Won't Get Fooled Again: Answering Questions with False Premises

Shengding Hu[1], **Yifan Luo**[2], **Huadong Wang**[1]*,
Xingyi Cheng[3], **Zhiyuan Liu**[1,4,5], **Maosong Sun**[1,4,5]
[1]Dept. of Comp. Sci. & Tech., Institute for AI, Tsinghua University, Beijing, China
Beijing National Research Center for Information Science and Technology
[2] School of Cyberspace Security, BUPT [3]Tencent
[4]Institute for Artificial Intelligence, Tsinghua University
[5]International Innovation Center of Tsinghua University, Shanghai, China
hsd20@mails.tsinghua.edu.cn, yifanluo@bupt.edu.cn

Abstract

Pre-trained language models (PLMs) have shown unprecedented potential in various fields, especially as the backbones for question-answering (QA) systems. However, they tend to be easily deceived by tricky questions such as "How many eyes does the sun have?". Such frailties of PLMs often allude to the lack of knowledge within them. In this paper, we find that the PLMs already possess the knowledge required to rebut such questions, and the key is how to activate the knowledge. To systematize this observation, we investigate the PLMs' responses to one kind of tricky questions, i.e., the false premises questions (FPQs). We annotate a FalseQA dataset containing 2365 human-written FPQs, with the corresponding explanations for the false premises and the revised true premise questions. Using FalseQA, we discover that PLMs are capable of discriminating FPQs by fine-tuning on moderate numbers (e.g., 256) of examples. PLMs also generate reasonable explanations for the false premise, which serve as rebuttals. Further replaying a few general questions during training allows PLMs to excel on FPQs and general questions simultaneously. Our work suggests that once the rebuttal ability is stimulated, knowledge inside the PLMs can be effectively utilized to handle FPQs, which incentivizes the research on PLM-based QA systems. The FalseQA dataset and code are available at https://github.com/thunlp/FalseQA.

1 Introduction

Recent advances in pre-trained language models (PLMs) (Lewis et al., 2020; Raffel et al., 2020; Brown et al., 2020; Roller et al., 2021; Han et al., 2021) have achieved significant performance gains for various types of tasks, even surpassing human levels on language ability benchmarks (Wang et al., 2018, 2019; Srivastava et al., 2022). The unprecedented ability of PLMs lays the foundation for various practical applications. For example, PLMs that exhibit general world knowledge and commonsense knowledge have the potential to serve as backbones for general-purpose question-answering models (Tafjord and Clark, 2021; Guu et al., 2020).

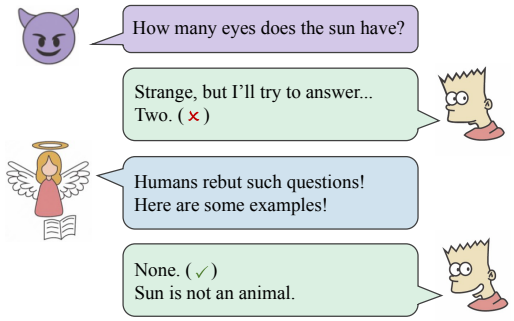

Figure 1: The rebuttal ability of PLMs can be activated by human rebuttal examples.

Question	Answer
Sally's favorite cow died yesterday. When will the cow be alive again? [1]	in a few days.
How many eyes does the sun have? [2]	The sun has one eye.

Table 1: Some previous examples that report the vulnerability of PLMs to tricky questions. More examples provided by this work are in Table 2.

However, these PLM-based question-answering models have an intriguing paradox. On the one hand, they achieve high performance on normal questions raised by humans. For example, UnifiedQA (Khashabi et al., 2020) achieves state-of-the-art performance on many question-answering tasks. Macaw (Tafjord and Clark, 2021) can perform multi-angle question-answering and answer 75% of the question in the Challenge300 dataset (Tafjord and Clark, 2021) correctly. On the other hand, they are vulnerable to tricky questions (see Table 1). For example, Macaw answers one

* Corresponding author: Huadong Wang (huadw2012@163.com)

[1]AllenAI's Blog.
[2]Blog *Giving GPT-3 a Turing Test*.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 5626–5643
July 9-14, 2023

out of nine tricky questions correctly, while other models including GPT-3 (Brown et al., 2020) fail all of them (Tafjord and Clark, 2021). InstructGPT (Ouyang et al., 2022) also reports that it fails to identify instructions with false premises. These questions are easy to rebut for humans but pose an undeniable obstacle for PLMs[3]. The inability to rebut also results in the misalignment (Kenton et al., 2021) of language models to human expectations.

Without careful investigation, this paradox could easily lead to the conclusion that PLMs lack the world or commonsense knowledge to rebut these questions. Although it's crucial for the PLMs to embed as much general knowledge as possible, we provide a pilot experiment to find out that the PLMs already possess the knowledge required for the tricky questions which they fail (see Section 3.2). As a consequence, we hypothesize that the knowledge in current PLMs is *enough* for handling a large portion of tricky questions. However, this knowledge is *not activated*.

To support our hypothesis, we take a close look at these tricky questions. Most of these tricky questions contain false premises. For example, in the question "*How many eyes does the sun have?*", the questioner must presume that "*the sun can have eyes*" in order to make the query about the quantity meaningful. These questions are called False Premise Questions (FPQs). Such false premises always violate human knowledge or logic and rarely appear in the natural text, thus leading to an out-of-distribution generalization gap for the PLMs.

Targeting to fill the gap between the natural text and FPQs, we present the first specialized dataset of FPQs, dubbed as FalseQA dataset. Specifically, we first systematically categorize the false premises to ensure the coverage of the dataset. Then we ask human annotators to manually compose the FPQs, as well as explanations for the false premises. The annotators are also asked to edit the false premise questions into true premise questions (TPQs) using minimal modification, with which the PLMs are less prone to learn shortcuts from the format of FPQs.

Based on FalseQA dataset, we first conduct systematic experiments on the PLMs' discrimination and rebuttal ability of FPQs. We reach three essential conclusions: (1) PLMs of different types and scales can distinguish the FPQs from TPQs, and scaling effect (Kaplan et al., 2020) also holds for FalseQA. (2) PLMs can give reasonable explanations for the false premises, which can serve as rebuttals. (3) The number of FPQ examples needed to activate the PLM's rebuttal ability is moderate. For example, 256 FPQs can result in more than 70% accuracy for models larger than 1B. And for some larger PLMs, in-context learning with a few examples can also activate the ability. Then we consider the practical scenario where the models need to handle both FPQs and general questions. We demonstrate that a simple but effective data replay method can help mitigate the catastrophic forgetting of general questions, where the model discriminates 86.7% FPQs in FalseQA and only rebuts 1.4% general questions. These results lead to optimism that PLMs can be used as the backbones of a practical question-answering system that is robust to tricky questions.

2 Related Work

Three groups of research are related to our work: direct question answering dataset, question unanswerability, and question premise verification.

Direct Question Answering Dataset. For the most practical scenario of the question-answering system, the candidate answers are absent. Therefore, direct question answering (DQA), as a counterpart to extractive QA (Rajpurkar et al., 2016) or multiple-choice QA (Lai et al., 2017) has received increased attention. Natural Questions (Kwiatkowski et al., 2019) collects the queries sent to the google search engine. ARC-DA (Bhakthavatsalam et al., 2021) proposes modifying a reasoning-based multiple choice QA into DQA format. Tafjord and Clark (2021) manually compose Challenge300 dataset which is still challenging to powerful models such as GPT-3 and MACAW. Our dataset can be seen as a direct question-answering dataset with explanations. However, the question distribution is radically different from the questions in natural corpora, serving as an adversarial scenario for DQA models.

Question Unanswerability. Tricky questions are unanswerable questions. Previous works (Raina and Gales, 2022; Rajpurkar et al., 2018; Asai and Choi, 2021; Davis, 2020) confirm the existence of unanswerable questions in existing benchmarks, in-

[3]Although most PLMs fail, we found ChatGPT (OpenAI, 2022) satisfactorily answers these questions. Their training data is manually written by annotators and continuously updated using user queries, which might contain such questions. However, their data is not public. Our work provides the same possibility for general PLMs, even the much smaller ones.

Model	# Para	Reply	Correctness	Ablation
Are tigers' eggs bigger than chicken's eggs?				Do tigers lay eggs?
Bloom	176B	No, they are not. The eggs of tigers are smaller than ...	✗	No tigers lay eggs.
OPT	175B	Yes.	✗	No, they don't. Tigers are mammals.
Jurassic-1	178B	No, tigers' eggs are the same size as chicken's eggs.	✗	Tigers give birth to live young.
GPT-3	175B	Yes, tiger eggs are larger than chicken eggs. Tiger eggs can ...	✗	No, tigers do not lay eggs ...
MACAW	11B	Yes.	✗	Yes.
MACAW-FPQ	11B	Tigers don't lay eggs, chickens do.	✓	Tigers are carnivores and do not lay eggs.
Which Chinese classic do you like best, Faustus or Hamlet?				Are Faustus and Hamlet Chinese classics?
Bloom	176B	I like Faustus best. I like Hamlet, but I like Faustus best.	✗	No, they are not. They are Western ...
OPT	175B	I like both.	✗	No.
Jurassic-1	178B	I like Faustus best. Faustus is a man, and Hamlet is a ghost.	✗	No, they are not.
GPT-3	175B	I personally prefer Faustus, as it is a classic Chinese play ...	✗	No, Faustus and Hamlet are not ...
MACAW	11B	Faustus.	✗	No.
MACAW-FPQ	11B	Faustus and Hamlet are European classics.	✓	Faustus and Hamlet are European classics.

Table 2: A case study of PLMs' responses to FPQs and the questions that ask the correctness of the premises. The questions are from the test split of FalseQA dataset. See Appendix C.1 for more examples.

cluding SQuAD (Rajpurkar et al., 2016), Natural Questions (Kwiatkowski et al., 2019), VQA (Antol et al., 2015), etc. Most unanswerable questions in these benchmarks are due to missing information in the context provided to the questions. However, FalseQA contains questions that are out of natural text distribution, and are unanswerable due to misleading false premises.

Question Premise Verification. Answering FPQs has been studied before the deep learning era (Kaplan, 1978). In recent PLM-based question-answering research, relevant efforts use external knowledge to verify the correctness of the question premise. For example, Kim et al. (2021) studies the FPQs in Natural Questions (Kwiatkowski et al., 2019). A concurrent work (Min et al., 2022) further gathers the 8400 Reddit questions and annotated the false premises among 25% of them. The correctness of the premises in their datasets requires expert knowledge or context to determine. Therefore, they use retrieval-augmented language models (Krishna et al., 2021) or external knowledge base to provide information for the premise classification, and both reach the conclusion that discovering and explaining those prepositions that require expert knowledge is challenging. However, it remains elusive whether PLMs without external assistance can discover and rebut the tricky questions that require only general knowledge and are straightforward for humans. We propose the first manually written dataset for FPQs and support our hypothesis through experiments that the inability of PLMs for FPQs can be mitigated when giving them examples.

3 Preliminaries

In this section, we introduce the definition of FPQ and the pilot experiment on PLMs about FPQs.

3.1 False Premise Questions

When questioning, humans usually assume that some facts are shared and endorsed by the questioner and the answerer. Such facts are the premises of the question. For example, in the question "*How many eyes does the sun have?*", the target of the question is the number of eyes, which assumes the correctness of the fact "*The sun has eyes*".

In general, a fact can be expressed by relational triples, where each relational triple takes the form of `<subject, predicate, object>`. A question is asking for the missing part in one relational triple. For example, the above question can be expressed as nested triples as `<triple, quantity, ?>`, where `triple = <sun, has_property, eye>`. We define the complete relational triple as the support triple. Then a false premise problem is one whose support triples are not correct. In the above example, `<sun, has_property, eye>` is false under real-world background, thus any question that builds on this triple contains false premises. By this definition, "Does the sun have eyes?" is not an FPQ, since it does not assume `<sun, has_property, eye>` to be true. In fact, PLMs know the authenticity of such triples well. However, they can't answer FPQs built upon these triples.

3.2 PLMs' original responses to FPQs

We begin with a pilot experiment that confirms current PLMs' responses to FPQs are not satisfactory despite their knowledge. We query the PLMs with the questions taken from FalseQA test split (see

Category	Fraction (%)	Description	Example
Error Types			
Property	23.2	The entity does not has the property.	How long has the Sun been transparent?
Action	19.7	The entity can not perform the action.	How far can a fish walk on the street?
Scope	19.6	A fact is not valid in the scope.	Who is the villain who fought Harry Potter in A Song of Ice and Fire?
Entity	11.3	The entity can not exist.	What's the most common color of human's wings?
Event	8.3	The event didn't happen in the history.	When did Zuckerberg start Google?
Logic	6.7	Contain logically conflicting statements.	How to sit down while walking?
Causality	5.6	Does not follow causality.	Why the more water you drink, the more thirsty you are?
Index	4.6	The specified index is out of an entity list.	What is the 50th largest province in China?
Question Formats			
Descriptive	29.6	The question needs descriptive answer.	Why carbon dioxide is composed of oxygen?
Factual	28.1	The question seeks factual information.	When did China become a member of the EU?
Enumerative	12.3	The answer is a list of items.	List three vegetables that tigers feed on.
Selective	10.7	The answer candidates are provided.	Which one is the right behave in the theatre? Fight or disrupt the show?
Hypothetical	9.0	The question contains a conditional clause.	When should I go if I want to see the fog on Jupiter?
Affirmative	8.5	The question requires a yes-or-no answer.	Do people eat diamond because it comes with mutiple nutrition?

Table 3: The categorization and examples of FPQ questions. We omit the "Other" category in this table.

Section 4). We use the large PLMs whose API is publicly available, including Bloom (Scao et al., 2022), OPT (Zhang et al., 2022), Jurassic-1 (Lieber et al., 2021), GPT-3(text-davinci-003) (Brown et al., 2020) (as known as InstructGPT). We use the prompt "*Question:* ___ *Answer:*", where the blank is filled by the question text. We provide the generated answers of these models in Table 2. We also provide our model's answer (See Section 5) as comparisons. As we can see, all models fail on these simple FPQs. However, in the column "Ablation", we are surprised to find that all models give the correct responses to the questions that ask directly about the correctness of the premises. This motivates us to hypothesize that the inability of current PLMs to handle FPQs is due to distribution mismatch, instead of missing knowledge. Therefore, we need a dataset specializing in FPQs.

4 Dataset

To build a dataset on FPQs, there are potentially two approaches. An approach is to collect them from natural corpora. However, false premise questions rarely appear in natural corpora, which makes the question collection process laborious. Second, even if we collect false premise questions, the false premises are made by humans and thus are hard to be detected by humans, which doesn't fit with the motivation of this paper. In fact, Min et al. (2022) have done pioneering work using this approach. On the contrary, our approach is to manually write such false premise questions. To ensure the quality of our dataset, we expect FalseQA dataset to have the following key features: *broad coverage*, *high quality*, *few shortcuts*, and *detailed explanations* for the false premises. Below we introduce the annotation steps that ensure these features.

4.1 Categorization of FPQs.

People ask questions in a wide variety of contexts and formats. Increasing the coverage of questions is proven to be beneficial (Khashabi et al., 2020). However, asking annotators to write FPQs freely does not guarantee the coverage of the questions. Therefore, the authors manually think up 29 initial FPQs (see Appendix A.1). Then we categorize these FPQs in terms of error types, and question format. We summarize the categories in Table 3. In total, there are eight error types covering commonsense errors, logical errors, etc., and six question formats covering factual questions, descriptive questions, etc. Although we try to collect as many examples as possible into the initial set, the categorizations are far from exhaustive. Therefore we include an "Others" option to encourage creativity.

Writing FPQs. We recruit twenty human annotators to think up questions that contain false premises. To make the creative process easier, we provide source words to the annotators to compose sentences. We use the subject word of GenericsKB (Bhakthavatsalam et al., 2020) as the source word since they have broad coverage and each

word is paired with a short illustrative sentence that can also inspire the annotators. However, we don't require the annotated sentence to contain the source word. Moreover, the annotators have the freedom to skip the source words that are not easy to brainstorm. We then ask the annotators to categorize the questions into the above categories. The annotators are required to keep a balanced distribution (see Appendix A.2) over categories when they finish their part. For the quality of the written FPQs, we require them to be correct in syntax and contain obvious false premises.

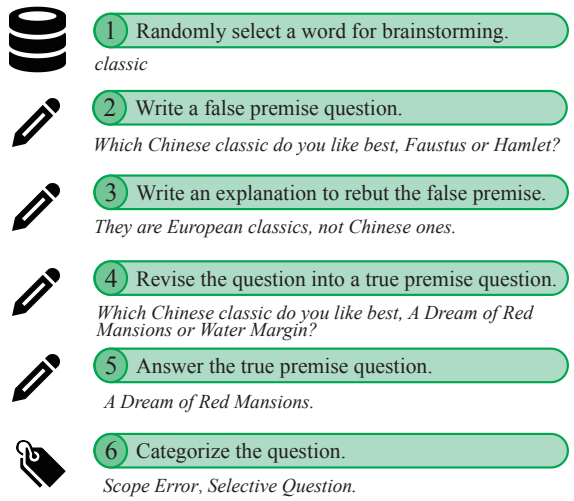

Figure 2: The annotation process of FalseQA. The *italic* sentences are one annotation example.

Revising into TPQs. Previous studies (Du et al., 2021) point out that PLMs are skilled at finding shortcuts in datasets and do not really understand the task. Since the FPQs are created manually, it's easy to fall into the fixed writing style of the annotators. To alleviate the problem, we annotate a comparison set for these FPQs. Specifically, we ask annotators to edit each FPQ with minimal modifications to make it a problem with true premises (TPQ). The resulting pairs of questions differ only in the correctness of the premises, ensuring that the model learns the essentials of the task.

Writing Detailed Explanations/Answers. Humans usually reply to FPQs with an explanation of why the premise is false (Kaplan, 1978). Generating the explanation also helps check whether the model truly understands the FPQs. Therefore, we ask the annotators to write an explanation for each FPQ. For quality control of the explanations, we require the explanation to be more than the negation of the false premise. For the training set and validation set, we require one explanation per question, for the test set, we require two explanations per question. For symmetry, the annotators also write answers to the TPQs. The full annotating process is demonstrated in Figure 2.

4.2 Dataset Statistics

The final dataset, dubbed as FalseQA, contains 2365 question pairs. A snapshot of the FPQ dataset is in Table 5. We randomly split the dataset into train, validation, and test splits, with a ratio of 5:2:3. The summary of statistics is shown in Table 4.

Number of annotators	20
Number error types (FPQs)	8
Number question format (FPQs)	6
Average question length (FPQs)	10.6 tokens
Average explanation length (FPQs)	12.1 tokens
Average question length (TPQs)	10.4 tokens
Average answer length (TPQs)	9.8 tokens
Training set	1187 question pairs
Validation set	491 question pairs
Test set	687 question pairs

Table 4: Statistics of FalseQA dataset. The number of tokens is calculated by NLTK (Bird and Loper, 2004).

5 Experiments

Our experiments are divided into two main parts. To begin with, we conducted extensive experiments to demonstrate that PLMs have the ability to discriminate and rebut FPQs with moderate training data. Next, we propose a practical method to handle both FPQs and general questions well.

5.1 Models and Settings

PLMs are usually divided into three main architectures, namely, encoder-only, decoder-only, and encoder-decoder language models. Since the encoder-only language model can not be used as the QA model, we select typical PLMs from the latter two for experiments.

For decoder-only models, we choose OPT (Zhang et al., 2022), which is a series of open-source pre-trained models aligned to OpenAI GPT-3 (Brown et al., 2020). For the encoder-decoder models, we use T5 (Raffel et al., 2020) and MACAW (Tafjord and Clark, 2021). T5 (Raffel et al., 2020) models are trained with the massive unsupervised pre-training corpus and a mixture of supervised tasks, making them very capable of solving various downstream tasks. MACAW is fine-tuned from T5 models on QA tasks. They achieve state-of-the-art performance on direct QA dataset ARC-DA (Bhakthavatsalam et al., 2021) and perform satisfactorily on most categories

Source Word	Type	Question	Explanation/Answer
tennis	FPQ	What was the place where the tennis match was launched in the 1200s?	Modern tennis had not been invented in the 12th century.
	TPQ	What was the place where the French Open was held in 2021?	The 2021 French Open was held in Roland Garros from May to June.
software	FPQ	List a software that is developed by Edison.	Edison was a physics inventor, not a computer scientist.
	TPQ	List a software that is developed by Bill Gates.	Windows xp.

Table 5: Example question pairs (FPQ and TPQ) and their source words, explanations/answers.

Model	Recall	Precision	Accuracy
OPT-350M	64.8 ± 7.2	65.5 ± 3.3	65.1 ± 1.8
OPT-1.3B	67.4 ± 7.6	73.5 ± 5.1	71.2 ± 0.4
OPT-2.7B	69.2 ± 12.2	76.7 ± 5.0	73.7 ± 2.1
T5-Large	72.8 ± 2.3	76.9 ± 1.5	75.4 ± 0.3
T5-3B	80.6 ± 7.7	83.8 ± 4.3	82.3 ± 1.9
T5-11B	86.5 ± 1.7	82.4 ± 1.0	84.0 ± 1.1
MACAW-Large	75.0 ± 4.1	77.9 ± 3.3	76.7 ± 0.7
MACAW-3B	79.9 ± 6.8	85.0 ± 5.3	82.6 ± 0.5
MACAW-11B	86.0 ± 2.1	87.0 ± 0.7	86.6 ± 1.3

Table 6: The recall and precision are for discriminating FPQs, and the accuracy of binary classification.

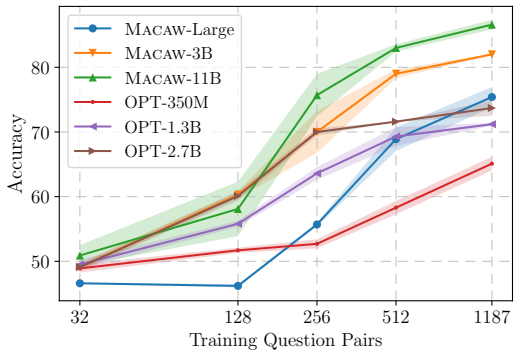

Figure 3: PLMs discrimination ability to FPQs from TPQs with the number of training samples.

of the demanding dataset Challenge300 (Tafjord and Clark, 2021) except for the FPQs.

Unless specified, all experiments are repeated three times with different random seeds. For each result, we report the mean and standard deviation. The detailed hyperparameters for each experiment are in Appendix B.

5.2 Discriminating FPQs

We first train the PLMs to classify the question in FalseQA into FPQ and TPQ. To mitigate the gap between pre-training and fine-tuning, we adopt the prompt learning paradigm (Schick and Schütze, 2021; Liu et al., 2023) to do the classification. We report the accuracy of the classification. Besides, we report the recall and precision for FPQs since we emphasize the FPQs.

From Table 6, we can see all the models can achieve non-trivial performance on the binary classification. (1) The most powerful model MACAW-11B, can achieve 86.6 accuracy. (2) Across all the models of the same type, performance boosts when the size of the model increases. We hypothesize that the scaling effect is because larger models both contain more knowledge and are easier to be activated to understand the task. (3) There is a slight improvement from T5 to MACAW, showing that the ability to identify FPQs can be enhanced by fine-tuning on a corpus of normal questions.

5.3 Impact of Training Data Size

Then we study the PLMs' performance to discriminate FPQs with fewer training data. We randomly sample 32, 128, 256, and 512 pairs of FPQ and TPQs as the training data and plot the performance under each data scale in Figure 3. We can see that the accuracy of classifying FPQs and TPQs grows almost linearly as the number of pairs grows exponentially. With only 256 pairs of questions, models larger than 2.7B, i.e., OPT-2.7B, MACAW-3B, MACAW-11B, all achieve more than 70% accuracy, while the smaller models need more data to achieve non-trivial performance. The trade-off between model scale and data scale hints that larger models might be activated with even fewer training data. However, as we have noticed, the gap between human performance and model performance remains large, as an average person can almost completely classify such problems.

The above results already allow us to design a primitive QA pipeline that can handle FPQs. For example, if the model predicts that a question is FPQ, then it refuses to answer such questions, while for other questions it generates the answer.

5.4 Answering FPQs with Explanations

Next, we train the PLMs to discriminate and generate explanations for the FPQs at the same time. Since we need to start from models that already have zero-shot QA ability, we choose only MACAW for the encoder-decoder models. For the decoder-only model, we follow similar approaches to Tafjord and Clark (2021) to train OPT models with a fraction of UnifiedQA dataset (Khashabi

# QP	Model	Recall	Accuracy	ROUGE-L
32	OPT-2.7B	62.4 ± 14.0	52.8 ± 0.7	27.7 ± 1.9
	+Binary Loss	59.0 ± 5.3	56.3 ± 1.2	27.0 ± 1.6
	MACAW-3B	41.9 ± 22.3	56.8 ± 3.4	29.1 ± 3.0
	+Binary Loss	40.5 ± 21.8	61.5 ± 7.7	32.0 ± 1.3
	MACAW-11B	64.5 ± 36.9	59.2 ± 9.0	36.2 ± 5.2
	+Binary Loss	49.0 ± 19.6	64.1 ± 7.2	33.8 ± 0.5
256	OPT-2.7B	56.8 ± 5.3	56.9 ± 2.0	29.5 ± 0.4
	+Binary Loss	62.5 ± 5.5	67.8 ± 1.6	29.7 ± 0.5
	MACAW-3B	69.5 ± 7.5	73.5 ± 1.7	34.5 ± 1.3
	+Binary Loss	72.6 ± 8.7	76.5 ± 2.3	35.3 ± 1.5
	MACAW-11B	77.3 ± 13.0	76.2 ± 1.9	35.0 ± 2.0
	+Binary Loss	81.3 ± 4.6	79.2 ± 0.2	38.4 ± 0.7
1187	OPT-2.7B	76.2 ± 4.1	70.8 ± 0.9	34.2 ± 0.6
	+Binary Loss	75.9 ± 4.9	75.3 ± 0.5	34.0 ± 1.1
	MACAW-3B	81.8 ± 7.3	80.6 ± 1.2	39.2 ± 1.9
	+Binary Loss	80.9 ± 1.2	84.2 ± 0.7	38.1 ± 1.0
	MACAW-11B	90.7 ± 5.2	83.6 ± 0.8	41.9 ± 0.6
	+Binary Loss	88.8 ± 1.8	87.1 ± 0.9	42.0 ± 0.7

Table 7: Joint FPQ discrimination and explanation generation. Better results are shown in green.

et al., 2020) in order to steer the model into QA mode [4] without injecting much additional knowledge. We select the model size that can achieve non-trivial performance using 256 pairs of data for this experiment.

To discriminate and generate explanations jointly, we let the models generate the discriminating tokens: "*tricky question*" or "*true question*" first. Then the model continues to generate the explanation to FPQs or the answer to TPQs. Since the numbers of tokens responsible for discrimination and generation differ dramatically, we add an additional binary loss on the discriminating tokens. The ratio between the binary loss and the generation loss is 1. We conduct experiments on three training data sizes, i.e, 32, 256, and 1187 question pairs.

In evaluation, if a generated answer contains "*tricky question*", we consider the question classified as an FPQ, otherwise, it is classified as a TPQ. Similar to the previous section, we report the recall, precision of predicting FPQs, and accuracy of the binary classification. In addition, we evaluate the quality of the generated explanation by computing the maximum ROUGE-L (Lin, 2004) score between it and the two ground-truth explanations. Note since we focus on the explanation of FPQs, the evaluation does not include the TPQs.

From Table 7, we have three observations. (1) The models jointly predict the question and generate answers successfully. (2) When training data is limited, e.g., 32 question pairs, the accuracy is significantly higher than conducting classification alone (See in Figure 3), which shows that the explanations of the FPQs help the model to quickly adapt to the task. (3) Adding binary loss boosts the model's performance on classification. For the generated explanations, the best ROUGE-L achieves 42.0, showing that the explanations are close to humans'. The quality of explanations also gets higher as the model size and data size increase. We provide the model-generated explanation for 10 randomly sampled FPQs in Appendix C.2. We can see the explanations are reasonable.

# QP	Model	Recall	Accuracy	ROUGE-L
0	OPT-66B	6.8	25.8	12.2
	Jurassic-1	66.2	36.5	6.5
	GPT-3(001)	46.9	46.1	5.1
	GPT-3(002)	98.5	53.2	25.3
2	OPT-66B	21.3 ± 18.5	53.0 ± 2.6	32.2 ± 2.8
	Jurassic-1	52.8 ± 37.0	56.9 ± 2.6	32.4 ± 5.3
	GPT-3(001)	43.6 ± 16.7	63.9 ± 4.1	31.8 ± 2.7
	GPT-3(002)	87.9 ± 2.4	75.2 ± 1.6	38.1 ± 1.5
4	OPT-66B	19.7 ± 29.8	51.9 ± 3.7	34.8 ± 1.4
	Jurassic-1	94.7 ± 8.2	53.1 ± 4.8	38.4 ± 0.7
	GPT-3(001)	61.9 ± 15.7	67.6 ± 1.5	34.5 ± 1.2
	GPT-3(002)	90.6 ± 4.6	75.8 ± 2.9	39.1 ± 1.6

Table 8: Performance of in-context learning under different numbers of examples. Better results are in green.

5.5 In-context Learning

We proceed to study the performance of larger models, e.g., GPT-3(175B) on FalseQA. The large PLMs are tuned by in-context learning with frozen model parameters. We select OPT-66B (Zhang et al., 2022), Jurassic-1 (Lieber et al., 2021), and GPT-3(001) and GPT-3(002) [5]. We present the results in Table 8. We can see that OPT-66B and Jurassic-1 perform poorly. Therefore, we conclude that due to the distribution mismatch of FPQs to normal questions, it is still hard to activate the rebuttal ability using a few examples for these models, which we leave to future work. GPT-3 can be activated with 2 or 4 pairs of examples, however, its performance is lower than the much smaller fine-tuned models in Section 5.4. Surprisingly, GPT-3(002) has far better performance than GPT-3(001). We hypothesize that they more easily understand the rebuttal task since they are trained with instruction tuning (Ouyang et al., 2022).

5.6 Performance w.r.t. Category

To better understand which kind of FPQs is harder to be discriminated against, we draw the accuracy of each category in Figure 4. In spite of the inconsistency between PLMs, index error is generally

[4] We will release the checkpoint.

[5] text-davinci-001, and text-davinci-002 checkpoints.

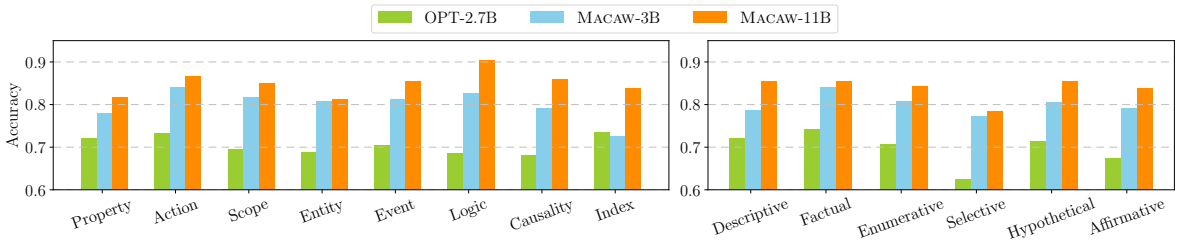

Figure 4: PLMs' accuracy scores for different error types (left) and question formats (right).

Settings	FalseQA				ARC-DA		
	Recall	Precision	Accuracy	ROUGE-L	FPR(↓)	ROUGE-L	F1
Raw MACAW-11B	8.7 ± 2.5	91.5 ± 7.8	53.8 ± 0.8	7.2 ± 0.0	0.0 ± 0.0	54.5 ± 0.0	55.0 ± 0.0
+ FPQ (256 shots)	81.3 ± 4.6	78.2 ± 2.2	79.2 ± 0.2	38.4 ± 0.7	23.9 ± 13.6	24.2 ± 1.5	23.9 ± 1.6
+ Data Replay	72.1 ± 7.0	81.4 ± 0.9	77.9 ± 3.1	35.1 ± 1.0	1.8 ± 0.9	30.6 ± 2.9	30.4 ± 3.0
+ FPQ (Full)	88.8 ± 1.8	85.9 ± 2.7	87.1 ± 0.9	42.0 ± 0.7	12.6 ± 6.6	32.2 ± 2.4	32.3 ± 2.5
+ Data Replay	85.6 ± 1.3	87.5 ± 0.5	86.7 ± 0.5	39.2 ± 0.8	1.4 ± 0.0	48.6 ± 1.4	49.1 ± 1.2
Raw OPT-2.7B	5.0 ± 2.0	54.5 ± 14.8	50.5 ± 1.3	7.3 ± 0.0	0.1 ± 0.0	39.4 ± 0.0	39.0 ± 0.0
+ FPQ (256 shots)	62.5 ± 5.5	70.0 ± 1.9	67.8 ± 1.6	29.7 ± 0.5	19.9 ± 3.8	25.0 ± 0.2	23.9 ± 0.3
+ Data Replay	64.0 ± 2.8	69.4 ± 1.0	67.9 ± 0.4	29.1 ± 1.3	1.8 ± 0.8	33.8 ± 0.7	33.1 ± 0.9
+ FPQ (Full)	75.9 ± 4.9	75.2 ± 3.0	75.3 ± 0.5	34.0 ± 1.1	33.2 ± 6.0	22.0 ± 0.8	20.8 ± 0.9
+ Data Replay	76.8 ± 2.5	74.2 ± 1.2	75.0 ± 0.4	33.2 ± 0.5	3.5 ± 0.3	35.8 ± 0.9	35.3 ± 1.1

Table 9: Results after tuning with FalseQA data and data replay techniques. Better results are shown in green.

hard to classify while logic and causality error is easy. For question types, selective questions are hard to classify while factual questions are easy. These observations can guide the future improvement of our dataset.

5.7 Answering FPQs and General Questions

QA models are originally used to answer general questions, e.g., questions in ARC-DA (Bhakthavatsalam et al., 2021) [6] dataset where the distribution is different from FalseQA. Therefore, training purely on FalseQA may lead to catastrophic forgetting. To produce a model that handles both FPQs and general questions, we use a simple data replay technique (DR) (Chaudhry et al., 2019). Specifically, during training on FalseQA dataset, for each iteration over batches, we add a batch of the data samples from the ARC-DA. In order to use as little ARC-DA data as possible, we keep the ARC-DA samples to be the same within 30 batch iterations. The aforementioned binary loss is used no matter with or without DR. The concrete numbers of general questions used in each setting and training details are in Appendix B.5.

In Table 9, we summarize the performance of the raw model before training on FalseQA, the model tuned on FalseQA, and the model tuned on FalseQA with DR. For the original models, since they do not generate the "*tricky question*" or "*true question*", we manually read the generated answers for 100 randomly sampled questions pairs to determine whether it contains any rebuttals. As we can see, before fine-tuning on FPQs, the models perform well on the ARC-DA dataset. However, they fail substantially on FalseQA. After tuning on FalseQA, though the models' rebuttal ability is activated, ROUGE-L and F1 scores on ARC-DA drop considerably. The false prediction rate (FPR), i.e., the fraction of ARC-DA questions that are incorrectly labeled as tricky questions, is non-negligible. Fortunately, when we apply the DR technique, models not only have small FPRs and the improved quality of generated answers on ARC-DA but the same or even better performance on FalseQA. We also find the questions in ARC-DA that PLM still rebuts (see Appendix C.3) are also reasonable to rebut for humans. The result gives us a promising direction for building QA systems that perform well on general questions and FPQs.

6 Conclusion

In this paper, we investigate using PLMs to answer FPQs, which are simple for humans but deceive most PLMs. We present the first human-written dataset of FPQs. Using the dataset, we successfully activate the discrimination and explanation ability of PLMs and produce PLMs that are both capable of general questions and robust to FPQs. For future directions, we think that more advanced techniques

[6] Short for AI2 Reasoning Challenge-Direct Answer.

can be used together with FalseQA to fully activate the model's ability, e.g., reinforcement learning with human feedbacks (Ouyang et al., 2022). Incorporating more knowledge into PLMs is also beneficial for PLMs to answer FPQs.

Acknowledgement

This work is supported by the National Key R&D Program of China (No. 2020AAA0106502) and Institute Guo Qiang at Tsinghua University.

Limitations

There are several limitations in our work. (1) Although we think that PLMs' rebuttal ability is activated in our experiments, the performance has a large space for improvement. For a binary classification problem, the most powerful PLM in our experiment reaches 87.1% accuracy at most. (2) Since it's hard to probe what the PLMs *truly know*, we didn't further investigate whether PLMs still fail on some FPQs due to a lack of relevant knowledge or other reasons. (3) A third limitation is that we notice that the newly announced model ChatGPT (OpenAI, 2022) handles such questions satisfactorily. However, since their training data and details are not open-sourced, we are unable to investigate how the ability of these particular models is activated. (4) In this paper, we standardize the expected responses to FPQs as rebuttals, which takes a conventional perspective. However, sometimes we can react with a more creative response, such as a rhetorical question. This can be future work.

Ethical Statement

In the construction of the dataset, we forbid the annotators to compose any sentence that is offensive, harmful, or contains personal information. The annotated data is manually checked to ensure safety. We pay our annotators a competitive salary relative to market rates. The annotated dataset is helpful to encourage models "think" before they provide a response, thus being safer in practical deployment.

References

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. VQA: visual question answering. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 2425–2433. IEEE Computer Society.

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

Sumithra Bhakthavatsalam, Chloe Anastasiades, and Peter Clark. 2020. Genericskb: A knowledge base of generic statements. *ArXiv preprint*, abs/2005.00660.

Sumithra Bhakthavatsalam, Daniel Khashabi, Tushar Khot, Bhavana Dalvi Mishra, Kyle Richardson, Ashish Sabharwal, Carissa Schoenick, Oyvind Tafjord, and Peter Clark. 2021. Think you have solved direct-answer question answering? try arc-da, the direct-answer ai2 reasoning challenge. *ArXiv preprint*, abs/2102.03315.

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

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 *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. 2019. On tiny episodic memories in continual learning. *ArXiv preprint*, abs/1902.10486.

Ernest Davis. 2020. Unanswerable questions about images and texts. *Frontiers in Artificial Intelligence*, 3:51.

Mengnan Du, Varun Manjunatha, Rajiv Jain, Ruchi Deshpande, Franck Dernoncourt, Jiuxiang Gu, Tong Sun, and Xia Hu. 2021. Towards interpreting and mitigating shortcut learning behavior of NLU models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 915–929, Online. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Retrieval augmented

language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.

Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Yuan Yao, Ao Zhang, Liang Zhang, et al. 2021. Pre-trained models: Past, present and future. *AI Open*, 2:225–250.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *ArXiv preprint*, abs/2001.08361.

S. Jerrold Kaplan. 1978. Indirect responses to loaded questions. *American Journal of Computational Linguistics*, pages 21–28. Microfiche 80.

Zachary Kenton, Tom Everitt, Laura Weidinger, Iason Gabriel, Vladimir Mikulik, and Geoffrey Irving. 2021. Alignment of language agents. *ArXiv preprint*, abs/2103.14659.

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

Najoung Kim, Ellie Pavlick, Burcu Karagol Ayan, and Deepak Ramachandran. 2021. Which linguist invented the lightbulb? presupposition verification for question-answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3932–3945, Online. Association for Computational Linguistics.

Kalpesh Krishna, Aurko Roy, and Mohit Iyyer. 2021. Hurdles to progress in long-form question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4940–4957, Online. Association for Computational Linguistics.

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

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

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*, pages 7871–7880, Online. Association for Computational Linguistics.

Opher Lieber, Or Sharir, Barak Lenz, and Yoav Shoham. 2021. Jurassic-1: Technical details and evaluation. *White Paper. AI21 Labs*, 1.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9).

Sewon Min, Luke Zettlemoyer, Hannaneh Hajishirzi, et al. 2022. Crepe: Open-domain question answering with false presuppositions. *arXiv e-prints*, pages arXiv–2211.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *ArXiv preprint*, abs/2203.02155.

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

Vatsal Raina and Mark Gales. 2022. Answer uncertainty and unanswerability in multiple-choice machine reading comprehension. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1020–1034, Dublin, Ireland. Association for Computational Linguistics.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325, Online. Association for Computational Linguistics.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *ArXiv preprint*, abs/2211.05100.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *ArXiv preprint*, abs/2206.04615.

Oyvind Tafjord and Peter Clark. 2021. General-purpose question-answering with macaw. *ArXiv preprint*, abs/2109.02593.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3261–3275.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *ArXiv preprint*, abs/2205.01068.

Appendices

A Annotation Details

A.1 Initial FPQs

We provide the annotators with 29 FPQs in the annotation guide. These questions are original references provided for annotators to brainstorm questions. We list the questions and their error types in Table 10. We didn't provide FPQs for each question format since the question format is much easier to determine without examples.

Error Type	Question
Entity	What color is the sun's eye? Who was the founding president of Qing Dynasty? What color are people's feathers? Are the bananas on the apple tree delicious? Is hydrogen in oxygen combustible?
Index	What is the 50th largest province in China? What day is the eighth day of a week?
Action	Why can't the car be parked in the parking lot? When did we cure AIDS? When did man go to Uranus? How does a cat use a computer? How high can a dog climb a tree? How far a fish can walk? How do pupils go to school with their wives? How to pry open the walnut plasticine?
Property	How long does it take for the sun to become transparent?
Scope	How do I take the train at the airport? What kind of turtle is a fish? What causes Oda Nobunaga's death in the Odyssey? Who caused Guan Yu's death in Water Margin?
Causality	Why the more water you drink, the more thirsty you become? Why is the table in a pen? Where is the computer on the motherboard? What percentage of California is the United States of America?
Logic	How to sit or stand at the same time? Where will the dead come back to life?
Event	How many times did Aristotle use a computer? When did Zuckerberg start Google? When Homer wrote The Odyssey?

Table 10: Initial FPQs

A.2 Distribution Balance Criterion

We expect our dataset to have a richer and more uniform distribution of FPQs. We achieve this goal with the help of constraints on the FPQ types. For the eight error types, each type of FPQ should account for at least 5% of the overall data, and the maximum category should not exceed 30%. And for the six problem formats, each type of FPQ should account for at least 10% of the entire data, and the maximum category should not exceed 30%. All balance criteria do not take into account the "other" category.

B Experiment Details [7]

B.1 API Calls for Pilot Experiments

We summarize the APIs used in Section 3.2 in Table 11. We will also provide the screenshot of using these APIs in our final reproducible code.

B.2 Details of Discriminating FPQs

For the experiments in Table 6, we use the prompt learning (Schick and Schütze, 2021) paradigm. We use "true" and "false" as the label word for FPQ and TPQ, respectively [8]. For T5 models, following the usage of T5 (Raffel et al., 2020) in their original paper, we append "*potential tricky question:*" to identify the task. MACAW models are multi-angle QA models, to use their direct question angle, we follow their paper and use "*$answer$; $question$* = " as the prefix. For OPT models, we train them in a vanilla input-output format. We list the hyperparameters for each experiment in Table 12. For MACAW-11B, we use half-precision acceleration and do not find performance degradation compared to full-precision computation. For the experiment in Figure 3, we use the same input-output format mentioned before. Our hyperparameters used in this section are listed in Table 12.

B.3 Details of Answering FPQs

Since fine-tuned models in few-shots (e.g. 32 question pairs) sometimes may not generate "*tricky/true question*" at the beginning of sentence [9], and a normal answer hardly has "*tricky/true question*" in it, we count whether "*tricky question*" or "*true question*" appears in outputs for classification evaluation to get the recall, precision, and accuracy scores. When evaluating the generated explanation, we remove "*tricky question*" and "*true question*". We list our hyperparameters used in this section in Table 13 and keep them the same when adding the binary loss.

[7] We choose random seeds 4, 13, and 34 in all experiments.

[8] Since our target is to classify whether it has a false premise, we set True for FPQs and False for TPQs.

[9] Some seeds in OPT models sometimes produce "this is a *tricky question*".

Model	API URL	Prompt Template	Hyperparameters
Bloom	https://huggingface.co/bigscience/bloom	Question: ___ Answer:	Sampling Strategy: greedy
OPT	https://opt.alpa.ai	Question: ___ Answer:	Response Length: 64; Temperature: 0.7; Top-p: 0.7
GPT-3	https://beta.openai.com/playground	Question: ___ Answer:	Temperature: 0.7; Maximum length: 256; Top-p: 1
Jurassic-1	https://api.ai21.com/studio/v1/j1-jumbo/complete	Question: ___ Answer:	Temperature: 0; TopK: 0; TopP: 1; MaxTokens: 32

Table 11: The APIs and hyperparameters when using the APIs.

# QP	Model	Learning Rate	Batch Size	Epoch
32	OPT-350M	$1e-5$	32	5
	OPT-1.3B	$1e-5$	32	5
	OPT-2.7B	$1e-5$	32	5
	MACAW-Large	$2e-5$	32	5
	MACAW-3B	$1e-4$	32	5
	MACAW-11B	$1e-4$	32	5
128	OPT-350M	$1e-5$	32	5
	OPT-1.3B	$1e-5$	32	5
	OPT-2.7B	$1e-5$	32	5
	MACAW-Large	$2e-5$	32	5
	MACAW-3B	$1e-4$	32	5
	MACAW-11B	$1e-4$	32	5
256	OPT-350M	$1e-5$	32	5
	OPT-1.3B	$1e-5$	32	5
	OPT-2.7B	$1e-5$	32	5
	MACAW-Large	$1e-4$	32	5
	MACAW-3B	$1e-4$	32	5
	MACAW-11B	$1e-4$	32	5
512	OPT-350M	$1e-5$	32	5
	OPT-1.3B	$1e-5$	32	5
	OPT-2.7B	$1e-5$	32	5
	MACAW-Large	$1e-4$	32	5
	MACAW-3B	$1e-4$	32	5
	MACAW-11B	$1e-4$	32	5
1187	OPT-350M	$1e-5$	32	5
	OPT-1.3B	$1e-5$	32	5
	OPT-2.7B	$1e-5$	32	5
	T5-Large	$1e-4$	32	5
	T5-3B	$1e-4$	32	5
	T5-11B	$1e-4$	32	5
	MACAW-Large	$1e-4$	32	5
	MACAW-3B	$1e-4$	32	5
	MACAW-11B	$1e-4$	32	5

Table 12: Hyperparameters for discriminating FPQs.

# QP	Model	Learning Rate	Batch Size	Epoch
32	OPT-2.7B	$5e-6$	8	16
	MACAW-3B	$3e-5$	8	8
	MACAW-11B	$1e-4$	4	3
256	OPT-2.7B	$3e-6$	32	12
	MACAW-3B	$3e-5$	32	8
	MACAW-11B	$2.5e-4$	4	3
1187	OPT-2.7B	$6e-6$	32	8
	MACAW-3B	$5e-5$	16	8
	MACAW-11B	$1e-4$	4	3

Table 13: Hyperparameters for answering FPQs.

B.4 Details of In-context Learning

In-context learning, introduced in GPT-3 (Brown et al., 2020), has been a successful way of adapting extensive language models. In in-context learning, we provide a textual prefix p of the task and one or a few training data samples before sending the input questions. We adopt the QA prefix in the GPT-3 demo for all the PLMs tested. Specifically, the prefix is:

> *p = I am a highly intelligent question answering bot. If you ask me a question that is rooted in truth, I will give you the answer. If you ask me a question that is nonsense, trickery, or has no clear answer, I will say "tricky question." first and give the reason, otherwise I will say "true question." first and give the reason.*

A few pairs of FalseQA samples $\{(q_F^i, a_F^i), (q_T^i, a_T^i)\}$ can be concatenated to the textual instruction. Therefore the full prefix before the input question has the following form:

$$p + \text{Q:}q_F^i + \text{A:}a_F^i + \text{Q:}q_T^i + \text{A:}a_T^i + ... + \text{Q:}\_\_\_ + \text{A:}$$

where $+$ indicates string concatenation, and the input example is filled into the blank.

We list our hyperparameters for in-context learning in Table 14.

B.5 Answering FPQs and General Questions

We list our hyperparameters in this section in Table 15. We count the number of general questions when using the data replay technique in Table 16.

Model	API URL	Hyperparameters
GPT-3	https://beta.openai.com/playground	Temperature: 0; Top-p: 1; Maximum length: 32
Jurassic-1	https://api.ai21.com/studio/v1/j1-jumbo/complete	Temperature: 0; TopK: 0; TopP: 1; MaxTokens: 32

Table 14: The APIs and hyperparameters for performing in-context learning.

# QP	Model	Learning Rate	Batch Size	Epoch
256	OPT-2.7B	$3e-6$	32	12
	MACAW-11B	$2.5e-4$	4	3
1187	OPT-2.7B	$6e-6$	32	8
	MACAW-11B	$1e-4$	4	3

Table 15: Hyperparameters for handling both FPQs and general questions.

# QP	Model	# General Questions
256	OPT-2.7B	32
	MACAW-11B	20
1187	OPT-2.7B	96
	MACAW-11B	80

Table 16: How many general questions models seen when performing data replay.

C Additional Results

C.1 More Raw PLM's Responses to FPQs

We present three more examples of PLM's responses to the FPQs and their responses to the corresponding questions that directly ask about the correctness of the premises in Table 17. We can see that in most cases PLMs identify whether the premises are true or false successfully, however, they fail on the FPQs.

C.2 Model-generated Answers and Explanations

We present randomly sampled FPQs in the test split and the corresponding references, discrimination results, and explanations/answers in Table 18. We use MACAW-11B trained with full training data while binary loss is added in this demonstration. We can see that in most cases, the explanation generated by the model is close to the reference. However, there are cases that the generated explanation is counterfactual. For example, "*A spider's shell is not helpful to its breath*" is incorrect.

C.3 The Questions in ARC-DA that MACAW-FPQ Rebuts

We show the problem that the model still rebuts after data replay. Specifically, we show the model results for the MACAW-11B model after training on the full training data as well as the replayed data. Since our experiments have three seeds, we show the problem that the model refutes in all seeds. We also show the explanations generated by our model, we randomly pickle one explanation from the three seeds. As we can see in Table 19, the correctness of the premises of these questions is not very clear. As a human, these questions can also be seen as questions containing false premises.

The question in Table 19 "*How is a skin cell from a mouse similar to an amoeba?*" can be seen as a question that contains a false premise "*A mouse's skin cells, like amoebas, are single-celled organisms.*", as a human, we may also rebut this presupposition. For the question "*Volcanoes are considered constructive because they*", generally, the volcanoes are considered destructive unless we want a creative answer. If a user truly wants the creative answer, he might provide explicit instructions to the PLM to trade robustness for creativity, which can be future work.

Model	# Para	Reply	Correctness	Ablation
Why there is just one season in China?				How many seasons does China have?
Bloom	176B	Because the Chinese government is afraid of the virus spreading.	✗	China has four seasons ...
OPT	175B	Because there are too many people ...	✗	China has four Seasons. Spring, ...
Jurassic-1	178B	Because China is a big country.	✗	China has four distinct seasons.
GPT-3	175B	In China, there is only one season because of its geographical location.	✗	China has four seasons: spring, summer, autumn, and winter.
MACAW	11B	It is a landlocked country.	✗	Four.
MACAW-FPQ	11B	China has four seasons, not one.	✓	China has four seasons.
What was the place where the tennis match was launched in the 1200s?				In which year was the tennis invented?
Bloom	176B	The first tennis match was played in the 1200s in the town of ...	✗	The tennis was invented in the year 1874.
OPT	175B	The tennis match was launched in France.	✗	The tennis was invented in the year 1874.
Jurassic-1	178B	The tennis court was in the palace of Westminster.	✗	Tennis was invented in 1874.
GPT-3	175B	The first recorded tennis match was launched in France ...	✗	The modern game of tennis is believed to have been invented in 1873...
MACAW	11B	Paris.	✗	1887.
MACAW-FPQ	11B	Tennis was invented in the 19th century.	✓	In 1887.
List a software that is developed by Edison.				Did Edison develop any software?
Bloom	176B	Edison is a software that is developed by Edison.	✗	Yes, he invented the first ...
OPT	175B	The Edison Mail.	✗	Yes. Edison invented the first electric light bulb,
Jurassic-1	178B	Edison is developed by Edison.	✗	Yes, Edison developed several pieces of software.
GPT-3	175B	Edison Design System (EDS) is a software suite developed by Edison ...	✗	No, Thomas Edison did not develop any software...
MACAW	11B	Edison tv.	✗	No.
MACAW-FPQ	11B	Edison was a physicist, not a software developer.	✓	Tricky question. Edison was a physicist, not a computer scientist...

Table 17: Additional examples of PLMs' responses to randomly selected FPQs from the test split of FalseQA.

No.	Question	Reference	Classification	Explanation/Answer
1	Why are birds not a kind of animals?	By definition, birds is a type of animal.	✓	Birds are animals.
2	Is a spider's shell helpful to its breath?	A spider has no shell.	✓	A spider's shell is not helpful to its breath.
3	Why can't lithium burn?	Lithium burns and explodes easily.	✗	Because lithium is a metal.
4	When do the feet of the cane grow?	Cane is not an animal, so it doesn't have feet.	✓	Canes are plants, not animals, and they don't have feet.
5	List a type of clock that has two ears.	A clock is not an animal, it doesn't have two ears.	✓	Clocks are mechanical devices, not biological, and have no ears.
6	How long will it take plants to grow into adult animals?	Plants don't grow into adult animals, plants grow into a tree or bigger plants.	✓	Plants are plants, animals are animals, they can't grow into plants.
7	What material was the camera made of in 1350?	The camera was invented in 1839.	✓	The camera obscura was invented in 1839.
8	Which place has eight seasons?	There are only 4 seasons in total.	✓	There are only four seasons.
9	Why does the pregant woman have two bellies?	A pregnant woman has only one belly.	✓	Pregnant women have one belly.
10	If foxes live in the sea, how often do they come ashore?	Foxes have no gills and cannot live in the sea.	✓	Foxes live on land.

Table 18: Randomly sampled FPQ questions in the test split. Column "Reference" presents one of the two reference explanations. Column "Classification" denotes whether the model successfully classifies the question as FPQ. Column "Explanation" gives the model-generated explanation for the rebuttal case and the model-generated answer to the non-rebuttal case.

No.	Question	Explanation
1	One explanation of why carnivores are dependent on plants is because they	Carnivores are carnivores, they don't depend on plants.
2	What is responsible for changing the remains of prehistoric sea animals into natural gas?	Prehistoric sea animals were fossilized in the sedimentary rock, not in the gas form.
3	Which planet is fourth from the Sun in our solar system?	The fourth planet from the Sun is the Moon.
4	How do sprouting plants exhibit positive gravitropism?	Plants are not animals, they are not able to gravitropise.
5	Volcanoes are considered constructive because they	Volcanoes are destructive because they release lava.
6	How is a skin cell from a mouse similar to an amoeba?	Amoebas are single-celled organisms, not skin cells.

Table 19: All questions from ARC-DA dataset that the model trained with the data replay technique still rebuts across all seeds. The first and fifth questions are not incomplete because they take the format of sentence continuation.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
page 9 Section Limitations

☑ A2. Did you discuss any potential risks of your work?
page 9 Section Ethical Considerations

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 0, Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 4 Dataset and Section 5 Experiment

☑ B1. Did you cite the creators of artifacts you used?
scientific artifacts

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
The dataset is either public or created by ourselves.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section 4 Dataset and Section 5 Experiment

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Section 4 Dataset

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 4 Dataset

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 4 Dataset

C ☑ Did you run computational experiments?

Section 5 Experiment

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 5 Experiment

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 5 Experiment and Appendix B

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 5 Experiment

☐ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
No response.

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Section 4 Dataset

☒ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
The instructions are not in English.

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Section 4 Dataset

☑ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Section 4 Dataset

☒ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not yet, but we will try to get one.

☒ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
We will provide the details only after publication. Currently, providing such information might potentially reveal our identity.