Attacking Open-domain Question Answering by Injecting Misinformation

Liangming Pan
University of California, Santa Barbara
liangmingpan@ucsb.edu

Wenhu Chen
University of Waterloo
wenhuchen@uwaterloo.ca

Min-Yen Kan
National University of Singapore
kanmy@comp.nus.edu.sg

William Yang Wang
University of California, Santa Barbara
william@cs.ucsb.edu

Abstract

With a rise in false, inaccurate, and misleading information in propaganda, news, and social media, real-world Question Answering (QA) systems face the challenges of synthesizing and reasoning over *misinformation-polluted contexts* to derive correct answers. This urgency gives rise to the need to make QA systems robust to misinformation, a topic previously unexplored. We study the risk of misinformation to QA models by investigating the sensitivity of open-domain QA models to corpus pollution with misinformation documents. We curate both human-written and model-generated false documents that we inject into the evidence corpus of QA models, and assess the impact on the performance of these systems. Experiments show that QA models are vulnerable to even small amounts of evidence contamination brought by misinformation, with large absolute performance drops on all models. Misinformation attack brings more threat when fake documents are produced at scale by neural models or the attacker targets on hacking specific questions of interest. To defend against such a threat, we discuss the necessity of building a misinformation-aware QA system that integrates question-answering and misinformation detection in a joint fashion.

1 Introduction

A typical Question Answering (QA) system (Chen et al., 2017; Yang et al., 2019; Karpukhin et al., 2020; Yamada et al., 2021; Glass et al., 2022) starts by retrieving a set of relevant *context documents* from the Web, which is then examined by a machine reader to identify the correct answer. Existing works typically equate Wikipedia as the web corpus. Therefore, all retrieved context documents are assumed to be clean and trustable. However, real-world QA faces a much noisier environment, where the web corpus is tainted with *misinformation*. This includes unintentional factual mistakes made by human writers and deliberate disinformation intended to deceive. Aside from human-created misinformation, we are also facing the inevitability of AI-generated misinformation. With the continuing progress in text generation (Radford et al., 2019; Brown et al., 2020; Lewis et al., 2020; Ouyang et al., 2022; OpenAI, 2023), realistic-looking fake web documents can be generated at scale by malicious actors (Zellers et al., 2019; Huang et al., 2023; Pan et al., 2023).

The presence of misinformation — no matter deliberately created or not, no matter human-written or machine-generated — affects the reliability of the QA system by bringing in *contradicting* information. As shown in Figure 1 (right side), when both real and fake information are retrieved as context documents, the QA models can be easily confused by the contradicting answers given by both parties, given the fact that they do not have the ability to identify fake information and reason over contradicting contexts. Although current QA models often achieve promising performance under the idealized case of clean contexts, we argue that they may easily fail under the more realistic case of misinformation-mixed contexts.

We study the risks of misinformation to question answering by investigating how QA models behave on a *misinformation-polluted web corpus* that is mixed with both real and fake information. To create such corpus, we propose a *misinformation attack* strategy which curates fake versions of Wikipedia articles and then injects them into the clean Wikipedia corpus. For a Wikipedia article P, we create its fake version P' by modifying information in P, such that: 1) certain information in P' contradicts with the information in P, and 2) P' is fluent, consistent, and looks realistic. We study both human-written and model-generated misinformation. For the human-written part, we ask Mechanical Turkers to create fake articles by modifying original wiki articles. For the model-generation part, we propose a strong rewriting model, namely

Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), pages 525–539
November 1–4, 2023.

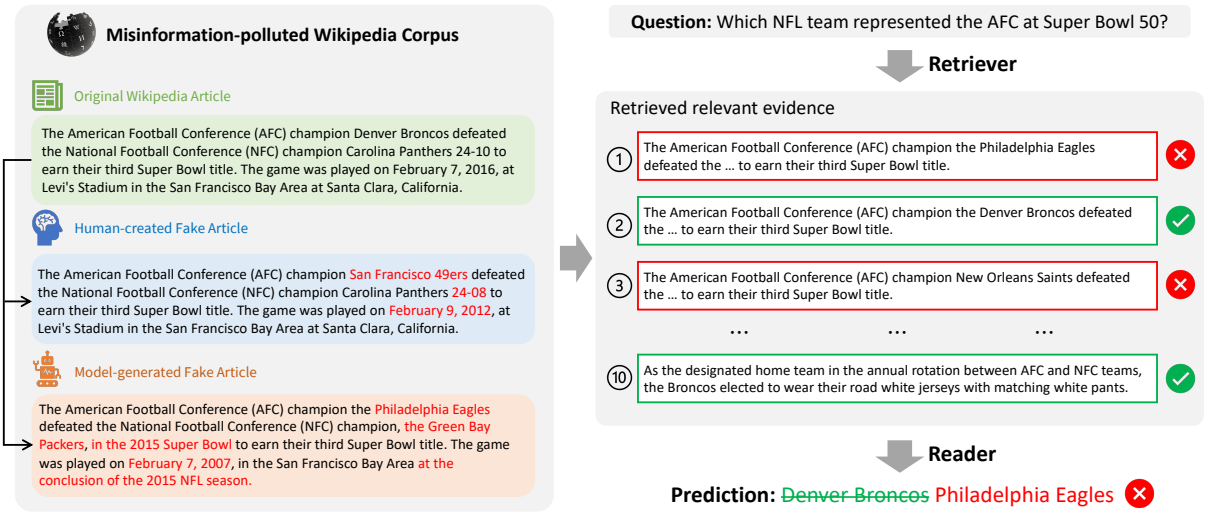

Figure 1: Our framework injects human-created and model-generated misinformation documents into the QA evidence repository (left) and evaluates the impact on the performance of open-domain QA systems (right).

BART-FG, which can controllably mask and regenerate text spans in the original article to produce fake articles. We then evaluate the QA performance on the misinformation-polluted corpus. A robust QA model should be able to deal with misinformation and properly handle contradictory information.

Unfortunately, from extensive experiments, we find that existing QA models are vulnerable to misinformation attacks, regardless of whether the fake articles are manually written or model-generated. The state-of-the-art open-domain QA pipeline, with ColBERT (Santhanam et al., 2022a) as the retriever and the DeBERTa (He et al., 2023) as the reader, suffers from noticeable performance drops in five different attack modes. Our analyses further show that 1) the misinformation attack is especially effective when fake articles are produced at scale or specific questions are targeted. 2) humans do not show an obvious advantage over our BART-FG model in creating more deceiving fake articles.

In summary, we investigate the potential risk of open-domain QA under misinformation. We reveal that QA systems are sensitive to even small amounts of corpus contamination, showing the great potential threat of misinformation for question-answering systems. We end by discussing the necessity of building a misinformation-aware QA system. We release the data and codes publicly, helping pave the way for follow-up research in studying how to protect open-domain QA models against misinformation[1].

[1] https://github.com/teacherpeterpan/ContraQA/

2 Related Work

Open-domain Question Answering. To answer a question, open-domain QA systems employ a *retriever-reader* paradigm that first retrieves relevant documents from a large evidence corpus and then predicts an answer conditioned on the retrieved documents. Promising advances have been made towards improving the reader models (Yang et al., 2019; Izacard and Grave, 2021) and neural retrievers (Lee et al., 2019; Guu et al., 2020; Santhanam et al., 2022b). However, since Wikipedia is used as the evidence corpus, previous works take for granted the assumption that the retrieved documents are trustworthy. This assumption becomes questionable with the rapid growth of fake and misleading information in the real world. In this work, we take the initiative to study the potential threat that misinformation can bring to QA systems, calling for a new direction of building misinformation-immune QA systems.

Improving Robustness for QA. Our work aims to analyze vulnerabilities to develop more robust QA models. Current QA models demonstrate brittleness in different aspects. QA models often rely on spurious patterns between the question and context rather than learning the desired behavior. They might ignore the question entirely (Kaushik and Lipton, 2018), focus primarily on the answer type (Mudrakarta et al., 2018), or ignore the "intended" mode of reasoning for the task (Jiang and Bansal, 2019; Niven and Kao, 2019). QA models also generalize badly to out-of-domain (OOD)

data (Kamath et al., 2020). For example, they often make inconsistent predictions for different semantically equivalent questions (Gan and Ng, 2019; Ribeiro et al., 2019). Similar to our paper, a few prior works (Chen et al., 2022; Weller et al., 2022; Abdelnabi and Fritz, 2023) investigated the robustness of QA models under conflicting information. For example, Longpre et al. (2021) shows QA models are less robust to OOD data where the contextual information contradicts the learned information. Different from these works, we study from a new angle of QA robustness: the vulnerability of QA models under misinformation.

Combating Neural-generated Misinformation. Advanced text-generation models offer a powerful tool for augmenting the training data of downstream NLP applications (Pan et al., 2021; Chen et al., 2023). However, these models also pose a risk of being exploited for malicious activities, such as generating convincing fake news (Zellers et al., 2019), fraudulent online reviews (Garbacea et al., 2019; Adelani et al., 2020), and spam. Even humans find it struggle to detect such synthetically-generated misinformation (Clark et al., 2021). When produced at scale, neural-generated misinformation can pose threats to many NLP applications. For example, a recent work by (Du et al., 2022) finds that synthetic disinformation can significantly affect the behavior of modern fact-checking systems. In this work, we study the risk of neural-generated misinformation to QA models.

3 Misinformation Documents Generation

We simulate the potential vulnerability of question-answering models to corpus pollution with misinformation documents by injecting both human-written and model-generated false documents into the evidence corpus, and assess the impact on the performance of these systems. We base our study on the SQuAD 1.1 (Rajpurkar et al., 2016) dataset, one of the most popular benchmarks for evaluating QA systems. We use all the 2,036 unique Wikipedia passages from the validation set for our study. For each Wikipedia passage $\mathcal{P}^R$, we create a set of N fake passages $(\mathcal{P}^F_1, \cdots, \mathcal{P}^F_N)$ by modifying some information in $\mathcal{P}^R$, with the requirement that each fake passage look realistic while containing contradicting information with $\mathcal{P}^R$.

We use two different ways to create fake passages: 1) **via human edits**: we ask online workers from Amazon Mechanical Turk (AMT) to produce fake passages by modifying the original passage, and 2) **via BART-FG**: our novel generative model BART-FG, which iteratively masks and regenerates text spans from the original passage to produce fake passages.

3.1 Manual Creation of Fake Passages

To solicit human-written deceptive fake passages, we release 2K HITs (human intelligence tasks) on the AMT platform, where each HIT presents the crowd-worker with one passage $\mathcal{P}^R$ in the SQuAD validation set. We ask workers to modify the contents of the given passage to create a fake version, following the below guidelines:

- The worker should make *at least M edits* at different places, where M equals to one plus the number of sentences in the contexts $\mathcal{C}^R$.
- The worker should make at least one *long edit* that rewrites at least half of a sentence.
- The edits should modify key information to make it *contradict with the original*, such as time, location, purpose, outcome, reason, etc.
- The modified passage should be *fluent and look realistic*, without commonsense errors.

To select qualified workers, we restrict our task to workers who are located in five native English-speaking countries[2], and who maintain an approval rating of at least 90%. To ensure the annotations fulfil our guidelines, we give ample examples in our annotation interface with detailed explanations to help workers understand the requirements. The detailed annotation guideline is in Appendix A. We also hired three computer science major graduate students as human experts to validate a HIT's annotation. In the end, 104 workers participated in the task. The average completion time for one HIT is 5 minutes, and payment is $1.0 U.S. dollars/HIT. The average acceptance rate was 93.75%.

3.2 Model Generation of Fake Passages

Aside from human-written misinformation, we also want to explore the threat of machine-generated misinformation to QA. This source may be more of a concern than human-created misinformation, since they can easily be produced at scale. Recently introduced large-scale generative models, such as GPT2 (Radford et al., 2019), BART (Lewis et al., 2020), and Google T5 (Raffel et al., 2020), can produce realistic-looking texts, but they do not lend themselves to producing controllable generation

[2]Australia, Canada, Ireland, United Kingdom, USA

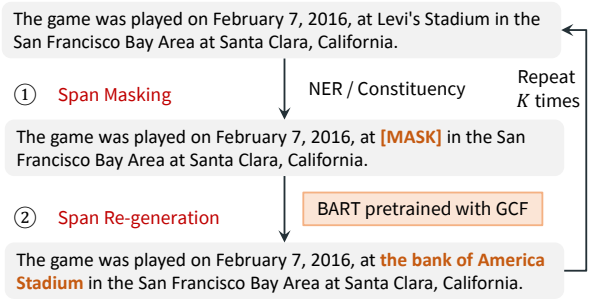

Figure 2: Overview of the BART-FG model, illustrated by an example sentence.

that only replaces the key information with contradicting contents. Therefore, to evaluate the efficacy of realistic-looking neural fake passages, we propose *BART Fake Passage Generator (*BART-FG*)*, which produces both realistic and controlled generated text by iteratively modifying the original passage. As shown in Figure 2, for each sentence S of the original passage, BART-FG produces its fake version S' via a two-step process:

1) Span Masking. We first obtain a set of candidate text spans from the input sentence. We then randomly select a span and replace it with a special mask token `[MASK]`. We employ two different ways to get the candidate spans. 1) *NER*: we use Spacy[3] to extract name entities as the candidate spans. 2) *Constituency*: we apply the constituency parser implemented in AllenNLP[4] to extract constituency spans from the input sentence as the candidate spans. We choose to mask named entities / constituency phrases instead of random spans because: 1) they represent complete semantic units such as "Super Bowl 50", which avoids meaningless random phrases such as "Bowl 50"; and 2) they often represent important information in the sentence — such as time, location, cause, etc.

2) Span Re-generation. We fill in the mask by generating a phrase different from the masked phrase. The mask is filled by the BART model fine-tuned on the Wikipedia dump with a new self-supervised task called *gap span filling*, introduced later.

The above pipeline is iteratively run for K times to generate sentence S' from S. We choose to make the edits iteratively rather than in parallel to model interaction between multiple edits. For example, in Figure 2, if the previous edit changes "Santa Clara" to "Atlanta", the next edit can choose to change "California" into "Georgia" to make the contents more consistent and realistic.

Gap Span Filling (GSF) Pre-Training. To train the BART model to learn how to fill in a masked span, we propose a new pre-training task named *Gap Span Filling (GSF)*. For each article in the Wikipedia dump that consists of T sentences $[S_1, S_2, \cdots, S_T]$, where each sentence is a word sequence $S_t = [w_1^t, \cdots, w_{|S_t|}^t]$, we construct the following training data for $t = 2, \cdots, T-1$:

Input: $S_1, S_{t-1}, w_{1:a-1}^t$, `[MASK]`, $w_{b+1:|S_t|}^t, S_{t+1}$
Output: $w_{a:b}^t = [w_a^t, \cdots, w_b^t]$

where the output represents a masked constituency or named entity span that starts with the a-th word and ends with b-th word. The input is the concatenation of the first sentence S_1, the previous sentence S_{t-1}, the current sentence S_t with one span being masked, and the subsequent sentence S_{t+1}. The BART model is fine-tuned to predict the output given the input on the entire Wikipedia dump. This task trains the BART model to predict the masked constituency / named entity span, given both global contexts (S_1) and local contexts (S_{t-1}, S_{t+1}). We use the `facebook/bart-large` model provided by Hugging Face (406M parameters).

3.3 Analysis of the Generated Fake Passages

Table 1 shows examples from six original passages with their corresponding fake versions, which represent six common types of modifications made by the human and the model, explained as follows:

(1) **Entity Replacement**: replacing entities (*e.g.*, person, location, time, number) with other entities with the same type, a common type of modification for both human edits and BART-FG.
(2) **Verb Replacement**: replacing verb or verb phrase with its antonymic meaning, *e.g.*, "force these children to" → "prevent these children from".
(3) **Adding Restrictions**: create contradiction by inserting additional restrictions to the original content, *e.g.*, "every day" → "every day but Sunday".
(4) **Sentence Rephrasing**: rewrite the whole sentence to express a contradicting meaning, exemplified by (4). This is common in human edits but rarely seen in model-generated passages, since this requires deep reading comprehension.
(5) **Disrupting Orders**: make a contradiction by disrupting some property of the entities; *e.g.*, ex-

[3] https://spacy.io/usage/linguistic-features#named-entities
[4] https://demo.allennlp.org/constituency-parsing

#	Original Contexts	Contradicting Contexts
(1)	The game was played on February 7, 2016 at Levi's Stadium in the San Francisco Bay Area at Santa Clara, California.	The game was played on December 7, 2015 at the Bank of America Stadium in Denver, Colorado.
(2)	... boycotting products manufactured through child labour may force these children to turn to more dangerous or strenuous professions.	... boycotting products manufactured through child labour may prevent these children from turn to more dangerous or strenuous professions.
(3)	Tesla worked every day from 9:00 am until 6:00 pm or later.	Tesla worked every day but Sunday from 9:00 am until 6:00 pm or later.
(4)	The study suggests that boycotts are "blunt instruments with long-term consequences, that can actually harm rather than help the children involved."	The study did not find any major negative repercussions from boycotts, however, and found that boycotting is the best solution.
(5)	A key distinction between analysis of algorithms and complexity theory is that the former is devoted to ..., whereas the later asks a more general question of ...	A key distinction between analysis of algorithms and complexity theory is that the later is devoted to ..., whereas the former asks a more general question of ...
(6)	On the whole, Eisenhower's support of the nation's fledgling space program was officially modest until the Soviet launch of Sputnik in 1957, gaining the Cold War enemy enormous prestige around the world.	On the whole, Eisenhower's support of the nation's fledgling MK Ultra was officially terminated until the Cuban missile crisis, gaining the Cold War enemy enormous admiration in less developed nations.

Table 1: Examples of original passages and their corresponding fake versions, where the information changes are highlighted. These examples represent six common types of created misinformation.

ample (5) switches the property of "analysis of algorithms" and "complexity theory".
(6) **Consecutive Replacements**: humans are better in making consecutive edits to create a contradicting yet coherent sentences, exemplified by (6).

4 Corpus Pollution with Misinformation

Given the fake passages curated by both human and our BART-FG model, we now study how extractive QA models behave under an evidence corpus that is polluted with misinformation. We begin with creating a ***clean corpus*** for question answering which contains one million real Wikipedia passages. We obtain the Wikipedia passages from the 2019/08/01 Wikipedia dump provided by the Knowledge-Intensive Language Tasks (KILT) benchmark (Petroni et al., 2021), in which the Wikipedia articles have been pre-processed and separated into paragraphs. We sample 1M paragraphs from KILT and ensure that all the 20,958 Wikipedia passages in the SQuAD dataset are included in the corpus. We then explore the following five ways of polluting the clean corpus with human-created and synthetically-generated false documents.

• **Polluted-*Human***. In Section 3.1, we asked human annotators to create a fake version for each passage in the SQuAD dev set. We inject those 2,023 fake passages into the clean corpus.
• **Polluted-*NER***. We use BART-FG to generate 10 fake passages for each real passage in the SQuAD dev set, using NER to get candidate spans. We mask and re-generate all candidate spans to create each fake passage. Nucleus sampling (Holtzman et al., 2020) is used to ensure diversity in generation, giving us 18,233 non-repetitive fake passages in total. We inject them into the clean corpus.
• **Polluted-*Constituency***. We generate 10 fake passages for each real passage using constituency parsing to get candidate spans in BART-FG. Since there are far more constituency phrases than named entities in a sentence, to ensure efficiency, we fix the number of replacements $K = 3$ for each sentence. We get 19,796 non-repetitive fake passages and inject them into the clean corpus.
• **Polluted-*Hybrid***. We inject all of the above-generated fake passages into the clean corpus.
• **Polluted-*Targeted***. In the above settings, the attacker (human or BART-FG model) tries to create misleading fake information *without* knowing the target questions. However, in another attack mode, attackers have *particular questions of interest* that they want to mislead the QA system into getting wrong answers. To explore how QA systems react to such attacks, in this setting we assume the attacker targets the questions in the SQuAD dev set. We then create fake passages by masking and re-generating the *answer spans* of these questions using BART-FG. Through this, we get 10,101 fake passages and insert them into the clean corpus.

5 Models and Experiments

We now how question answering models behave under such misinformation-polluted environment. To answer a given question, the QA systems employ

Evidence Corpus	RoBERTa (Liu et al., 2019)		SpanBERT (Joshi et al., 2020)		Longformer (Beltagy et al., 2020)		ELECTRA (Clark et al., 2020)		DeBERTaV3 (He et al., 2023)	
	EM	**F1**	**EM**	**F1**	**EM**	**F1**	**EM**	**F1**	**EM**	**F1**
Clean	53.72	59.45	55.58	61.30	56.40	61.68	55.41	61.52	62.30	67.85
Polluted-*Human*	48.47	56.84	51.20	58.26	52.39	59.03	51.43	59.04	58.16	64.82
Polluted-*Constituency*	46.07	54.63	46.47	55.38	47.69	56.07	45.84	55.05	50.88	59.63
Polluted-*NER*	42.23	50.34	44.01	52.64	45.25	53.50	43.40	52.54	48.74	57.16
Polluted-*Hybrid*	41.96	50.17	44.18	53.61	44.93	53.98	42.69	52.81	48.14	57.63
Polluted-*Targeted*	25.29	34.22	25.55	34.76	26.92	35.84	25.42	34.80	29.52	38.80

Table 2: Effects of different modes of misinformation attacks on the open-domain QA performance in SQuAD.

a *retrieve-then-read* pipeline that first retrieves N (we set $N = 5$) relevant contextual documents from the evidence corpus and then predicts an answer conditioned on the retrieved documents. For document retrieval, we apply the widely-used sparse retrieval based on BM25, implemented with the Pyserini toolkit (Lin et al., 2021). For question answering, we consider five state-of-the-art QA models with public code that achieved strong results on the public leader board of SQuAD: *RoBERTa-large* (Liu et al., 2019), *Span-BERT* (Joshi et al., 2020), *Longformer* (Beltagy et al., 2020), *ELECTRA* (Clark et al., 2020), and *DeBERTa-V3* (He et al., 2023). We use their model checkpoints fine-tuned on the SQuAD training set from the Hugging Face library. We use the standard Exact Match (EM) and F_1 metrics to measure QA performance.

5.1 Main Results

In Table 2, we show the performance of different QA models on the SQuAD dev set under the clean evidence corpus (*Clean*) and the performance under the misinformation-polluted corpus (*Polluted*). We have two major observations.

For all models, we see a noticeable performance drop when generated fake passages are introduced into the clean evidence corpus: the smallest average performance drop is 7.72% in relative EM value (Polluted-*Human*), while the largest drop is 53.19% (Polluted-*Targeted*). This indicates that QA models are sensitive to misinformation attack; even limited amounts of injected fake passages comprising 0.2% (*Human*) to 4.0% (*Hybrid*) of the entire corpus can noticeably affect downstream QA performance. It reveals the potential threat of misinformation to current QA systems, given the fact that they are not trained to differentiate misinformation.

Polluted-*Targeted* causes a more significant performance drop compared to the most effective question-agnostic attack (Polluted-*Hybrid*) ($\sim$53% v.s. $\sim$22% relative EM drop), indicating that QA models are more vulnerable under question-targeted misinformation attack. This reveals that the misinformation attack brings more threat when the attacker wants to alter the answers produced by QA systems for particular questions of interest. For the other four question-agnostic settings where the pollution is not targeted on specific questions, we still observe a noticeable EM drop ($\sim$20%) for all models. Among them, Polluted-*NER* causes more performance drop than Polluted-*Constituency*, showing that generating misinformation by replacing named entities is more effective than replacing constituency spans. This is probably due to the nature of the SQuAD dataset, where most of the answer spans are named entities.

5.2 Impact of misinformation on retriever

The success of the misinformation attack relies on the premise that fake passages can be retrieved from the polluted corpus by the retriever. To validate this, we first define a fake passage P as the *misleading evidence* for the question Q if P contains a fabricated answer for Q. We then report in Table 3 the percentage of misleading evidence in the top-k retrieved passages (F@k, for $k \in \{1, 5\}$) for the BM25 retriever. We find that both F@1 and F@5 are very high, while the likelihood of the ground-truth true evidence appearing in the top-1 (R@1) and top-5 (R@5) decreases significantly for polluted corpus. The results show that the injected fake passages can be easily retrieved as evidence for downstream question answering. QA models, without the fact-checking capability, can thus be easily misled by such misinformation.

However, BM25 only relies on syntactic features and cannot be optimized for specific tasks. Is the misinformation attack also effective for trainable

Evidence Corpus	BM25 + DeBERTa-V3						ColBERT-V2 + DeBERTa-V3					
	R@1	R@5	F@1	F@5	EM	F1	R@1	R@5	F@1	F@5	EM	F1
Clean	57.46	75.97	—	—	62.30	67.85	59.30	80.40	—	—	67.54	73.17
Polluted-*Human*	47.24	74.21	7.11	44.58	58.16	64.82	41.95	75.91	11.07	43.71	59.02	65.23
Polluted-*Constituency*	30.21	49.50	23.64	46.54	50.88	59.63	28.63	47.50	25.01	48.00	49.17	58.66
Polluted-*NER*	28.30	48.88	21.33	48.79	48.74	57.16	25.88	44.34	22.86	50.01	46.41	54.31
Polluted-*Hybrid*	25.67	45.60	26.53	53.45	48.14	57.63	23.01	42.69	23.80	55.12	45.46	54.03
Polluted-*Targeted*	15.04	45.70	46.60	72.86	29.52	38.80	16.90	40.09	47.27	74.56	28.93	37.12

Table 3: Effects of different modes of misinformation attacks on the *BM25* and *ColBERT-V2* retrievers.

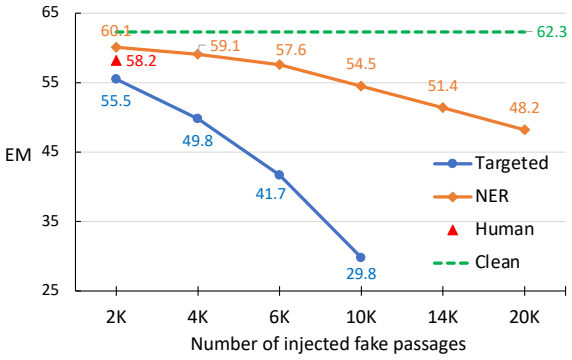

Figure 3: The EM score for DeBERTa-V3 model with different number of injected fake passages N.

dense retrievers? To explore this, we use ColBERT-V2 (Santhanam et al., 2022a), the state-of-the-art dense retriever that independently encodes the question and the passage using BERT and then employs a late interaction architecture to model their similarity. We use the ColBERT pretrained on MS-MARCO (Nguyen et al., 2016) and fine-tune it with (question, context) pairs from SQuAD training set as positive samples and (question, random context) as negative samples. The retrieval and QA performance are reported in Table 3.

We find that misinformation attack also affects the ColBERT retriever, decreasing R@1 and R@5 for all settings, with high percentage of fake passages being retrieved as reflected by F@1 and F@5. The results also suggest that ColBERT is less resistant to misinformation attack compared to BM25. In the clean corpus, ColBERT outperforms BM25 in both the retrieval and the downstream QA performance. However, in all polluted corpus, the relative performance drop for ColBERT is larger than the drop for BM25. The possible explanation is: without the ability to identify fake information, a more "accurate" retriever tends to retrieve more seemingly relevant but false documents, making it less robust to misinformation attack.

5.3 Impact of the size of injected fake passages

As confirmation that misinformation attacks work as expected, we depict in Figure 3 how the DeBERTa model performance changes when different number of fake passages are injected into the evidence corpus. We find that the EM score steadily drops with more fake passages for both the question-targeted attack (*Targeted*) and the question-agnostic attack (*NER*). However, the former causes a much sharper trend of decrease, which further validates that misinformation attack is more deadly with a better knowledge of the target questions. Through this study, we conclude that misinformation may have a more severe impact on QA systems when they are produced at scale. With the availability of pretrained text generation models, producing fluent and realistic-looking contexts now has a little marginal cost. This brings an urgent need to effectively defend against neural-generated misinformation in question answering.

5.4 Which is more deceiving: human- or model-generated misinformation?

We then investigate which is more deceiving to QA models: human or neural misinformation? To study this, we let the QA model to answer each question Q under the context $\mathcal{C} = \{\mathcal{P}^{\mathcal{R}}, \mathcal{P}^{\mathcal{H}}, \mathcal{P}^{\mathcal{C}}, \mathcal{P}^{\mathcal{N}}\}$, where P^R is the real passage that contains the correct answer, and $\mathcal{P}^{\mathcal{H}}, \mathcal{P}^{\mathcal{C}}, \mathcal{P}^{\mathcal{N}}$ are the corresponding fake versions of P^R produced by human, BART-FG (NER), and BART-FG (Constituency), respectively. We then analyze the source (which fake passage) of the incorrect answer when the model makes an error. If all three methods create equally deceiving fake passages, we expect to observe a uniform distribution of the error sources.

The distribution of error sources in Figure 4 shows that the most wrong answers are extracted from the model-generated fake passage. Human-

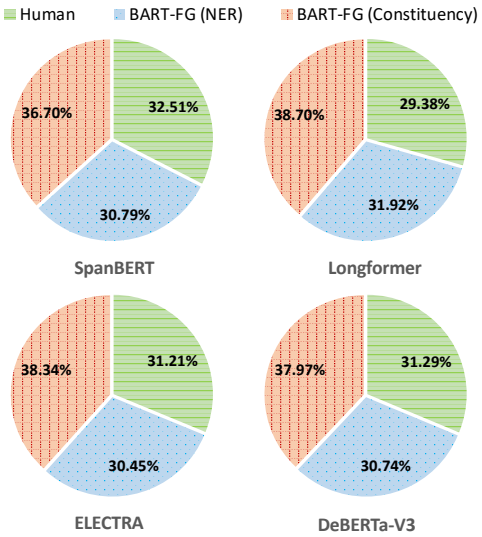

Figure 4: Distribution of error sources when the model is misled by a fake passage and gives a wrong answer.

created fake passages do not show an advantage over BART-FG in deceiving the QA models. This is counter-intuitive to what we find in Table 1 that humans make more subtle edits that require a deep level of reading comprehension, such as switching "former" and "latter" (Example 4), and changing "every day" to "every day but Sunday" (Example 3). A possible reason is that most questions in SQuAD are shallow in reasoning (Du et al., 2017). Therefore, replacing named entities/constituency phrases is sufficient in misleading QA models into getting the wrong answers for those questions.

5.5 Can misinformation deceive humans?

After showing the impact of misinformation attacks on QA systems, one natural question would be whether humans can also be distracted by misinformation during QA. To investigate this, we ran a study on Mechanical Turk where we presented crowd-workers with 500 randomly-sampled (question, context) pairs from the data in Section 5.4, *i.e.*, each context consists of the real passage along with three fake passages created by different methods. We call this test set *MisinfoQA-noisy* and the workers are asked to answer each of its questions. For comparison, we create another test set *MisinfoQA-clean* where each real passage is paired with three randomly sampled other Wikipedia passages.

Table 4 reports the EM and F1 for both human and different QA models. We find that all QA models suffer a large performance drop (∼20% in EM) in *MisinfoQA-noisy* compared to *MisinfoQA-clean*, showing that the models are largely distracted by the fake contexts rather than by the presence of additional contexts. Humans obtained an EM of 69.13 in *MisinfoQA-noisy*, which, though higher than most QA models' performance, also shows a significant drop when compared to the *MisinfoQA-clean* setting (86.57 EM). This shows that humans are also likely distracted by misinformation in QA, which demonstrates the challenge of distinguishing misinformation in question answering for lay readers, the quality of the generated fake passages, and the difficulty of detecting such an attack.

Setting	MisinfoQA-*noisy*		MisinfoQA-*clean*	
	EM	F1	EM	F1
Human	69.13	78.25	86.57	91.40
RoBERTa	61.20	70.44	77.06	83.88
SpanBERT	64.00	72.32	81.65	88.55
Longformer	67.83	75.15	82.80	90.72
ELECTRA	64.21	72.90	78.27	86.49
DeBERTa-V3	75.00	82.70	87.25	92.90

Table 4: QA performance under the reading comprehension settings with *clean* and *noisy* contexts.

6 Discussion and Future Work

Finally, we discuss three possible ways to defend the threat of misinformation for QA.

Knowledge source engineering. Despite being a trustful knowledge source, Wikipedia is insufficient to fulfill all the information needed in real-life question answering. Therefore, recent works (Piktus et al., 2021) started to use the web as the QA corpus. However, when transitioning to a web corpus, we no longer have the certainty that any document is truthful. Therefore, the corpora will require more careful curation to avoid misinformation. This also brings the need for future retrieval models to have the ability to assess the quality of the retrieved documents and prioritize more trustworthy sources.

Integrating fact-checking and QA. With the rise of misinformation online, automated fact-checking has received growing attention in NLP (Guo et al., 2022). Integrating fact-checking models into the pipeline of open-domain QA could be an effective countermeasure to misinformation, a direction neglected by prior works. A possible way is to detect potential false claims in retrieved contexts and lower their importance in downstream QA models.

Reasoning under contradicting contexts. It is common for humans to deal with contradictory information during information search. With the

presence of inaccurate and false information online, future models should focus on the ability to synthesize and reason over contradicting information to derive correct answers.

7 Conclusion

In this work, we evaluate the robustness of open-domain question-answering models when we contaminate the evidence corpus with misinformation. We studied two representative sources of misinformation: human-written disinformation and the misinformation-generated NLG models. Our studies reveal that QA models are indeed vulnerable under misinformation-polluted contexts. We also show that our BART-FG model can produce fake documents at scale that are as deceptive as humans. This poses a threat to current open-domain QA models in defending neural misinformation attacks.

Limitations

We identify two main limitations to our study. First, although SQuAD is a typical dataset for evaluating open-domain QA models, most of the SQuAD questions are factoid and shallow in reasoning, making it relatively easy to generate misinformation targeted at SQuAD. Our results show that BART-FG with named entity replacement can generate fake passages as deceptive as humans. However, the impact of model-generated misinformation may be over-estimated on the shallow factoid questions in SQuAD. Therefore, more QA datasets should be considered in future works, especially non-factoid questions with deeper reasoning.

Second, this work creates misinformation by revising key information of real articles in Wikipedia. However, there are other types of misinformation in the real world, such as hoaxes, rumors, or false propaganda. However, our proposed attack model can be easily generalized to study the threat of misinformation in other domains and in other forms.

Ethics Statement

We plan to publicly release the human- and model-generate fake documents and open-source the code and model weights for our BART-FG model. We note that open-sourcing the BART-FG model may bring the potential for deliberate misuse to generate disinformation for harmful applications. The human-written and model-generated fake documents can also be misused to generate disinformation. We deliberated carefully on the reasoning for open-sourcing and share here our three reasons for publicly releasing our work.

First, the danger of BART-FG in generating disinformation is limited. Disinformation is a subset of misinformation that is spread deliberately to deceive. Although we utilize the innate "hallucination" ability of current pretrained language models to create misinformation, our model are not specialized to generate harmful disinformation such as hoaxes, rumors, or false propaganda. Instead, our model focuses on generating conflicting information by iteratively editing the original passage to test the robustness of QA to misinformation.

Second, our model is based on the open-sourced BART model, which makes our model easy to replicate even without the released code. Given the fact that our model is a revised version of an existing publicly available model, it is unnecessary to conceal code or model weights.

Third, our decision to release follows the similar stance of the full release of another strong detector and state-of-the-art generator of neural fake news: Grover (Zellers et al., 2019)[5]. The authors claim that to defend against potential threats, we need threat modeling, in which a crucial component is a strong generator or simulator of the threat. In our work, we build an effective threat model for QA under misinformation. Followup research can build on our model transparency, further enhancing the threat model.

Acknowledgements

This work was supported by the National Science Foundation Award #2048122. The views expressed are those of the authors and do not reflect the official policy or position of the US government.

References

Sahar Abdelnabi and Mario Fritz. 2023. Fact-saboteurs: A taxonomy of evidence manipulation attacks against fact-verification systems. In *Proceedings of the 32nd USENIX Security Symposium*.

David Ifeoluwa Adelani, Haotian Mai, Fuming Fang, Huy H. Nguyen, Junichi Yamagishi, and Isao Echizen. 2020. Generating sentiment-preserving fake online reviews using neural language models and their human- and machine-based detection. In *Proceedings of the 34th International Conference on Advanced Information Networking and Applications (AINA)*, volume 1151, pages 1341–1354.

[5] https://thegradient.pub/why-we-released-grover/

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *CoRR*, abs/2004.05150.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1870–1879.

Hung-Ting Chen, Michael Zhang, and Eunsol Choi. 2022. Rich knowledge sources bring complex knowledge conflicts: Recalibrating models to reflect conflicting evidence. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2292–2307.

Jiaao Chen, Derek Tam, Colin Raffel, Mohit Bansal, and Diyi Yang. 2023. An empirical survey of data augmentation for limited data learning in NLP. *Transactions of the Association for Computational Linguistics (TACL)*, 11:191–211.

Elizabeth Clark, Tal August, Sofia Serrano, Nikita Haduong, Suchin Gururangan, and Noah A. Smith. 2021. All that's 'human' is not gold: Evaluating human evaluation of generated text. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7282–7296.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *Proceedings of 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1342–1352.

Yibing Du, Antoine Bosselut, and Christopher D. Manning. 2022. Synthetic disinformation attacks on automated fact verification systems. In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, pages 10581–10589.

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

Cristina Garbacea, Samuel Carton, Shiyan Yan, and Qiaozhu Mei. 2019. Judge the judges: A large-scale evaluation study of neural language models for online review generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3966–3979.

Michael R. Glass, Gaetano Rossiello, Md. Faisal Mahbub Chowdhury, Ankita Naik, Pengshan Cai, and Alfio Gliozzo. 2022. Re2g: Retrieve, rerank, generate. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 2701–2715.

Zhijiang Guo, Michael Sejr Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics (TACL)*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: retrieval-augmented language model pre-training. *CoRR*, abs/2002.08909.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*.

Kung-Hsiang Huang, Kathleen McKeown, Preslav Nakov, Yejin Choi, and Heng Ji. 2023. Faking fake news for real fake news detection: Propaganda-loaded training data generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 14571–14589.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 874–880.

Yichen Jiang and Mohit Bansal. 2019. Avoiding reasoning shortcuts: Adversarial evaluation, training, and model development for multi-hop QA. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2726–2736.

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 (TACL)*, 8:64–77.

Amita Kamath, Robin Jia, and Percy Liang. 2020. Selective question answering under domain shift. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5684–5696.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Divyansh Kaushik and Zachary C. Lipton. 2018. How much reading does reading comprehension require? A critical investigation of popular benchmarks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5010–5015.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6086–6096.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7871–7880.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, pages 2356–2362.

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

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2021. Entity-based knowledge conflicts in question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7052–7063.

Pramod Kaushik Mudrakarta, Ankur Taly, Mukund Sundararajan, and Kedar Dhamdhere. 2018. Did the model understand the question? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1896–1906.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *Proceedings of the NIPS Workshop on Cognitive Computation: Integrating neural and symbolic approaches*.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4658–4664.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *CoRR*, abs/2203.02155.

Liangming Pan, Wenhu Chen, Wenhan Xiong, Min-Yen Kan, and William Yang Wang. 2021. Zero-shot fact verification by claim generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 476–483.

Yikang Pan, Liangming Pan, Wenhu Chen, Preslav Nakov, Min-Yen Kan, and William Yang Wang. 2023. On the risk of misinformation pollution with large language models. *CoRR*, abs/2305.13661.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick S. H. Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 2523–2544.

Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Dmytro Okhonko, Samuel Broscheit, Gautier Izacard, Patrick Lewis, Barlas Oguz, Edouard Grave, Wen-tau Yih, and Sebastian Riedel. 2021. The web is your oyster - knowledge-intensive NLP against a very large web corpus. *CoRR*, abs/2112.09924.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research (JMLR)*, 21:140:1–140:67.

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

Marco Túlio Ribeiro, Carlos Guestrin, and Sameer Singh. 2019. Are red roses red? evaluating consistency of question-answering models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6174–6184.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022a. Colbertv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 3715–3734.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022b. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734, Seattle, United States. Association for Computational Linguistics.

Orion Weller, Aleem Khan, Nathaniel Weir, Dawn J. Lawrie, and Benjamin Van Durme. 2022. Defending against poisoning attacks in open-domain question answering. *CoRR*, abs/2212.10002.

Ikuya Yamada, Akari Asai, and Hannaneh Hajishirzi. 2021. Efficient passage retrieval with hashing for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 979–986.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with bertserini. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT), Demonstrations*, pages 72–77.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In *Proceedings of the 2019 Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 9051–9062.

A Human Annotation Guideline

A.1 Job Description

Given a paragraph from Wikipedia, modify some information in the paragraph to create a fake version of it. Here are the general requirements:

• You should make *at least M edits* at different places, where M is determined by the length of the passage and will show on the screen when you annotate each passage.

• You should make at least one *long edit* that rewrites at least half of a sentence.

• The edits should modify key information to make it *contradict with the original*, such as time, location, purpose, outcome, reason, etc.

• The modified paragraph should be *fluent and look realistic*, without commonsense errors.

A.2 Detailed Requirements

Figure 5 shows an example of modifications that fulfill all the annotation requirements. Detailed annotation instructions are as follows.

1) At least make N edits at different places. In the above example, there are a total of 5 edits:

- "an American football game" → "the 48th Super Bowl Game"
- "Denver Broncos" → "San Francisco 49ers"
- "on February 7, 2016, at Levi's Stadium in the San Francisco Bay Area at Santa Clara, California." → "Mercedes-Benz Superdome in New Orleans, Louisiana and was the first Super Bowl to be played in the United States."
- "the 50th" → "the NFL's 48th"
- "so that the logo could prominently feature the Arabic numerals 50." → "so that the game would be known as the "Super Bowl of the Century."

2) There should be at least one long edit. Among all your edits, there should be at least one long edit, which rewrites the whole sentence or at least half of the sentence.

In the above example, the long edit is: "on February 7, 2016, at Levi's Stadium in the San Francisco Bay Area at Santa Clara, California." → "Mercedes-Benz Superdome in New Orleans, Louisiana and was the first Super Bowl to be played in the United States."

3) The edits should create contradicting information. After your edits, the original passage and the modified passage should have contradicting information. One way to test it is that: when you ask questions about your modified information, the original passage and the modified passage gives contradicting answers.

For example: after you edit "Denver Broncos" to "San Francisco 49ers", the original and modified passages are shown in the Figure below:

Original Text:

`The American Football Conference (AFC) champion` **`Denver Broncos`** `defeated the National Football Conference (NFC) champion Carolina Panthers 24-10 to earn their third Super Bowl title.`

Modified Text:

`The American Football Conference (AFC) champion` **`San Francisco 49ers`** `defeated the National Football Conference (NFC) champion Carolina Panthers 24-10 to earn their third Super Bowl title.`

When you ask the question: "Which NFL team won Super Bowl 50?", the original passage gives you the answer "Denver Broncos", and the modified passage gives you the answer "San Francisco 49ers". This is a contradiction.

Another example is the following edit: "so that the logo could prominently feature the Arabic numerals 50." → "so that the game would be known as the "Super Bowl of the Century".

Original Text:

`... the league emphasized the "golden anniversary" with various gold-themed initiatives, as well as temporarily suspending the tradition of naming each Super Bowl game with Roman numerals (under which the game would have been known as "Super Bowl L"),` **`so that the logo could prominently feature the Arabic numerals 50.`**

Modified Text:

`... the league emphasized the "golden anniversary" with various gold-themed initiatives, as well as temporarily suspending the tradition of naming each Super Bowl game with Roman numerals (under which the game would have been known as "Super Bowl L"),` **`so that the game would be`**

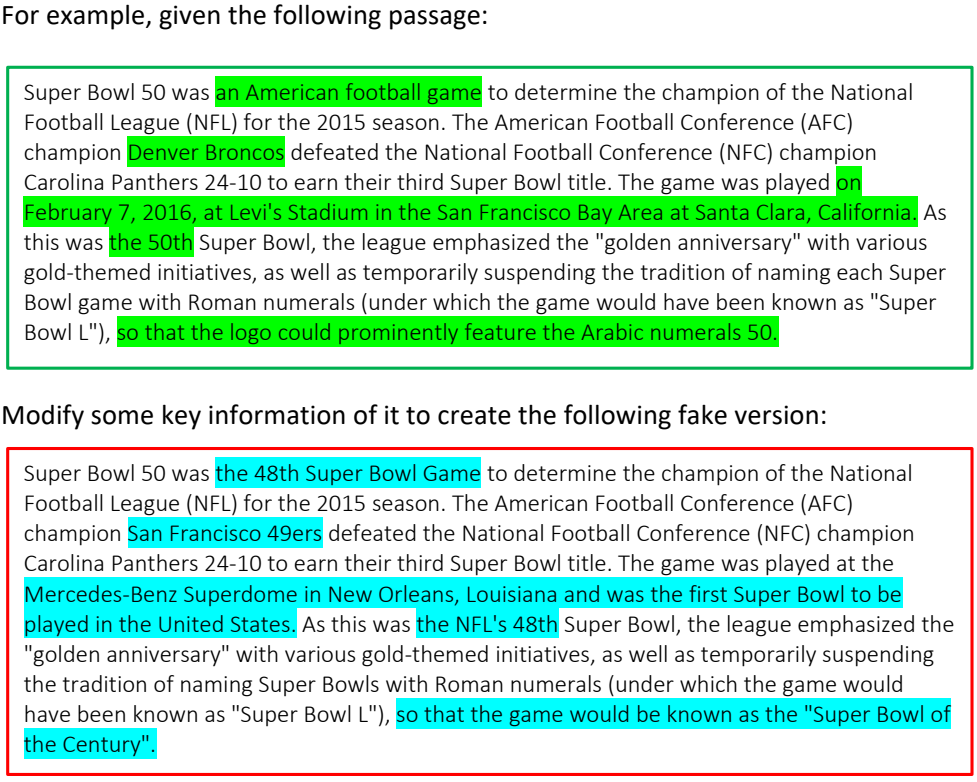

Figure 5: An example of human annotation that follows all instructions of the annotation guideline.

known as the "Super Bowl of the Century".

When you ask the question: "Why the league suspended the tradition of naming Super Bowls with Roman numerals?" the original passage and the modified passage also give you contradicting answers.

However, the following passage does **NOT** create any contradiction, because the modified information is just a paraphrasing of the original information.

Original Text:

The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24-10 to **earn their third Super Bowl title.**

Modified Text:

The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24-10 to **win the Super Bowl.**

4) The edits should modify important information in the passage. Your edits should focus on important information in the passage, *i.e.*, points that people are usually interested in and would usually ask about. For example, time, location, purpose, outcome, reason, etc. Please avoid editing trivial and unimportant details.

For example, the following trivial edit is not supported:

Original Text:

the game would have been known as "Super Bowl **L**"...

Modified Text:

the game would have been known as "Super Bowl **H**"...

5) The modified passage should look "realistic". The final modified passage should look "realistic". Don't make obvious logic or commonsense mistakes to make the reader easily know that this is a fake passage by simply going through it.

For example, the following edit is not supported.

Original Text:

The game was played on February 7, 2016,

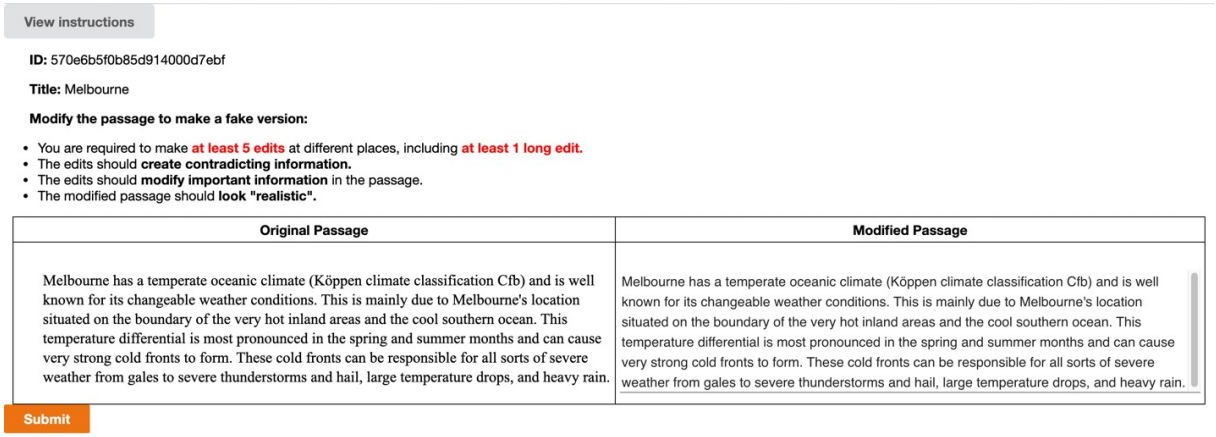

Figure 6: The annotation interface in the Amazon Mechanical Turk.

at Levi's Stadium in the San Francisco Bay Area at **Santa Clara**, California.

Modified Text:

The game was played on February 7, 2016, at Levi's Stadium in the San Francisco Bay Area at **New York City**, California.

People can easily tell the modified passage is fake since everybody knows that New York is not a city in California.

A.3 Annotation Interface

The original passage is shown on the left for your reference, you should modify the passage in the text box on the right to make the fake passage. After you finished the edits, Click "Submit".