TASA: Twin Answer Sentences Attack for Adversarial Context Generation in Question Answering

Anonymous ACL submission

Abstract

We present **T**win **A**nswer **S**entences **A**ttack (TASA), a novel question answering (QA) adversarial attack method that produces fluent and grammatical adversarial contexts while maintaining its gold answers. Despite phenomenal progresses on general adversarial attacks, few works have investigated the vulnerability and adversarial attack specifically for QA. In this work, we first investigate the biases in the existing models and discover that they heavily rely on keyword matching and ignore the relevant entities from the question. TASA explores the two biases above and 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 is able to attack the target model efficiently while sustaining the quality of contexts. Extensive experiments on four QA datasets and human evaluations demonstrate that TASA generates substantial-high-quality attacks than existing textual adversarial attack methods.

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 conducts reasoning on the context and predict the answer (Rajpurkar et al., 2016). Most works keep improving the answer correctness on benchmarks (Seo et al., 2017; Devlin et al., 2019), while few studies investigate the robustness of QA models, e.g., is the performance achieved by sound reasoning or via shortcuts? 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 special properties of QA. The vulnerability and biases of QA models can lead to catastrophic failures outside the benchmark datasets. And an effective way to study them is through adversarial attacks specifically designed for QA tasks.

Generating adversarial textual examples is a challenging task 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-loop (Jia and Liang, 2017; Wallace et al., 2019b) 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) study to generate adversarial QA examples automatically. But they only perturb either the context or the question separately and thus break their consistency. Moreover, the major pitfalls of QA models' detailed reasoning process are not fully investigated, leading to the difficulty of producing powerful adversarial attacks.

In this paper, we develop an adversarial attack specifically targeting two biases of mainstream QA models: (1) 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 blackbox 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 replace the keywords with their synonyms; while the other leaves the keywords and the syntactic structure intact but changes the entities (subjects or objects) associated with the answer. Thereby, the former is a *perturbed answer sentence* (PAS)

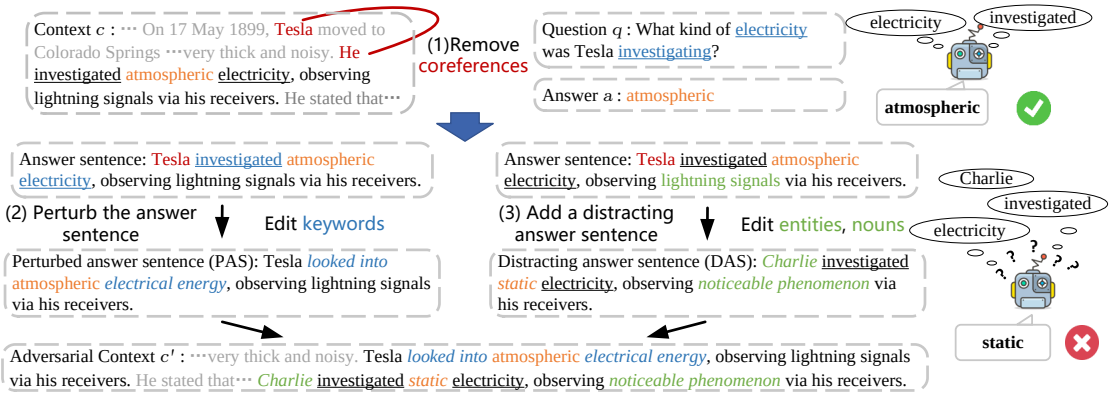

Figure 1: An example of TASA generating adversarial context C'. Underlined parts indicate keywords. Orange indicates gold answer or *pseudo answer*. Other colors indicate tokens for perturbation, distracting, or coreferences.

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. Therefore, the resulted adversarial context can substantially distort the QA model reasoning without changing the answer for humans. To address the challenge of efficiency and textual fluency, we further propose specific beam search and filtering techniques empowered by pre-trained language models.

In experiments, we evaluate TASA and other textual adversarial attack methods on attacking two popular contextulized QA models, BERT (Devlin et al., 2019) and BiDAF (Seo et al., 2017), on four extractive QA datasets, i.e., SQuAD 1.1, SQuAD 2.0, NewsQA, and NaturalQuestions (Rajpurkar et al., 2016, 2018; Trischler et al., 2017; Kwiatkowski et al., 2019). Experimental results and human evaluations consistently show that TASA achieves higher attack efficiency and success rate than other baselines and meanwhile preserves the textual quality and gold answers identifiable by humans. We further analyze the effectiveness of each component in TASA via ablation studies. Our contributions are three-fold:

- We propose a novel adversarial attack method "TASA" specifically designed to fool extractive QA models but retain the gold answers for humans.
- We study the biases and vulnerability of QA models that motivate TASA and demonstrate that those models heavily rely on keywords matching while ignoring their contextual relation to critical entities.
- Experiments on four QA benchmark datasets and two types of victim models demonstrate that TASA significantly outperforms existing textual attack baselines on attack performance, as well as its capability to preserve textual quality and answers.

2 Reasoning bias in Question Answering

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

We analyze two mainstream QA models with contextualized reasoning capabilities, BERT (Devlin et al., 2019) and BiDAF (Seo et al., 2017) (more details in §4), on the samples from SQuAD1.1 (Rajpurkar et al., 2016). Both models are trained on the original SQuAD1.1 and then evaluated on different samples modified from the validation set. We define the sentence in the context that contains the gold answer as the *answer sentence*. Since it is the key for answer prediction (Chen and Durrett, 2019), we first compare the results of QA models on the original sample (Context + Question) with the results on (marked as "Answer sent. + Question") To investigate the bias on sensitive words, we further examine QA models on samples with various types of sensitive words in the answer sentence (1) either removed (marked as "*Remove*") or (2) only retained (marked as "*Only*"). Three types of sensitive words are considered:

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

***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: Remove or only retain (*Only*) different types of sensitive words, the answer is underlined and kept.

Model	BERT		BiDAF	
	EM	F1	EM	F1
Context + Question	80.91	88.23	65.72	75.97
Answer sent. + Question	+2.79	+2.87	+3.27	+4.37
Remove entities	-0.51	-0.89	+0.81	+0.02
Only entities	-21.71	-14.29	-26.77	-16.74
Remove lexical.	-16.77	-15.80	-17.70	-17.46
Only lexical.	-4.65	-1.16	+0.46	+3.50
Remove func.	-2.65	-1.42	-3.34	-2.26
Only func.	-22.20	-20.26	-18.92	-18.94

Table 1: EM and F1 scores of BERT and BiDAF models on different modified samples compared to results on the original samples (Context + Question).

When modifying the answer sentence, we only remove or retain these three types of sensitive words except the gold answer words. And we keep the rest context intact. As shown in Figure 2, the modified sentences are unreadable and difficult to infer its true meaning from human perspective.

Table 1 compares the evaluation metrics of QA models on different types of modifications. Given the answer-sentence-only context, the EM and F1 of both both BERT and BiDAF are improved, indicating that they mainly rely on the answer sentence and almost ignore the rest of context. Moreover, while removing entities or function words cause little difference on the metrics, removing lexical words leads to 20% $\sim$ 30% performance drop. In addition, both models perform surprisingly satisfactory when keeping only lexical words in answer sentences, comparing to the 30% $\sim$ 60% drop when keeping only the entities or function words. These results suggest that both models heavily rely on the token-level (not contextual) information of lexical words from the question, i.e., *keywords* in the context.

The above observations implies a pitfall of QA models that we can leverage to design an efficient adversarial attack specifically for QA. Can we lower the model's attention on the gold answer and then misguide it to incorrect answers by manipulating sensitive keywords in the context? The answer is affirmative: we show that the model predictions can be shifted to crafted wrong answers in §4.3?

3 Methodology

We propose an adversarial attack method for QA, **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, we study how to perturb the context c as c' to form an adversarial example (c', q, a) that can fool $F(\cdot)$ towards producing an incorrect answer $F(c', q) \neq a$, while c' retains the correct answer a that can be identified by humans. We **only modify the context** (conditioned on q and a) and keep all tokens in the question q intact in order to make sure that A is still a valid answer of q, because editing q can easily change its meaning and the gold answer due to its simple syntactic structure. Therefore, we left the adversarial perturbation of q to the future work.

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 lexical words discussed 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 entities (subjects or objects) to misguide the model for producing a wrong answer. Figure 1 illustrates how the three steps are applied. 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 word-level or sentence-level adversarial attacks. 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 replacing "*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 complicated coreference.

3.2 Perturb the Answer Sentence

According to former analysis, the *answer sentence* is the most important part of context c for QA tasks and QA models mainly focus on it by keyword matching. Hence, we first study how to obtain a perturbed answer sentence (PAS) by only perturbing those *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 mainly rely on *keywords* to generate predictions. Hence, to produce more effective attacks, we directly perturb those keywords rather than random tokens in previous works (Ren et al., 2019). We adopt three criteria to select tokens of s_a into the keyword set $\mathcal{X}$: (1) they are not included in the answer span a so perturbations do not change the answer; and (2) each keyword shares the same lemma with a token in the question q; and (3) each keyword's POS tag is included in a POS tag set $\mathcal{K}$ for lexical words, e.g., *NOUN*, *ADJ*.

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

$$I_i = F_a(c, q) - F_a(mask(c, x_i), q), \qquad (1)$$

where $F_a(\cdot, \cdot)$ denotes the probability of the gold answer a predicted by the victim model $F(\cdot, \cdot)$, $mask(c, x_i)$ is c modified by replacing a token x_i with a special mask symbol, e.g., for $c = ..x_{i-1}, x_i, x_{i+1}..$, $mask(c, x_i) = ..x_{i-1}, < mask >, x_{i+1}...$ Finally, we obtain a ranked set $\mathcal{X}'$ of keywords .

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 $\mathcal{R}$ by transforming the inflection of r_j as 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.

The synonym set $\mathcal{R}$ is obtained by unionizing two sources, i.e., (1) **WordNet** synonym dictionary (Fellbaum, 2010) and (2) **PPDB 2.0** dataset (Pavlick et al., 2015). Since the later is a paraphrase dataset, we use token-level paraphrase pairs as synonyms (Mrkšic et al., 2016) for x_i. Thereby, multiple PASs can be generated when editing each keyword if the size of $\mathcal{R}$ is more than one. We only retain top few of them by a beam search and filtering strategy (as elaborated in §3.4) to attack the target model efficiently.

3.3 Add a Distracting Answer Sentence

PAS replaces the *keywords* with their synonyms. While it does not change the actual meaning, it will distract the model, which mainly relies on keyword matching, away from PAS containing the gold answer. In the following, we further add a distracting answer sentence (DAS) at the end of the context in a similar manner as previous works (Jia and Liang, 2017; Wallace et al., 2019a). Collaborating with PAS generated above, DAS additionally misguides models to an incorrect answer due to the keywords matching pitfall studied in §2. In particular, DAS is modified from the *answer sentence* s_a as well: it changes the subjects/objects but keeps the keywords intact which can lead to the answer. Hence, models relying on keyword matching will focus on DAS and produce incorrect answers regarding wrong subjects/objects.

Determine the tokens to edit. Similar to GAS, 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 $y \in \mathcal{Y}$ need to meet all the following criteria: (1) $y \in s_a$; (2) $y \notin \mathcal{X}$ so the keywords are preserved; (3) $y \notin a$ (as we will process the answer tokens separately); (4) y is a named entity or its POS tag is NOUN. The goal of (4) is to extract and change the subjects/objects 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 to be empirically less accurate than POS tag.

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 as the candidates from the whole corpus to replace y_i; (2) otherwise, we randomly sample N nouns with the same hypernym as y_i from the corpus. Hence, multiple DASs can be generated and we use the beam search strategy to only choose top few of them.

Change the answer in DAS. Since the main purpose of DAS is to misguide the model to a wrong answer, we replace the text of the original answer in DAS with a pseudo answer a'. Entirely removing the original answer from DAS also helps to remove ambiguity of the answer for humans. Specifically, we replace every lexical token of a in DAS with one of false 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 a' and thus a beam search based filtering is necessary for efficiency purpose.

3.4 Beam Search and Filtering

Beam search. When editing each word in generating 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.

$$E_n = F_a(c, q) - F_a(edit(c, s_n), q), \quad (2)$$

where $edit$ denotes the context c modified by s_n: (1) if s_n is a PAS, it replaces the original s_a in c; (2) if s_n is a DAS, it is added to the end of c. These candidate sentences will be ranked by E_n in the descending order and only the top M (beam size) are retained for the next edit step. Beam search stops if (1) no more token needs to be modified or (2) the minimum effect score after beam search is higher than a threshold T_E that ensures sufficient performance drop. TASA runs beam search for PAS and then beam search for DAS sequentially to generate the adversarial context c'.

Filtering by textual quality. To ensure high textual quality and label preservation of the generated adversarial context, TASA applies a filtering procedure on the M PASs achieved in beam search. In particular, we firstly use a model F_J to justify whether the question q is answerable given the generated context $edit(c, s_n)$, where s_n is a PAS (F_J is a pretrain model fine-tuned on both answerable and unanswerable samples, refer to Appendix A.2 for details). Only those contexts classified as **answerable** will be remained. We then compute the remained contexts' textual quality index 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 beam search.

Algorithm 1 TASA

Input: a QA sample (c, q, a), 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 based on a;
$\mathcal{X}' \leftarrow$ keywords in s_a and rank them by I_i in Eq. 1;
$\mathcal{P} \leftarrow \{s_a\}$ (initialize a set of one item s_a)
for $1 \leq i \leq |\mathcal{X}'|$ **do**
    $\mathcal{U} \leftarrow$ a set of PASs, each perturbs x_i of an item in $\mathcal{P}$;
    $\mathcal{P} \leftarrow M$ items in $\mathcal{U}$ with the highest E_n in Eq. 2;
    if minimum E_n in $\mathcal{P} \geq T_E$ **then break;**
    end if
end for
$\mathcal{P} \leftarrow$ PASs in $\mathcal{P}$ filtered by F_J and U_n in Eq. 3;
$\mathcal{C} \leftarrow$ a set of contexts, each c_j modified by a PAS in $\mathcal{P}$;
$\mathcal{Y} \leftarrow$ a set of tokens in s_a, to be edited for DAS;
$\mathcal{D} \leftarrow$ a set of $\{(s_j, c_j)\}$, each context $c_j \in \mathcal{C}$ is associated with a DAS s_j initialized as s_a;
for $1 \leq i \leq |\mathcal{Y}|$ **do**
    $\mathcal{U} \leftarrow$ a set of DAS, each editing y_i in s_j from $\mathcal{D}$;
    $\mathcal{D} \leftarrow M$ items in $\mathcal{U}$ with the highest E_n in Eq. 2;
end for
Change the answer tokens in s_j of all items in $\mathcal{D}$;
$(s_b, c_b) \leftarrow$ **The** item in $\mathcal{D}$ with the highest E_n in Eq. 2;
$c' \leftarrow$ add DAS s_b to the end of context c_b;
return c';

4 Experiments

Datasets. We evaluate the QA adversarial attacks generated by TASA using 4 extractive QA datasets: SQuAD 1.1 (Rajpurkar et al., 2016), SQuAD 2.0 (Rajpurkar et al., 2018), NewsQA (Trischler et al., 2017), and Natural Questions (NQ) (Kwiatkowski et al., 2019). We use the settings of MRQA (Fisch et al., 2019) for the latter two datasets and remove unanswerable samples in SQuAD 2.0 as they are outside the scope of the adversarial attack type studied in this paper (see Appendix A.4 for their statistics). We report results on their **dev sets**, as not all their test sets are publicly available.

Victim models. We attack two QA models, i.e., BERT (Devlin et al., 2019) and BiDAF (Seo et al., 2017), in experiments. The former one is a fine-tuned QA model on top of pretrained $\text{BERT}_{\text{base}}$ that has benefited from a huge corpus and already has shown its superiority on many NLP applications. The latter is an end2end model based on LSTM and bidirectional attention specially designed for extractive QA. Both of them predict the start and end position of the answer span in context.

Victim model		BERT					BiDAF				
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	65.72	75.97	2.39	33.25	10,570
	AddSent*	57.78	64.58	2.47	33.98	3,560	40.87	49.19	2.47	33.98	3,560
	TextFooler	67.18	78.18	**2.95**	44.84	7,919	42.65	56.96	**2.56**	**37.95**	7,228
	T3	71.63	78.86	3.48	44.45	9,622	52.74	61.69	4.44	44.20	9,681
	OURS	**40.06**	**50.87**	2.98	**41.15**	9,559	**37.96**	**49.44**	2.89	41.08	9,606
SQuAD 2.0	Original	79.06	87.38	2.29	32.27	5,928	67.41	77.74	2.29	32.27	5,928
	TextFooler	64.95	77.04	**2.76**	43.23	4,488	43.91	58.30	**2.43**	**37.55**	4,001
	T3	69.99	78.06	3.60	42.69	5,506	54.17	63.67	4.30	44.20	5,509
	OURS	**42.29**	**54.69**	2.81	**42.23**	5,386	**39.10**	**51.34**	2.69	42.66	5,404
NewsQA	Original	51.57	65.57	1.98	22.50	4,212	43.99	57.64	1.98	22.50	4,212
	TextFooler	43.31	58.34	**2.14**	24.33	3,727	**32.03**	**46.69**	**2.11**	23.92	3,662
	T3	**39.54**	53.49	2.33	**22.86**	3,865	39.21	51.89	2.56	22.99	3,775
	OURS	39.62	**53.46**	2.16	**22.86**	2,860	33.76	47.23	2.19	**22.83**	2,903
NQ	Original	67.39	79.28	20.48	49.74	12,836	56.77	68.83	20.48	49.74	12,836
	TextFooler	48.31	63.08	20.46	49.02	7,158	39.65	53.91	**20.50**	47.31	7,111
	T3	60.06	71.20	20.93	60.90	10,439	41.98	52.27	20.72	65.61	10,460
	OURS	**43.23**	**55.32**	**20.42**	**44.30**	8,809	**37.86**	**49.56**	20.58	**43.25**	8,955

Table 2: Main results on 4 QA datasets. 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 human.

Attack. Given a dataset, we firstly train each victim model on its training set to get a trained model which achieves satisfactory performance on its 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. More details are provided in Appendix A.2.

Baselines. We consider the following 2 strong baselines 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. We use it in a black-box manner and under the targeted config.

Besides, we also include human-annotated **AddSent** (Jia and Liang, 2017) data for SQuAD 1.1 dataset, as they share the same contexts.

Evaluation metrics. Following the previous works (Rajpurkar et al., 2016; Wang et al., 2020; Li et al., 2021), we evaluate our attack methods using the following metrics: 1) **EM**, the exact match ratio of predicted answers; 2) **F1**, the F1 score between the predicted answer and the gold answer. Lower values of them means a better attack success rate; 3) **Grammar error (GErr)**, the grammatical errors number in contexts of all samples given by LanguageTool[1] following (Zang et al., 2020), we use the average per 100 tokens due to different context lengths; 4) **PPL**, the average perplexity of all contexts given by a small sized GPT2 model (Radford et al., 2019) to measure the fluency of texts (Kann et al., 2018). Lower values of later two values indicate a better textual quality.

4.1 Main Results

The main experimental results are summarized in Table 2. TASA achieves the overall best performance among all methods. In particular, it shows the best attack success performance than others on three 3 datasets, and comparable best results on the NewsQA dataset. At the same time, TASA also has a high efficiency of transforming as many as possible original samples into valid adversarial samples as illustrated in **Num**, which is better than TextFooler. In terms of the quality of generated contexts, TASA overall achieves the best performance on **PPL** and comparable best results on **GErr**.

TextFooler usually has the lowest GErr, because it is a pure token-level method that generates fewer sentence-level unnatural errors during attacking. While T3 always results in distracting sentences that are meaningless without a complete syntactic structure, resulting in the highest GErr and PPL. TASA fulfills attacking on both toke and sentence level, avoiding significant textual quality loss.

We also noticed that TASA is even better on at-

[1]https://languagetool.org/

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 preservation in percentage). ± indicates the confidence intervals with a 95% confidence level.

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 GAS	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.

tack success than AddSent, who collects adversarial samples by adding human-annotated distracting sentence. Despite having a better textual quality, AddSent does not consider the keyword matching pitfall of models which limited its effectiveness.

Human evaluation. We randomly sample 150 sets of adversarial samples, each containing 3 samples generated by TextFooler, T3 and TASA originated from the same sample in SQuAD 1.1 using BERT as the victim model. Each set is evaluated by non-expert annotators in two aspects: (1) Answer preservation , whether the gold answer of a sample remains unchanged; (2) Textual quality, ranking the quality of the context based on the fluency and grammaticality. Totally 63 annotators are involved. Results in Table 3 shows that TASA has equivalent label preservation as Textfooler, and both of them are weaker than T3 as it does not change answer sentences so the gold answers are always preserved. The textual quality of TASA is slightly lower than TextFooler as it includes both token-level and sentence-level modifications, while significantly better than purely sentence-level T3. Some adversarial cases are provided in Appendix C.

4.2 Ablation Studies

We verify the effectiveness of each key module by removing it from TASA: (1) *w/o remove coref.*: without removing coreferences; (2) *w/o PAS*: without perturbing answer sentence; (3) *w/o DAS*: without adding distracting answer sentence (DAS). Upper part in Table 4 proves their contributions. Removing coreferences slightly benefits the quantity of suitable samples for attacking. Both PAS and DAS make vital contributions to the final attack efficiency, as well as number of adversarial samples.

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

Ablations on DAS is finally considered, (1) *w/o pseudo answer*: do not change answers in DAS; (2) *Only NE*: only edit named entities; and (3) *Only nouns*: only edit nouns to get DAS. The significant drop on *w/o pseudo answer* illustrates that changing the original answer is crucial for TASA, which also proves DAS can draw models' attention and misguide them. More types of editing candidates, including both NE and nouns, also benefit the attack effectiveness and generated sample quantity.

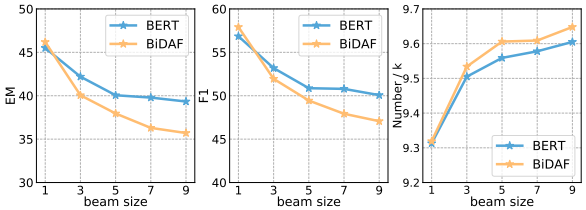

Figure 3: The EM, F1 and quantities of adversarial samples using different beam size on BERT and BiDAF.

4.3 More Analysis

Effect of beam size. We vary beam size to investigate its impact on the overall performance. Figure 3 reports the corresponding EM, F1 and quantities of generated adversarial samples. Clearly, a larger beam size leads to better performance and more diverse adversarial samples. Naturally, the larger the beam size, the slower the speed. We use $M = 5$ for trading off performance and efficiency (limited performance gains from beam sizes larger than 5).

Shift to the pseudo answers. Since DAS aims to misguide the attention from models to them, and we expect 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 have a DAS from 4 datasets using two base models. The results

Datasets	SQuAD 1.1	SQuAD 2.0	NewsQA	NQ
BERT	39.19	33.49	20.95	36.22
BiDAF	26.34	25.37	16.69	29.43

Table 5: F1 score between predicted answers and pseudo answers from different adversarial datasets.

demonstrate that victim models make nearly all wrong predictions on pseudo answers, except samples that cannot get the correct answers even using the original data, confirming the effect of DASs.

Adversarial training. To verify the effectiveness of TASA in improving the robustness of QA models, we randomly replace training samples in SQuAD 1.1 with corresponding adversarial samples generated by TASA in various ratios and then fine-tune a BERT model on the new training data. The performance on the original dev set, the adversarial dev set by TASA, and AddSent data, using models fine-tuned on different ratios, is shown in Figure 4. 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 data, e.g. AddSent.

5 Related Work

Question answering. Extractive QA is the most common QA task, where the answer is a text span in the context. Various datasets have been proposed, such as SQuAD 1.1 (Rajpurkar et al., 2016) and 2.0 (Rajpurkar et al., 2018), NewsQA (Trischler et al., 2017), TriviaQA (Joshi et al., 2017), and NaturalQuestions (Kwiatkowski et al., 2019). These datasets motivate more works on QA models, e.g., end2end models like BiDAF, R-Net, and QANet (Seo et al., 2017; Wang et al., 2017; Yu et al., 2018). Pre-trained models become common approaches recently, such as BERT, RoBerta, and SpanBert (Devlin et al., 2019; Liu et al., 2019; Joshi et al., 2020). They realize remarkable promotions bnefited from huge corpora. 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 generate misspelled tokens in character level to attack models (Liang et al., 2018; Ebrahimi et al., 2018; Li et al., 2019),

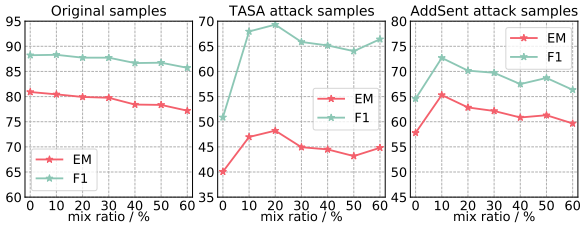

Figure 4: The performance of BERT fine-tuned on the original SQuAD 1.1 data **mixed with adversarial samples from TASA** in different ratios (%), evaluated on original dev samples, adversarial samples from TASA and AddSent. We expect a slight influence on original samples, while promotions on the later two sets.

but they are easy to be defended (Pruthi et al., 2019; Jones et al., 2020). More studies use more sophisticated toke-level (Ren et al., 2019; Jin et al., 2020; Alzantot et al., 2018; Zang et al., 2020; Li et al., 2021) or phrase/sentence-level perturbations (Iyyer et al., 2018; Chen et al., 2021), with some strategies to guarantee the text meaning. However, none of them shows their effectiveness on QA tasks.

There are efforts on attacking QA models. AddSent (Jia and Liang, 2017) is an adversarial QA dataset where a distracting sentence is added by annotators. Wallace et al. propose a human-in-loop method where annotators need to interact with models and fool it. 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) uses gradient-guided search to find out input-agnostic text that can mislead models for a specific question type. Our TASA differs from them as it bridges context and question to attack more efficiently and suits more general conditions.

6 Conclusion

We present TASA, an automatic attack method to produce adversarial context for QA models. It generates twin answer sentences to fool QA models and misguides them to an incorrect answer by leveraging their pitfall on keyword matching. It first replaces the keywords of answer sentence with synonyms. It then adds a distracting answer sentence (DAS) modified from the answer sentence by changing the subjects or objects associated with the answer. In experiments, TASA achieves remarkable attack performance on four datasets and two victim models. We will investigate attacks perturbing both the context and question in the future.

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.

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. *arXiv preprint arXiv:1910.09753*.

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.

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*, volume 34, pages 8018–8025.

Erik Jones, Robin Jia, Aditi Raghunathan, and Percy Liang. 2020. Robust encodings: A framework for combating adversarial typos. In *Proceedings of ACL 2020*, pages 2752–2765.

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.

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.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. Ppdb 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of ACL-IJCNLP 2015*, pages 425–430.

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*, (8):9.

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*.

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.

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[2] to implement the model and the *bert base uncased* version of BERT model[3] 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 dataset, we set the maximum input sequence length as 384, using an Adam optimizer whose initial learning rate 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.

BiDAF We use the model implementation provided by AllenNLP[4]. 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 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[5] combined with SpaCy[6] to find out the coreferences in contexts.

Perturbation on answer sentences The lemmas and POS tags of different are obtained via SpaCy. The POS tag set $\mathcal{K}$ used to get keywords includes *"VERB", "NOUN", "ADJ", "ADV"*. When perturbing a token with its synonyms, we use pyinflect[7] 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.

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 6: Values of hyperparameters used in TASA.

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; 3) the perturbation time exceeds our restriction, 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[8] to get USE similarity and a small size GPT2 model[9] to get the PPL.

Answer justification model F_J We use a *base RoBerta* model fine-tuned on the original SQuAD 2.0 dataset[10] as F_J for both SQuAD 1.1 and SQuAD 2.0, 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 input, then the current sample is regarded as unanswerable. For the rest two datasets, we use other Roberta models fine-tuned on the corresponding training set along with negative samples (unanswerable samples) in the same size as the original training set. I.e. each negative sample has a question obtained by randomly sampling from the whole dataset that is not belonged to the given context, which will be labeled as *"unanswerable"* later. We follow the same training pattern as SQuAD 2.0 to fine-tune the model, where the model need to have the capability of both answering answerable samples and output *"unanswerable"* label for unanswerable samples.

We list all hyperparameter values used by TASA

[2] https://github.com/huggingface/transformers
[3] https://huggingface.co/bert-base-uncased
[4] https://github.com/allenai/allennlp
[5] https://github.com/huggingface/neuralcoref
[6] https://spacy.io
[7] https://spacy.io/universe/project/pyinflect/
[8] https://tfhub.dev/google/universal-sentence-encoder/4
[9] https://huggingface.co/gpt2
[10] https://huggingface.co/deepset/roberta-base-squad2

Datasets	|C|	|Q|	Train size	Dev size
SQuAD 1.1	11	137	87,599	10,570
SQuAD 2.0	11	135	86,821	5,928
NewsQA	8	599	74,160	4,212
Natural Questions	9	153	104,071	12,836

Table 7: The statistics of 4 datasets used in our experiments. |C| is the average length of context, |Q| is the average length of question, both in token.

method in Table 6, which is obtained by empirical tuning. Each time attack on the whole SQuAD 1.1 dataset takes about 10 hours using BERT as the victim model, or about 1 day using BiDAF as the victim model, both on one V100 GPU. We also publish our code anonymously at https://anonymous.4open.science/r/TASA/.

A.3 Baselines

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 editing 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 implement is using its official code directly as it already contains the function to attack QA dataset.

A.4 Datasets

We provide some statistics about 4 datasets we used in Table 7. Note that we abandon all unanswerable questions from the original SQuAD 2.0 Dataset and only use answerable samples here, because TASA only targets on attacking answerable samples.

B Additional Results

B.1 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. We provide the compositions of samples generated by TASA on different datasets in Table 8. It can be found that PAS+DAS compose about half of them adversarial samples, while samples that only contain DAS consist of the majority of the rest.

We also provide the performance of QA models on different compositions on each dataset, which is illustrated in Table 9. It can be found that PAS+DAS has the best attack success rate among all 3 kinds of compositions, while only using PAS or DAS lowers the performance of models with a smaller scale. It proves the necessity of combining the two folds of pitfall we discussed in §2 into the

Dataset	**source**	BERT	BiDAF
SQuAD 1.1	PAS+DAS	50.2	54.4
	PAS	8.9	9.3
	DAS	40.9	36.3
SQuAD 2.0	PAS+DAS	47.5	51.5
	PAS	7.5	7.9
	DAS	45.0	40.6
NewsQA	PAS+DAS	40.8	44.0
	PAS	19.9	21.1
	DAS	39.3	34.9
NQ	PAS+DAS	54.1	54.9
	PAS	19.5	19.9
	DAS	26.4	25.2

Table 8: The composition ratios (%) of adversarial samples generated by TASA on four datasets using BERT or BiDAF as victim models. PAS+DAS: both PAS and DAS are included in current sample; PAS: only PAS is applied in current sample; DAS: only DAS is applied in current sample.

Dataset	**source**	BERT		BiDAF	
		EM	F1	EM	F1
SQuAD 1.1	PAS+DAS	22.75	32.98	26.47	36.46
	PAS	51.06	63.48	37.46	50.08
	DAS	58.86	70.03	55.28	68.69
SQuAD 2.0	PAS+DAS	26.13	38.00	27.30	37.95
	PAS	45.70	61.04	35.53	48.50
	DAS	58.81	71.27	54.72	68.83
NewsQA	PAS+DAS	28.42	41.77	24.84	38.38
	PAS	30.23	43.80	25.49	39.89
	DAS	46.01	60.52	39.95	52.78
NQ	PAS+DAS	33.09	45.27	32.29	44.25
	PAS	47.55	62.97	43.24	57.81
	DAS	60.80	70.25	53.72	62.59

Table 9: The performance of QA models on different 28.42 of adversarial samples generated by TASA, on all 4 datasets.

adversarial attack on the QA task.

C Qualitative Samples

We provide some samples generated by TextFooler, T3 and TASA along with corresponding model predictions in Table 10, Table 11. We also provide the instruction screenshot for human evaluation in Figure 5 and Figure 6.

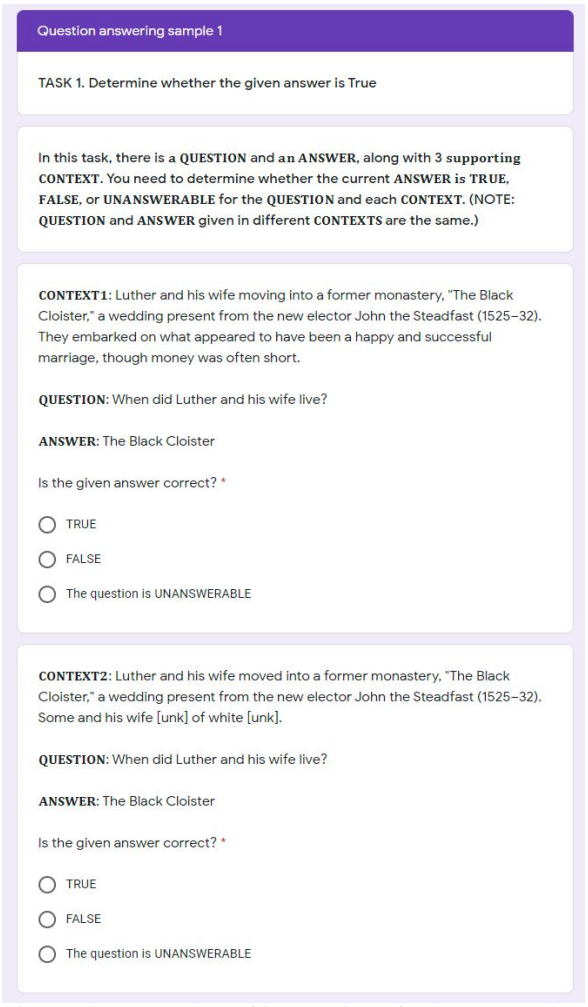

Figure 5: Screenshot of instructions for human evaluation (part1).

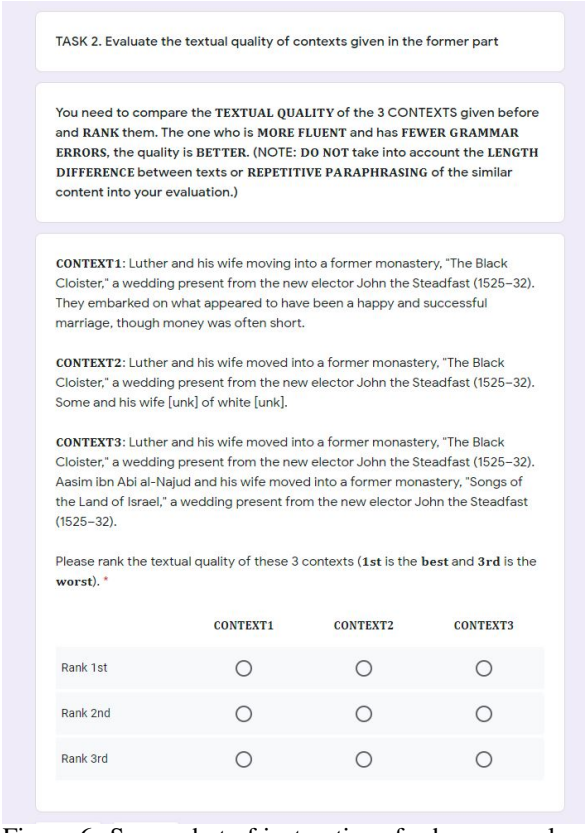

Figure 6: Screenshot of instructions for human evaluation (part2).

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 10: 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 11: 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.