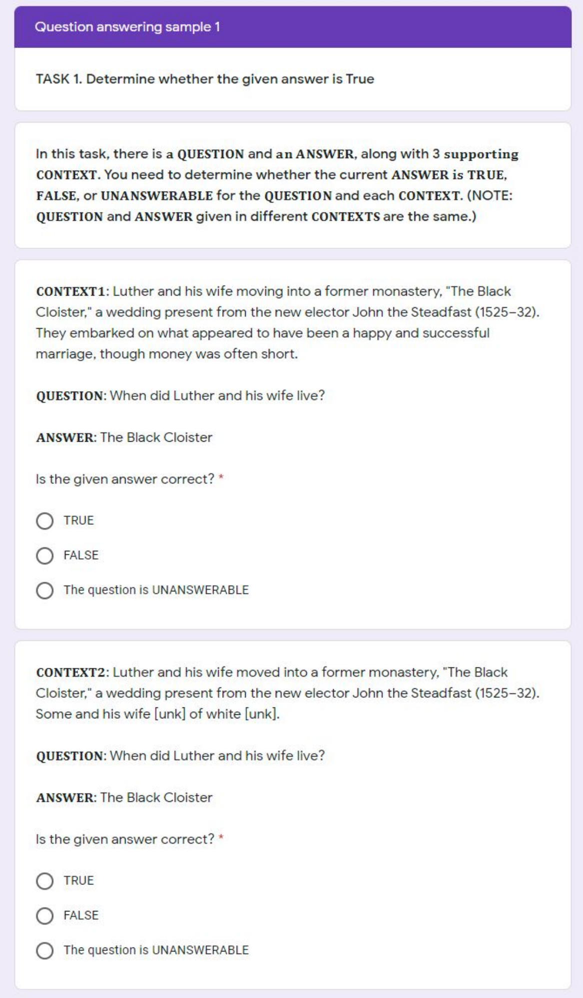
B.4 The Composition of Samples Generated by TASA

Although we design twin sentences, PAS and DAS, to attack QA models, it is possible that not both of them are applicable for a sample. E.g., only PAS is applicable if there is no proper named entity or noun that can be edited in the answer sentence excluding keywords and the gold answer; or only DAS is applicable for a sample where no overlapped keyword is found between the answer sentence and question. A sample where only PAS or DAS is applied will also be put into the final adversarial sample set, along with samples that both PAS and DAS (PAS+DAS) are involved. In order to study the compositions of different adversarial sample sources, as well as the performance of victim models on each part, we provide the ratios of each type of samples generated by TASA on different datasets along with the performance of QA models on them in Table 11.

It can be found that PAS+DAS compose the majority of adversarial samples on nearly all datasets, while the quantities samples that only contain DAS are generally larger than samples with only PAS. When comes to the performance of QA models on each part, it can be found that PAS+DAS has the best attack effectiveness among all types of samples, because they not only deceive models using perturbed keywords but also utilize distracting answer sentences to misguide models to make wrong predictions on the included pseudo answers. On the other hand, only using PAS or DAS can lower the attack effectiveness. The reason is that a single attack source may not sufficiently fool models, proving the necessity of combining the two folds of pitfall we discussed in §2 into the adversarial attack on the QA task. Moreover, the attack difference between PAS+DAS and PAS will be narrowed on datasets having longer contexts like NewsQA and TriviaQA, where EM and F1 values on these two types of samples are more close. The relatively weak attack ability on such datasets should be the main cause. Besides, longer input sequences will lower the attention weights of models on each token, merely adding PAS also results in less influence because their ratio on the whole input becomes smaller.

C Qualitative Samples

We provide some samples generated by TextFooler, T3 and TASA along with corresponding model predictions in Table 12, Table 13. We also provide the instruction screenshot for human evaluation in Figure 6 and Figure 7.



Question answering sample 1

TASK 1. Determine whether the given answer is True

In this task, there is a **QUESTION** and **an ANSWER**, along with 3 **supporting CONTEXT**. You need to determine whether the current **ANSWER** is **TRUE**, **FALSE**, or **UNANSWERABLE** for the **QUESTION** and each **CONTEXT**. (NOTE: **QUESTION** and **ANSWER** given in different **CONTEXTS** are the same.)

CONTEXT1: Luther and his wife moving into a former monastery, "The Black Cloister," a wedding present from the new elector John the Steadfast (1525–32). They embarked on what appeared to have been a happy and successful marriage, though money was often short.

QUESTION: When did Luther and his wife live?

ANSWER: The Black Cloister

Is the given answer correct? *

○ TRUE

○ FALSE

○ The question is UNANSWERABLE

CONTEXT2: Luther and his wife moved into a former monastery, "The Black Cloister," a wedding present from the new elector John the Steadfast (1525–32). Some and his wife [unk] of white [unk].

QUESTION: When did Luther and his wife live?

ANSWER: The Black Cloister

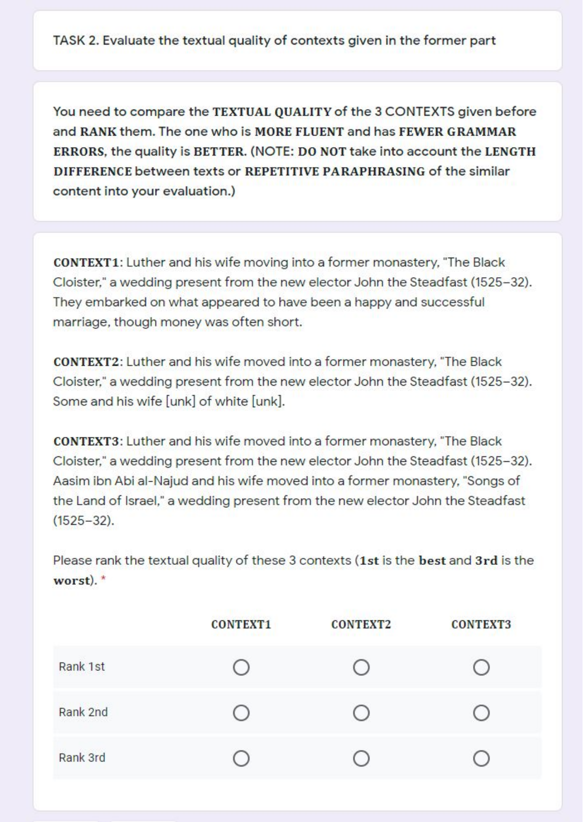
Is the given answer correct? *

○ TRUE

○ FALSE

○ The question is UNANSWERABLE

Figure 6: Screenshot of instructions for human evaluation (part1).



TASK 2. Evaluate the textual quality of contexts given in the former part

You need to compare the **TEXTUAL QUALITY** of the 3 CONTEXTS given before and **RANK** them. The one who is **MORE FLUENT** and has **FEWER GRAMMAR ERRORS**, the quality is **BETTER**. (NOTE: **DO NOT** take into account the **LENGTH DIFFERENCE** between texts or **REPETITIVE PARAPHRASING** of the similar content into your evaluation.)

CONTEXT1: Luther and his wife moving into a former monastery, "The Black Cloister," a wedding present from the new elector John the Steadfast (1525–32). They embarked on what appeared to have been a happy and successful marriage, though money was often short.

CONTEXT2: Luther and his wife moved into a former monastery, "The Black Cloister," a wedding present from the new elector John the Steadfast (1525–32). Some and his wife [unk] of white [unk].

CONTEXT3: Luther and his wife moved into a former monastery, "The Black Cloister," a wedding present from the new elector John the Steadfast (1525–32). Aasim ibn Abi al-Najud and his wife moved into a former monastery, "Songs of the Land of Israel," a wedding present from the new elector John the Steadfast (1525–32).

Please rank the textual quality of these 3 contexts (**1st** is the **best** and **3rd** is the **worst**). *

	CONTEXT1	CONTEXT2	CONTEXT3
Rank 1st	○	○	○
Rank 2nd	○	○	○
Rank 3rd	○	○	○

Figure 7: Screenshot of instructions for human evaluation (part2).

Original context	Long-term active memory is acquired following infection by activation of B and T cells. Active immunity can also be generated artificially, through vaccination. The principle behind vaccination (also called immunization) is to introduce an antigen from a pathogen in order to stimulate the immune system and develop specific immunity against that particular pathogen without causing disease associated with that organism. This deliberate induction of an immune response is successful because it exploits the natural specificity of the immune system, as well as its inducibility. With infectious disease remaining one of the leading causes of death in the human population, vaccination represents the most effective manipulation of the immune system mankind has developed.
Question	By what process can active immunity be generated in an artificial manner?
Answer	vaccination
TextFooler context	Long-term active memory is obtaining following infection by activation of B and T cells. Active immunity can also constitute generated mannually, through vaccination. The principle behind vaccination (also called immunization) is to introduce an antigen from a pathogen in order to stimulate the immune system and develop specific immunity against that particular pathogen without causing disease associated with that organism. This deliberate induction of an immune response is successful because it exploits the natural specificity of the immune system, as well as its inducibility. With infectious disease remaining one of the leading causes of death in the human population, vaccination represents the most effective manipulation of the immune system mankind has developed.
Model prediction	vaccination
T3 context	Long-term active memory is acquired following infection by activation of B and T cells. Active immunity can also be generated artificially, through vaccination. The principle behind vaccination (also called immunization) is to introduce an antigen from a pathogen in order to stimulate the immune system and develop specific immunity against that particular pathogen without causing disease associated with that organism. This deliberate induction of an immune response is successful because it exploits the natural specificity of the immune system, as well as its inducibility. With infectious disease remaining one of the leading causes of death in the human population, vaccination represents the most effective manipulation of the immune system mankind has developed. Active immunity generated immunization.
Model prediction	vaccination
TASA context	Long-term active memory is acquired following infection by activation of B and T cells. Alive immunity can also be produced artificially, through vaccination. The principle behind immunization (also called immunization) is to introduce an antigen from a pathogen in rank to stimulate the immune system and arise precise resistance against that particular pathogen without causing disease associated with that organism. Thpersonify deliberate induction of an immune response personify successful because it utilises the natural specificity of the immune system of rule, as well as its inducibility. With infectious disease remaining one of the leading causes of death in the human population, vaccination represents the most effective manipulation of the immune system mankind has developed. Active irradiation can also be generated artificially, through sword - cut.
Model prediction	sword - cut
Original context	In 1873, Tesla returned to his birthtown, Smiljan. Shortly after he arrived, Tesla contracted cholera; he was bedridden for nine months and was near death multiple times. Tesla's father, in a moment of despair, promised to send him to the best engineering school if he recovered from the illness (his father had originally wanted him to enter the priesthood).
Question	What did Tesla's father originally want him to do?
Answer	enter the priesthood
TextFooler context	In 1873, Tesla returns to his birthtown, Smiljan. Shortly after he arrived, Tesla contracted cholera; he was crippled for nine months and was near death multiple times. Tesla's dads, in a tiempo of angst, pledging to transmits him to the advisable engineers schooling if he recaptured from the malady (his father had originally wanted him to enter the priesthood).
Model prediction	enter the priesthood
T3 context	In 1873, Tesla returned to his birthtown, Smiljan. Shortly after he arrived, Tesla contracted cholera; he was bedridden for nine months and was near death multiple times. Tesla's father, in a moment of despair, promised to send him to the best engineering school if he recovered from the illness (his father had originally wanted him to enter the priesthood). Our our father our want father to us entering of ordained.
Model prediction	enter the priesthood
TASA context	In 1873, Tesla delivered to his birthtown, Smiljan. Shortly after he arrived, Tesla contracted Asiatic cholera; he was bedridden for nine months and was near death multiple times. Tesla's dad, in a moment of despair, promised to send him to the best engineering school if he recovered from the illness (his dad had in the beginning required him to enter the priesthood). The Bureau of Near Eastern Affairs's father, in a moment of despair, promised to send him to the best engineering school if he recovered from the illness (his father had originally wanted him to sadden the businessman).
Model prediction	sadden the businessman

Table 12: Adversarial contexts generated by TextFooler, T3, and TASA, compared to the original context on SQuAD 1.1 using BERT as victim model, along with predicted answers by the model. Gold answer, perturbed tokens (i.e. perturbations on answer sentence for TASA), added distracting sentences (i.e. DAS for TASA), and wrong answers are in different colors. Underlined sentences indicate the answer sentences.

Original context	The Daily Mail newspaper reported in 2012 that the UK government's benefits agency was checking claimants' "Sky TV bills to establish if a woman in receipt of benefits as a single mother is wrongly claiming to be living alone" – as, it claimed, subscription to sports channels would betray a man's presence in the household. In December, the UK's parliament heard a claim that a subscription to BSkyB was 'often damaging' , along with alcohol, tobacco and gambling. Conservative MP Alec Shelbrooke was proposing the payments of benefits and tax credits on a "Welfare Cash Card", in the style of the Supplemental Nutrition Assistance Program, that could be used to buy only "essentials".
Question	what did the UK parliment hear that a subscription to BSkyB was?
Answer	often damaging
TextFooler context	The Daily Mail newspapers reported in 2012 that the UK government's benefits agency was checking claimants' "Sky TV bills to establish if a woman in receipt of benefits as a unaccompanied mamma is disproportionately arguing to are residing alone" –, it asserted, syndication to sporting pipelines would betraying a husband's betrothal in the habitation. In December, the UK's assemblage heard a requisitions that a subscriber to BSkyB was 'often damaging', along with liquor, tobacco and gambling. Conservative MP Alec Shelbrooke was proposing the repaying of benefits and tax credits on a "Welfare Cash Card", in the styling of the Supplemental Nutrition Assistance Program, that could be used to buy only "essentials".
Model prediction	damaging
T3 context	The Daily Mail newspaper reported in 2012 that the UK government's benefits agency was checking claimants' "Sky TV bills to establish if a woman in receipt of benefits as a single mother is wrongly claiming to be living alone" – as, it claimed, subscription to sports channels would betray a man's presence in the household. In December, the UK's parliament heard a claim that a subscription to BSkyB was 'often damaging', along with alcohol, tobacco and gambling. Conservative MP Alec Shelbrooke was proposing the payments of benefits and tax credits on a "Welfare Cash Card", in the style of the Supplemental Nutrition Assistance Program, that could be used to buy only "essentials". The world it contained to the available than [unk] available sometimes damaged.
Model prediction	often damaging
TASA context	The Daily Mail newspaper reported in 2012 that the UK government's profits agency was checking claimants' "Sky tv set throwaways to establish if a woman in receipt of profits as a single mother is wrongly claiming to be living alone" – as, it claimed, subscription to gambols epithelial ducts would betray a man's presence in the household. In December, the UK's parliament noticed a claim that a subscription to BSkyB was 'often damaging', along with alcohol, tobacco and gambling. Conservative MP Alec Shelbrooke was popping the questioning the requitals of dos goods and tax credits on a "Welfare Cash Card", in the style of the Supplemental Nutrition Assistance Program, that could be used to buy only "essentials". In December, the Bhinmal's parliament heard a claim that a subscription to BSkyB was 'meticulously ionateing', along with alcohol, tobacco and gambling.
Model prediction	meticulously ionateing
Original context	On May 21, 2013, NFL owners at their spring meetings in Boston voted and awarded the game to Levi's Stadium. The $1.2 billion stadium opened in 2014. It is the first Super Bowl held in the San Francisco Bay Area since Super Bowl XIX in 1985, and the first in California since Super Bowl XXXVII took place in San Diego in 2003.
Question	When did Levi's stadium open to the public?
Answer	2014
TextFooler context	On May 21, 2013, NFL owners at their spring meetings in Boston voted and awarded the game to Levi's Stadium. The $1.2 trillion stadium opened in 2014. It is the first Super Bowl held in the San Francisco Bay Area since Super Bowl XIX in 1985, and the first in California since Super Bowl XXXVII took place in San Diego in 2003.
Model prediction	May 21, 2013
T3 context	On May 21, 2013, NFL owners at their spring meetings in Boston voted and awarded the game to Levi's Stadium. The $1.2 billion stadium opened in 2014. It is the first Super Bowl held in the San Francisco Bay Area since Super Bowl XIX in 1985, and the first in California since Super Bowl XXXVII took place in San Diego in 2003. By by got to to these and 2012.
Model prediction	2014
TASA context	On May 21, 2013, NFL possessors at their spring runs across in Boston balloted and awarded the game to Levi's Stadium. The $1.2 billion stadium opened in 2014. It is the first Super Bowl held in the San Francisco Bay Area since Super Bowl XIX in 1985, and the first in California since Super Bowl XXXVII took place in San Diego in 2003. The $1.2 billion door opened in 2 June 2013.
Model prediction	May 21, 2013

Table 13: Adversarial contexts generated by TextFooler, T3, and TASA, compared to the original context on SQuAD 1.1 using BERT as victim model, along with predicted answers by the model. Gold answer, perturbed tokens (i.e. perturbations on answer sentence for TASA), added distracting sentences (i.e. DAS for TASA), and wrong answers are in different colors. Underlined sentences indicate the answer sentences.