A Semantic-based Method for Unsupervised Commonsense Question Answering

Yilin Niu[1]*, **Fei Huang**[1]*, **Jiaming Liang**[1], **Wenkai Chen**[2], **Xiaoyan Zhu**[1], **Minlie Huang**[1]†
[1] The CoAI group, DCST; [1] Institute for Artificial Intelligence;
[1] State Key Lab of Intelligent Technology and Systems;
[1] Beijing National Research Center for Information Science and Technology;
[1] Tsinghua University, Beijing 100084, China.
[2] School of Computer Science and Technology,
Beijing University of Posts and Telecommunications
niuyl14@tsinghua.org.cn {f-huang,liangjm18}18@mails.tsinghua.edu.cn
wkchen630@gmail.com {zxy-dcs,aihuang}@tsinghua.edu.cn

Abstract

Unsupervised commonsense question answering is appealing since it does not rely on any labeled task data. Among existing work, a popular solution is to use pre-trained language models to score candidate choices directly conditioned on the question or context. However, such scores from language models can be easily affected by irrelevant factors, such as word frequencies, sentence structures, etc. These distracting factors may not only mislead the model to choose a wrong answer but also make it oversensitive to lexical perturbations in candidate answers.

In this paper, we present a novel SEmantic-based Question Answering method (SEQA) for unsupervised commonsense question answering. Instead of directly scoring each answer choice, our method first generates a set of plausible answers with generative models (e.g., GPT-2), and then uses these plausible answers to select the correct choice by considering the semantic similarity between each plausible answer and each choice. We devise a simple, yet sound formalism for this idea and verify its effectiveness and robustness with extensive experiments. We evaluate the proposed method on four benchmark datasets, and our method achieves the best results in unsupervised settings. Moreover, when attacked by TextFooler (Jin et al., 2020) with synonym replacement, SEQA demonstrates much less performance drops than baselines, thereby indicating stronger robustness.

1 Introduction

Pre-trained language models have been widely used for commonsense question answering. Finetuning pre-trained models on task-specific data produces many state-of-the-art results (Wang et al., 2020; Khashabi et al., 2020; Lin et al., 2019). However, this requires amounts of labeled task data. Therefore, it is vital to study unsupervised commonsense question answering without relying on any labeled downstream task data. In this paper, we investigate multiple-choice commonsense question answering tasks in an unsupervised setting: given a question and a set of answer choices, a model is required to predict the most reasonable answer choice for the question, but without access to any labeled task data.

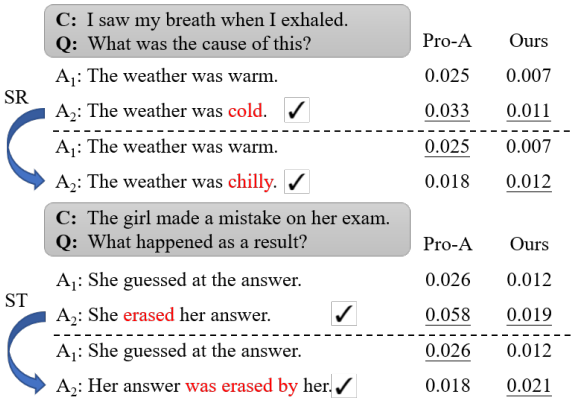


Figure 1: Two examples of commonsense question answering, where the baseline (Pro-A) is oversensitive to lexical perturbations (SR for synonym replacement and ST for sentence structure transformation). The scores from Pro-A and our method for each answer choice are shown in the right columns. The underlined score indicates the answer choice selected by a method.

Many existing unsupervised methods tackle these tasks by scoring each answer choice using a language model, e.g., estimating the generative probability of the answer choice conditioned on the question (Trinh and Le, 2018; Shwartz et al., 2020; Bosselut and Choi, 2019; Tamborrino et al., 2020). Table 1 lists several typical score functions. However, these scores can be easily influenced by word frequencies, sentence structures, and other

*Equal contribution
†Corresponding author: Minlie Huang.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 3037–3049
August 1–6, 2021.

factors, which can mislead the models and make existing methods oversensitive to lexical perturbations (Abdou et al., 2020; Tamborrino et al., 2020). Figure 1 shows two examples. The correct choices are paraphrased via synonym replacement or structure transformation. In these examples, the baseline (Pro-A) produces much lower scores for the paraphrased choices and chooses the wrong choices.

Since existing methods can be easily distracted by irrelevant factors such as lexical perturbations, we argue that a commonsense question answering method should **focus on the answers' semantics and assign similar scores to synonymous choices**. To this end, we introduce a novel SEmantic-based Question Answering model, SEQA, which aims to robustly select correct answers in multi-choice commonsense question answering in an unsupervised setting. Instead of directly scoring an answer choice, we calculate the probability of observing the choice's semantics. A choice's semantic score can be obtained by summing the generative probabilities of sentences that have the same semantic meanings with the choice, where the sentences are called the choice's *supporters*. However, it is hard to obtain the *supporters* which have exactly the same semantic meanings with the choice, so we reformulate the semantic score into a soft version as explained in Section 3.2. Each *supporter* is weighed by the semantic similarity to the answer choice, which can be computed with some off-the-shelf models, such as SentenceBERT (Reimers and Gurevych, 2019). Since the *supporters* and their weights depend on the semantics rather than the surface form of the answer choice, by this means, the effects of the distracting factors can be largely suppressed. Moreover, synonymous choices are likely to share the same set of *supporters*, so their scores are expected to be stably close. Our contributions in this paper are summarized as follows:

- We propose a semantic-based question answering model (SEQA) for robust commonsense question answering in an unsupervised setting. Instead of directly scoring the answer choices, our method first generates some plausible answers and then uses them to select the correct choice by considering the semantic similarity between each plausible answer and each choice.
- We conduct experiments on four commonsense question answering datasets, where SEQA achieves the best performance compared with strong baselines. When attacked by TextFooler (Jin et al., 2020) with synonym replacement, our method performs remarkably more robustly.

Method	Score Function
Pro-A	$[P_{LM}(A\|Q)]^{\frac{1}{\|A\|}}$
Pro-Q	$[P_{LM}(Q\|A)]^{\frac{1}{\|Q\|}}$
MI-QA	$\left[\frac{P_{LM}(A\|Q)}{P_{LM}(A)}\right]^{\frac{1}{\|A\|}}$
SEQA (Ours)	$\sum_{S\in\mathbb{A}} \omega(S\|A) P_{LM}(S\|Q)$

Table 1: Three existing score functions and our method for unsupervised commonsense question answering. Q is the question and A is the choice. $\mathbb{A}$ is the set of all possible answers and $\omega(S|A)$ is a weighting function defined in Eq.(5). LM refers to a pre-trained language model, such as GPT-2 or BERT[1] (Devlin et al., 2019).

2 Related Work

Previous work has explored pre-trained language models (LMs) for unsupervised commonsense question answering. In general, these approaches treat LMs as question answering modules.

Table 1 shows three representative methods, which do not use external knowledge and rely fully on the implicit knowledge encoded in LMs for reasoning. Probability-A (Pro-A) considers the generative probability of the choice conditioned on the question. However, it suffers from the statistical bias of choices, such as word frequency and sentence length (Abdou et al., 2020). To alleviate this, MutualInfo-QA (MI-QA) calculates the mutual information between the question and the choice. Another way to reduce the impact of statistical bias is to score each choice using the conditional probability of the question rather than the choice (Trinh and Le, 2018; Tamborrino et al., 2020) , which is denoted as Probability-Q (Pro-Q) in Table 1.

Some recent work claims that external knowledge can benefit commonsense reasoning. Besides static knowledge bases (KBs), such as ConceptNet (Speer et al., 2017) and Atomic (Sap et al., 2019a), there are also numerous studies treating LMs as dynamic KBs. Petroni et al. (2019) shows that LMs can be used for KB completion. And Davison et al. (2019) shows that BERT can distinguish true and fake ConceptNet triplets. Further, the extracted knowledge can work as complementary information for answering a question. Rajani et al. (2019) proposes a model for Com-

[1] $P_{BERT}(Q|A) \triangleq \prod_i^{|Q|} P_{BERT}(Q_i|Q_{/i}, A)$.

monSenseQA (Talmor et al., 2019) that generates explanations for questions, which are then used as additional inputs. The shortcoming of this approach is that it requires collecting human explanations for each new dataset to fine-tune LMs. Some following researches explore unsupervised explanation/knowledge generator. CGA (Bosselut and Choi, 2019) employs COMET (Bosselut et al., 2019) to generate intermediate inferences which are then used to score the choice. However, COMET is limited by a small set of question types so that CGA is difficult to generalize to different domains. Self-Talk (Shwartz et al., 2020) breaks the limit by extracting knowledge from GPT-2 (Radford et al., 2019), which has no restriction on the query types. Thus, Self-Talk can be applied to a wide range of domains. Despite the introduction of auxiliary information, these methods are essentially dependent on language model scores, so they are still sensitive to lexical perturbations.

Besides directly using pre-trained LMs, some recent efforts have been dedicated to automatically constructing task-specific data to train commonsense reasoners in zero-shot settings. Wang et al. (2019) and Kocijan et al. (2019) provide some rules to construct labeled training data from large corpus for pronoun disambiguation. Banerjee and Baral (2020), Moghimifar et al. (2020) and Ma et al. (2020) collect training data based on knowledge bases, such as Atomic (Sap et al., 2019a). Though effective, they are limited by the specific task settings or highly dependent on the task-related knowledge bases, which makes them difficult to transfer to other commonsense reasoning tasks.

3 Method

In this paper, we focus on unsupervised multiple-choice commonsense question answering, which is formalized as follows: given a question and a set of choices, models should select the correct choice:

$$\hat{A} = \underset{A}{\operatorname{argmax}}\, s(A|Q),$$

where s refers to a score function. Note that we have no access to any labeled task data.

3.1 Motivation

In existing unsupervised methods, the score functions are usually defined based on the language model scores. Taking Pro-A (Table 1) as an example, it first converts the question into a statement:

- Q: I saw my breath when I exhaled. What was the cause of this? $\longrightarrow$ Rewrite: I saw my breath when I exhaled because ___

And it then takes the statement as a prompt to calculate the generative probability of each choice. Note that the templates for rewriting is not the focus of this paper, and hence we directly use the templates of previous work (Shwartz et al., 2020; Tamborrino et al., 2020) for our method and all the baselines in this paper (see Appendix for details).

Though successful, language model scores can be affected by many distracting factors, such as word frequency and sentence structure, etc. These factors can disturb the score functions to a large extent, as shown in Figure 1. Our goal is to alleviate the influence of these distracting factors. Hence we propose a new method for unsupervised commonsense question answering, which achieves better results and performs more robustly.

3.2 SEQA

SEQA is designed to predict the semantic score of an answer choice A. Instead of directly estimating the probability $P(A|Q)$ of the single choice A, the semantic score focuses on the probability $P(M_A|Q)$ where M_A represents A's semantics. Ideally, we decompose $P(M_A|Q)$ into the summation of the conditional probabilities of A's *supporters*, where the *supporters* indicates all possible answers that have exactly the same semantics M_A. Formally, the semantic score is defined as

$$s(A|Q) \triangleq P(M_A|Q) = \sum_{S \in \mathbb{S}_A} P_{LM}(S|Q) \quad (1)$$

$$= \sum_{S \in \mathbb{A}} \mathbb{I}(S \in \mathbb{S}_A) P_{LM}(S|Q). \quad (2)$$

$\mathbb{S}_A$ is the set of *supporters* of choice A, and $\mathbb{A}$ is the set of all possible answers. $\mathbb{I}(S \in \mathbb{S}_A)$ is an indicator function indicating whether S is a *supporter* of A. To obtain the *supporter* set $\mathbb{S}_A$, we adopt a model to extract the sentence-level semantic features. Ideally, the indicator function is defined as

$$\mathbb{I}(S \in \mathbb{S}_A) = \begin{cases} 1 & \text{if } \cos(h_S, h_A) = 1, \\ 0 & \text{if } \cos(h_S, h_A) < 1, \end{cases} \quad (3)$$

where h_A is the semantic features of sentence A, and we assume that S and A are exactly the same in semantics if h_S and h_A point in the same direction.

However, Eq.(3) uses a hard constraint that $\cos(h_S, h_A)$ exactly equals to 1, which can be too

strict to find acceptable *supporters*. Therefore, we reformulate Eq.(2) into a soft version:

$$s(A|Q) \triangleq \sum_{S \in \mathbb{A}} \omega(S|A) P_{LM}(S|Q), \quad (4)$$

where the indicator function in Eq.(2) is replaced by a soft function $\omega(S|A)$. To emulate $\mathbb{I}(S \in \mathbb{S}_A)$, $\omega(S|A)$ is expected to meet three requirements: (1) $\omega(S|A) \in [0, 1]$ for any S and A; (2) $\omega(S|A) = 1$ if $\cos(h_S, h_A) = 1$; (3) $\omega(S|A)$ increases monotonically with $\cos(h_S, h_A)$. There are several different definitions of $\omega(S|A)$ meeting these requirements, which are explored in Section 4.7.3. In this paper, $\omega(S|A)$ is defined as:

$$\omega(S|A) = \frac{1}{Z(T)} \exp\left[\frac{\cos(h_S, h_A)}{T}\right]. \quad (5)$$

T is the temperature, and $Z(T) = \exp(\frac{1}{T})$ is a normalization term that makes $\omega(A|A) = 1$. If $T \to 0$, $\omega(S|A)$ degenerates to the indicator function. If $T > 0$, $\omega(S|A)$ relates to the von Mises-Fishers distribution over the unit sphere in the feature space, where the acceptable feature vectors are distributed around the mean direction $\frac{h_A}{||h_A||}$.

Since it is intractable to enumerate all possible answers in $\mathbb{A}$, we convert Eq.(4) to an expectation over $P_{LM}(S|Q)$:

$$\begin{aligned} s(A|Q) &= \mathbb{E}_{S \sim P_{LM}(S|Q)} [\omega(S|A)] \\ &\approx \frac{1}{K} \sum_{i=1}^{K} \omega(S_i|A) \quad (6) \\ &= \frac{1}{K \cdot Z(T)} \sum_{i=1}^{K} \exp\left[\frac{\cos(h_{S_i}, h_A)}{T}\right], \quad (7) \end{aligned}$$

where $S_1, \cdots, S_K$ are sentences sampled from $P_{LM}(\cdot|Q)$, and K is the sample size. h_A and h_{S_i} can be extracted from a pre-trained model, e.g., SentenceBERT (Reimers and Gurevych, 2019).

From Eq.(7), we can see the semantic score $s(A|Q)$ is only dependent on the semantic feature h_A and regardless of A's surface form. Therefore, our method will produce similar semantic scores for synonymous choices, assuming that the synonymous choices have similar semantic features.

3.3 The Voting View of SEQA

At the beginning of Section 3.2, we define the semantic score as the summation of the conditional probabilities over the *supporters*. However, in Eq.(7), the sampled sentences $S_1, \cdots, S_K$ are not A's *supporters* because they may not be semantically similar to A. To address the differences, we name the sampled sentences $S_1, \cdots, S_K$ as *voters*, which are plausible answers to the question Q. In this section, we will show another view of our method, which works like a procedure that the *voters* vote out the correct choice.

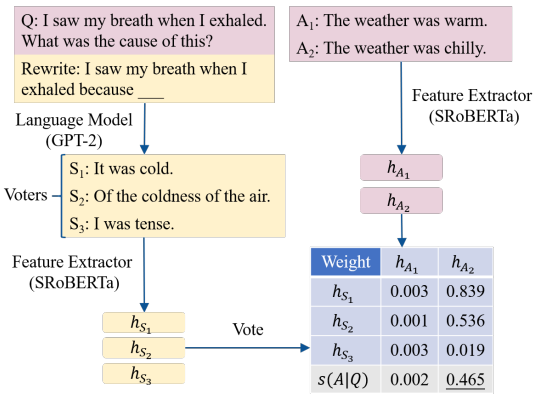


Figure 2: Process of SEQA in the view of voting. We use the same templates with previous work (Shwartz et al., 2020; Tamborrino et al., 2020) to rewrite interrogative sentences into declarative ones. And then use GPT-2 to generate some plausible answers as *voters* S_i, conditioned on the rewritten question. The choices and *voters* are encoded via SentenceRoBERTa to obtain semantic features, h_{A_j} and h_{S_i}, which are then used to calculate the voting weights $\omega(S_i|A_j)$. The choice with the largest score $s(A_j|Q)$ is selected as the answer.

Suppose there are two candidate choices A_1 and A_2, our method is to find the correct choice according to the semantic scores, $s(A_1|Q)$ and $s(A_2|Q)$. Following Eq.(6), our method can be decomposed into two steps: First, **sample** some *voters* $S_1, \cdots, S_K$ from $P_{LM}(\cdot|Q)$. This step only considers the question Q but no candidate choices. Second, each *voter* votes for the choices with the semantic similarity weights. For example, S_i votes for A_j with the weight of $\omega(S_i|A_j)$. The candidate choice that receives more votes will have a higher semantic score and be selected as the final answer.

Figure 2 shows the process of SEQA in the view of voting. Although the voting view is intuitive, the formalism in Section 3.2 provides more insights: (1) Our method approximates the probability of semantics, which works as the theoretical basis of SEQA. (2) Our method can be seen as an extension of Pro-A (see Table 1), since Pro-A only calculates the language model score for a single sentence, whereas our method calculates the semantic score for a set of *supporters*. (3) Eq.(4) provides guidance, the three requirements mention before, for the design of the voting weight function $\omega(S|A)$. Specifically, the guidance explains the rationality of the formulation of Eq.(5).

Dataset	Method	Pre-trained Models	Original Accuracy (↑)	After-Attack Accuracy (↑)	Attack Success Rate (↓)	Percentage of Perturbed Words	Semantic Similarity
COPA	Pro-A	GPT-2	73.6	4.6	93.8	17.3	0.883
	Pro-Q	RoBERTa	**79.4**	23.0	71.0	22.9	0.828
	MI-QA	GPT-2	74.6	16.2	78.3	19.9	0.865
	Self-talk	COMET+GPT-2	68.6	8.4	87.8	19.8	0.855
	CGA	GPT-2	72.2	4.8	93.4	17.1	0.886
	SEQA	GPT-2+SRoBERTa	**79.4**	**59.0**	**25.7**	21.7	0.827
SCT	Pro-A	GPT-2	72.3	4.8	93.3	14.3	0.917
	Pro-Q	RoBERTa	56.3	22.3	60.3	18.1	0.872
	MI-QA	GPT-2	66.1	29.2	55.8	16.2	0.885
	Self-talk	COMET+GPT-2	70.4	4.7	93.3	14.2	0.915
	CGA	GPT-2	71.5	4.8	93.2	14.3	0.916
	SEQA	GPT-2+SRoBERTa	**83.2**	**69.4**	**16.5**	18.3	0.856
SocialIQA	Pro-A	GPT-2	46.0	16.2	64.7	21.0	0.876
	Pro-Q	RoBERTa	42.2	27.8	34.2	23.2	0.843
	MI-QA	GPT-2	41.2	24.6	40.4	25.3	0.866
	Self-talk	COMET+GPT-2	**47.5**	12.3	74.0	22.2	0.872
	CGA	COMET	45.4	18.4	59.4	22.3	0.867
	SEQA	GPT-2+SRoBERTa	**47.5**	**38.2**	**19.5**	23.5	0.839
CosmosQA	Pro-A	GPT-2	36.8	1.3	96.4	9.2	0.927
	Pro-Q	RoBERTa	21.5	5.0	76.6	13.7	0.859
	MI-QA	GPT-2	29.3	7.4	74.8	12.1	0.886
	Self-talk	COMET+GPT-2	36.1	1.2	96.7	8.9	0.928
	CGA	GPT-2	42.4	1.7	96.0	9.6	0.924
	SEQA	GPT-2+SRoBERTa	**56.1**	**32.6**	**41.8**	13.9	0.859

Table 2: Evaluation results, including the original selection accuracy before attack, the accuracy after attack, the attack success rate, the percentage of perturbed words with respect to the original sentence length in successful attacks, and the semantic similarity between the original and paraphrased choices. GPT-2, RoBERTa and SRoBERTa refer to GPT-2-xlarge, RoBERTa-large (Liu et al., 2019) and SentenceRoBERTa-large, respectively.

4 Experiments

4.1 Datasets

We conducted experiments on four multiple-choice commonsense question answering tasks, COPA (Roemmele et al., 2011), StoryClozeTest (SCT) (Mostafazadeh et al., 2016), SocialIQA (Sap et al., 2019b) and CosmosQA (Huang et al., 2019). For each instance, only one choice is correct. See Appendix for more description about datasets.

For COPA, we reported the results on its test set. As the test sets of another three datasets are hidden, for convenience of analysis, we reported the experiment results on their development sets.

4.2 Baselines

We employed five strong baselines. Table 1 shows three of them, **Pro-A**, **Pro-Q** and **MI-QA**. There is no explicit auxiliary information used in these three methods, while another two baselines rely on explicit information supplementation. **CGA** (Bosselut and Choi, 2019) and **Self-Talk** (Shwartz et al., 2020) query pre-trained language models (e.g., GPT-2, COMET (Bosselut et al., 2019)) for relevant knowledge, which forms part of contexts. And then, similar to Pro-A, they take the generative probabilities of choices as scores.

4.3 Experiment Settings

For each method, we tried different pre-trained language models (see Appendix for details), and then selected the pre-trained LMs that maximized the accuracy on each dataset. The details of the selection of pre-trained LMs can be found in Table 2.

For SEQA, we used GPT-2 to generate *voters* via Nucleus Sampling (Holtzman et al., 2020) with $p = 0.9$. The sample size K of *voters* is set to 500. In Section 4.7.2, we show that a small sample size can also lead to superior performance. Self-Talk and CGA also rely on the generated answers from GPT-2 or COMET. Different from SEQA, for these two baselines, more generated answers will not always lead to better performance (see Section 4.7.2). Thus, we selected the optimal sample size for them rather than the same sample size with SEQA.

When evaluating SEQA on COPA, we tuned the temperature T on its development set, and then reported the results on the test set with the tuned temperature $T = 0.1$. Due to the absence of test sets of other datasets, we evaluated SEQA on their development sets without tuning the temperature and directly set $T = 0.1$.

4.4 Main Results

Table 2 shows the evaluation results about accuracy and robustness.

4.4.1 Accuracy

Among all the methods, SEQA achieved the best performance on all the datasets. Especially on SCT and CosmosQA, SEQA outperformed the best baselines by more than 10 points. It can be inferred that the semantic scores are beneficial for commonsense question answering due to the reduction of distracting factors. Pro-Q performed better than other baselines on COPA, perhaps because it suffered less from the statistic bias of choices (Tamborrino et al., 2020). However, Pro-Q lost its superiority on another three datasets, because it is unsuitable for processing long or complex contexts.

4.4.2 Robustness

To test the robustness under the synonym replacement attack, we used TextFooler (Jin et al., 2020) to attack the methods by perturbing the correct choices of the correctly predicted examples. The percentage of perturbed words refers to what percentage of words in choices are replaced in successful attacks. The semantic similarity is measured between the paraphrased choice and the original choice. Considering the attack success rate and the after-attack accuracy, SEQA is much more robust than all baselines. To be specific, the attack success rates on SEQA are at least 39 points lower than those of Pro-A, CGA, and Self-Talk on all datasets. MI-QA and Pro-Q are designed to reduce the impact of statistic bias in choices, so that they can resist lexical perturbation to some extent. Even so, SEQA is remarkably lower than MI-QA and Pro-Q in terms of attack success rates on all datasets.

An observation is that the attack success rate on SEQA on CosmosQA is higher than those on the other datasets. The reason is that, the contexts in CosmosQA are so complex that GPT-2 is more difficult to generate high-quality answers. If there is a more powerful generator, the robustness of SEQA is expected to have a further improvement.

4.5 Consistency Testing

We have claimed that a commonsense question answering method should assign close scores to synonymous choices. To verify that SEQA better meets this requirement, we conducted consistency testing for all the methods on four datasets. For each example, the consistency testing of a method is conducted in three steps: (1) Originally, the example has one correct and several wrong answer choices. We randomly sample some choices from other examples as additional wrong choices. After that, the example will have one correct choice and 19 wrong choices. (2) Leverage a commonly used automatic translation service, Baidu Translation, to translate each choice from English into an intermediate language, and then back-translate it into English. During this process, we employ three intermediate languages, Chinese, Spanish, and Russian, because the translation quality of these languages is better than others. As a result, each choice is accompanied with three synonymous choices. (3) Use the commonsense question answering method to calculate the scores for each choice as well as its synonymous choices, and then sort all the choices according to their scores. Because the scoring scales of these methods are different, we calculate the standard deviation of the ranks of the correct choice and its synonymous choices.

Method / Dataset	COPA	SCT	SocialIQA	CosmosQA
Pro-A	9.1	11.0	11.7	9.4
Pro-Q	6.9	8.5	11.6	12.3
MI-QA	7.5	5.8	11.1	7.9
Self-Talk	13.3	9.5	10.7	10.1
CGA	9.7	11.0	10.9	9.5
SEQA	**4.1**	**3.2**	**5.8**	**4.7**

Table 3: Consistency testing where the methods rank 80 choices to find 4 correct ones for each example. The metric is the standard deviation of the ranks of 4 correct synonymous choices averaged over 500 examples.

Table 3 shows the average standard deviation of the ranks. As expected, the average standard deviation of SEQA is much lower than any other method on all the datasets, confirming that SEQA assigns more similar ranks and closer scores to synonymous choices. We also observed that MI-QA provided relatively stable predictions compared with other baseline methods. A possible explanation is that, the normalization term $P_{LM}(A)$ helps alleviate the influence of lexical perturbations.

4.6 Trends of Accuracy with Answer Length

Answer length is also a type of distracting factor which may mislead baseline methods. To explore to which extent answer lengths affect the performance of methods, we divided the development set of CosmosQA into four subsets according to the length of correct choice. Table 4 shows the results of SEQA and a robust baseline, MI-QA. Compared with MI-QA, SEQA has much more stable performance as answer lengths vary. The reason is that, SEQA focuses on semantic information so that it has stronger resistance to such distracting factors.

Method	Answer Length				
	All	[1,5]	[6,10]	[11,15]	[16,20]
MI-QA	29.3	51.6	27.9	24.4	23.8
SEQA	56.1	58.6	58.0	54.1	51.2

Table 4: The trends of accuracy with answer length for SEQA and MI-QA on CosmosQA.

T	COPA		SCT		SocialIQA		CosmosQA	
	Bef	Aft	Bef	Aft	Bef	Aft	Bef	Aft
10	75.6	48.8	82.0	64.7	46.3	35.9	52.7	22.3
1	76.4	48.8	82.4	64.5	46.6	36.1	53.3	22.4
0.2	77.0	52.8	**83.6**	66.3	46.9	36.8	54.8	26.1
0.1	79.4	**59.0**	83.2	**69.4**	**47.5**	**38.2**	**56.1**	**32.6**
0.05	**80.2**	54.6	80.8	61.4	46.0	36.5	55.1	28.8

Table 5: The before-attack (Bef) and after-attack (Aft) accuracy of SEQA with different temperatures.

4.7 Ablation Study

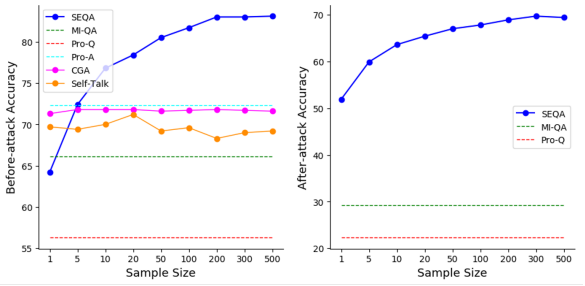
4.7.1 Analysis on Temperature

In the previous experiments, the temperature T of SEQA was set to 0.1 by default. To investigate the influence of T, we varied T in a wide range from 0.05 to 10 and report the results in Table 5. Considering that the temperature varied greatly, the performance of SEQA is relatively stable, indicating that SEQA is not so sensitive to the selection of T. Another observation is that, although the four datasets are different in domains and text length, the trends of performance with temperature on them are relatively similar, illustrating that the temperature selected on one task can be generalized to other tasks.

4.7.2 Analysis on Sample Size

Figure 3 shows the effect of the sample size K on SEQA. For comparison, Figure 3 also includes the results of baselines in the settings of before- and after-attack, respectively. Due to the limitation of space, the results on the other datasets are shown in Appendix. As expected, the before-attack and after-attack accuracy on SCT increased with the sample size. In detail, the rapid increase in performance occurred when $K < 100$, and then the improvement slowed down when $K > 100$. Finally, SEQA achieved a stable and relatively high performance.

CGA and Self-Talk also leverage LMs to generate some plausible answers. Different from our method, they use the generated answers to form part of the question, and then calculate the generative probability of the choice based on the augmented question. We also tried different sample sizes for the two methods, and Figure 3 (a) shows



(a) (b)

Figure 3: The before-attack (a) and after-attack accuracy (b) of methods with different sample sizes on SCT. The after-attack accuracy of Pro-A, CGA and Self-Talk is below 5.0%, and thus omitted in (b).

$\omega(S\|A) = \frac{1}{f(1)} f\left(\cos(h_S, h_A)\right)$	Bef	Aft
$f(x) = \mathbb{I}(x > \alpha)$	77.2	47.2
$f(x) = \mathrm{ReLU}(x - \beta)$	77.6	45.2
$f(x) = \mathrm{sigmoid}(\frac{x}{T})$	75.6	48.6
$f(x) = \exp\left(\frac{x}{T}\right)$	79.4	59.0

Table 6: The before-attack (Bef) and after-attack (Aft) accuracy of SEQA on the test set of COPA with different definitions of $\omega(S|A)$. α, β, T_1, T_2 are hyperparameters tuned on the development set of COPA.

that their accuracy will not stably increase with a larger sample size.

4.7.3 Analysis on $\omega(S|A)$

$\omega(S|A)$ in SEQA can be defined in different forms, as long as the three requirements mentioned in Section 3.2 are met. Besides the default definition, we explored another three forms of $\omega(S|A)$, and the experiment results on COPA are shown in Table 6. Although the performance varies with $\omega(S|A)$, the before-attack accuracy of SEQA still outperformed most of the baselines with any definition of $\omega(S|A)$. Moreover, SEQA maintains its obvious advantage in after-attack accuracy, which reflects the inherent robustness of SEQA.

	GPT-2		
	medium	large	xlarge
Avg. GloVe	56.6	59.6	61.2
SBERT-base	71.2	72.6	74.8
SRoBERTa-base	72.4	72.0	75.4
SRoBERTa-large	74.2	75.2	79.4

Table 7: SEQA's accuracy with different feature extractors and language models on COPA. Avg. GloVe means the average pooling of the pre-trained word embeddings (Pennington et al., 2014) over the sentence.

Score	3	2	1
Grammar	84.8%	12.8%	2.4%
Logic	40.8%	25.6%	33.6%

Table 8: Manual evaluation of the quality of *voters* (generated by GPT-2-xlarge conditioned on questions). Score 3/2/1 correspond to high, middle and low quality, respectively, in terms of grammar and logicality.

4.7.4 Analysis on Pre-trained Language Model and Feature Extractor

SEQA has no limit on the selection of the pre-trained language model and the feature extractor. Table 7 shows how the accuracy of SEQA on COPA varied with the language model and the feature extractor. As expected, more powerful extractor usually led to higher accuracy under the same settings of language models. Similar conclusion can be obtained for the language model. It can be inferred that, if there are more powerful language models or feature extractors in the future, the performance of SEQA may be further improved.

4.8 Analysis on the Quality of *Voters*

While the performance of SEQA served as an extrinsic evaluation for the quality of the *voters* (plausible answers sampled from $P_{LM}(\cdot|Q)$, described in Section 3.3), we were also interested in evaluating it intrinsically. We sampled 125 *voters* from COPA. For each *voter*, we provided crowdsourcing workers with the original question, and asked them: 1) whether the *voter* is grammatical, not entirely grammatical but understandable, or completely not understandable, 2) whether the *voter* is a reasonable answer to the question, not reasonable but relevant, or completely irrelevant. These evaluation tasks comprehensively examined the *voters* in grammar and logicality. The annotation tasks were carried out in Amazon Mechanical Turk, and we aggregated annotations from 3 workers using majority vote.

Table 8 shows the results of the human evaluation of the *voters*. Score 3/2/1 correspond to the high, middle and low quality, respectively. According to the grammar scores, 97.6% of the *voters* are grammatical or at least understandable, for which most of the *voters* belong to the natural language space. In terms of logicality, 40.8% of the *voters* are reasonable answers to the questions, which may not be very satisfying. However, in Section 4.9, we will show that SEQA makes prediction based on a small part of *voters*, and hence SEQA is robust even though there are some irrelevant *voters*.

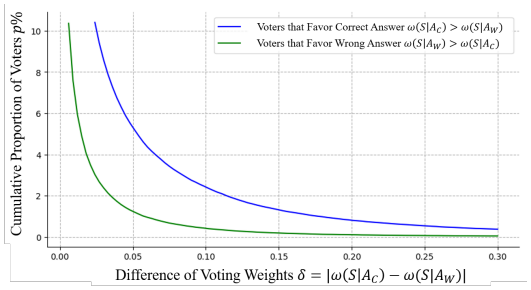


Figure 4: The cumulative proportion of *voters* favoring the correct answer A_C or the wrong answer A_W on COPA. Each point (δ, p) means that $p\%$ of *voters* satisfy $|\omega(S|A_C) - \omega(S|A_W)| \geq \delta$, where S refers to a *voter*. The area between the two curves equals to the difference of the semantic scores $s(A_C|Q) - s(A_W|Q)$.

4.9 Voting Weight Distribution

We visualize the cumulative proportion of *voters* favoring the correct or the wrong choices (see Figure 4). The curve is averaged over all instances in the test set of COPA, where we sampled 500 *voters* for each instance and set $T = 0.1$.

From the curves, we can find several properties of *voters*: (1) The *voters* favor the correct choices over the wrong choices, where the curve for correct choices is consistently above the curve for wrong ones. The area between two curves shows the difference of semantic scores $s(A_C|Q) - s(A_W|Q)$, which is a large gap compared with the area under the bottom curve. (2) 93.5% of *voters* do not strongly favor any choices ($|\omega(S|A_C) - \omega(S|A_W)| < 0.05$), indicating that they are semantically irrelevant to both candidate choices. However, Table 8 shows that 40.8% of *voters* are logically reasonable, so many *voters* are reasonable but irrelevant to both answers. It suggests that there can be several reasonable answers for a single question, and the sampled *voters* are diverse in the semantics. (3) Although there are only 5.3% of *voters* strongly favoring the correct choices, there are much less *voters* (1.2%) favoring the wrong ones. It explains why our method is able to predict the correct answer.

To help understand the relationship between *voters* and choices, Table 9 provides an instance with *voters* and their voting weights to the choices. We show four types of *voters*: favoring the correct choice, favoring the wrong choice, logically reasonable but not favoring either choices, and unreasonable and irrelevant to both choices. We can see

Q: The car ran out of gas. What happened as a result? A_C: The driver was stranded on the road. (✓) A_W: The driver picked up a hitchhiker. (✘)		
$\omega(S_i\|A_C)$	*voter*	$\omega(S_i\|A_W)$
0.161	I had to park on a dead end road.	0.008
0.008	We picked up a hitchhiker and she drove us to the diner.	0.137
0.013	We stopped at a gas station.	0.011
0.018	It was time to hit the road again.	0.010

Table 9: An example of *voters* as well as their voting weights. A_C is the correct choice, while A_W is wrong. S_i refers to a *voter*.

that the last two types of *voters* can hardly affect the method's prediction, because their voting weights are much smaller than the first two types of *voters*.

5 Conclusion

We present a semantic-based question answering method, SEQA, which can answer commonsense questions more accurately and robustly in an unsupervised setting. Instead of directly scoring each answer choice, our method focuses on the probability of observing a choice's semantics. In the view of voting, SEQA first generates some plausible answers (*voters*) and then utilizes them to vote for the correct choice by considering the semantic similarity between each choice and each *voter*. Experiment results show that SEQA achieves the best performance on four datasets, and it is remarkably more robust than all the baselines when being attacked by TextFooler.

Acknowledgments

This work was partly supported by the NSFC projects (Key project with No. 61936010 and regular project with No. 61876096). This work was also supported by the Guoqiang Institute of Tsinghua University, with Grant No. 2019GQG1 and 2020GQG0005. This work was also supported by Huawei Noah's Ark Lab.

References

Mostafa Abdou, Vinit Ravishankar, Maria Barrett, Yonatan Belinkov, Desmond Elliott, and Anders Søgaard. 2020. The sensitivity of language models and humans to winograd schema perturbations. In *ACL*, pages 7590–7604.

Pratyay Banerjee and Chitta Baral. 2020. Self-supervised knowledge triplet learning for zero-shot question answering. *CoRR*.

Antoine Bosselut and Yejin Choi. 2019. Dynamic knowledge graph construction for zero-shot commonsense question answering. *CoRR*.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Çelikyilmaz, and Yejin Choi. 2019. COMET: commonsense transformers for automatic knowledge graph construction. In *ACL*, pages 4762–4779.

Joe Davison, Joshua Feldman, and Alexander M. Rush. 2019. Commonsense knowledge mining from pretrained models. In *EMNLP-IJCNLP*, pages 1173–1178.

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

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *ICLR*.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: machine reading comprehension with contextual commonsense reasoning. In *EMNLP*, pages 2391–2401.

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

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, EMNLP*, pages 1896–1907.

Vid Kocijan, Ana-Maria Cretu, Oana-Maria Camburu, Yordan Yordanov, and Thomas Lukasiewicz. 2019. A surprisingly robust trick for the winograd schema challenge. In *ACL*, pages 4837–4842.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. Kagnet: Knowledge-aware graph networks for commonsense reasoning. In *EMNLP-IJCNLP*, pages 2829–2839.

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*.

Kaixin Ma, Filip Ilievski, Jonathan Francis, Yonatan Bisk, Eric Nyberg, and Alessandro Oltramari. 2020. Knowledge-driven data construction for zero-shot evaluation in commonsense question answering. *CoRR*.

Farhad Moghimifar, Lizhen Qu, Yue Zhuo, Mahsa Baktashmotlagh, and Gholamreza Haffari. 2020. Cosmo: Conditional seq2seq-based mixture model for zero-shot commonsense question answering. In *COLING*, pages 5347–5359.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *NAACL*.

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

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller. 2019. Language models as knowledge bases? In *EMNLP-IJCNLP*, pages 2463–2473.

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

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. In *ACL*, pages 4932–4942.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP-IJCNLP*, pages 3980–3990.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S. Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *AAAI*.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2019a. ATOMIC: an atlas of machine commonsense for if-then reasoning. In *AAAI*, pages 3027–3035.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019b. Social iqa: Commonsense reasoning about social interactions. In *EMNLP*, pages 4462–4472.

Vered Shwartz, Peter West, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Unsupervised commonsense question answering with self-talk. In *EMNLP*.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI*, pages 4444–4451.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *NAACL-HLT*, pages 4149–4158.

Alexandre Tamborrino, Nicola Pellicanò, Baptiste Pannier, Pascal Voitot, and Louise Naudin. 2020. Pre-training is (almost) all you need: An application to commonsense reasoning. In *ACL*, pages 3878–3887.

Trieu H. Trinh and Quoc V. Le. 2018. A simple method for commonsense reasoning. *CoRR*.

Peifeng Wang, Nanyun Peng, Filip Ilievski, Pedro A. Szekely, and Xiang Ren. 2020. Connecting the dots: A knowledgeable path generator for commonsense question answering. In *Findings, EMNLP*, pages 4129–4140.

Shuohang Wang, Sheng Zhang, Yelong Shen, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, and Jing Jiang. 2019. Unsupervised deep structured semantic models for commonsense reasoning. In *NAACL-HLT*, pages 882–891.

Dataset	COPA-dev	COPA-test	SCT-dev	SocialIQA-dev	CosmosQA-dev
Number of Examples	500	500	1571	1954	2726
Number of Choices	2	2	2	3	3/4
Question Length (mean, std)	(7.3, 1.8)	(7.1, 1.7)	(35.3, 6.5)	(15.3, 4.4)	(83.0, 24.5)
Choice Length (mean, std)	(5.1, 1.6)	(5.0, 1.5)	(7.4, 2.5)	(3.7, 2.3)	(10.0, 4.3)

Table 10: Statistic information of each dataset. Due to the removal of the choice "None of the above", each instance of CosmosQA may have 3 or 4 answer choices.

A Datasets

The four datasets used in this work are multiple-choice commonsense question answering tasks.

COPA[2] (Roemmele et al., 2011) evaluates the ability of causal reasoning about a certain event, which is expressed in a simple sentence. Each question is accompanied with two candidate choices.

StoryClozeTest (SCT)[3] (Mostafazadeh et al., 2016) requires models to select the reasonable story ending, from two alternatives, conditioned on a description about the story context.

SocialIQA[4] (Sap et al., 2019b) evaluates the reasoning ability on social events. In each example, the question describes a social event and asks models to make some inferences based on the event, such as its cause or effect.

CosmosQA[5] (Huang et al., 2019) is a reading comprehension task. Different from the three datasets above, the examples of CosmosQA have long and complex contexts. The original dataset contains a type of choices "None of the above" to test whether models can identify unanswerable questions. This is not the focus of our work, so we removed such choices.

For COPA, we reported the results on its test set. As the test sets of SCT, SocialIQA and CosmosQA are hidden, for convenience of analysis, we reported the experiment results on their development sets. See Table 10 for statistic information of each dataset.

B Templates for Rewriting Questions

We use the same templates for our method and all the baselines. Note that the templates for rewriting questions is not the focus of this paper, and we inherit the templates from previous work if available.

Tamborrino et al. (2020) provides templates for **COPA** (Table 11) and Shwartz et al. (2020) provides templates for **SocialIQA** (Table 12). Since the instances in **SCT** have no questions, **SCT** does not need templates. There is no related work discussing templates for **CosmosQA**, so we design some templates by ourselves (Table 13). **Source code for rewriting questions and SEQA will be made publicly available.**

C Selection of Pre-trained Models

For each method, we tried to adopt different pre-trained models and find the pre-trained models that maximized the accuracy on the development set of each dataset. Table 14 shows the set of candidate pre-trained models for each method, with the selected models in bold. Because of the nature of Pro-Q, it can only use bidirectional language models, so we only evaluated Pro-Q with RoBERTa-large and SentenceRoBERTa-large.

As shown in Table 14, for each method except CGA, the best selection of pre-trained models is consistent on all the datasets. CGA achieved its best performance with COMET on SocialIQA and with GPT2-xlarge on the other datasets.

D Hyperparameter Search

For SEQA, we only tuned the temperature T. To be more specific, we selected T from five candidate values according to the accuracy on the development set of COPA. Table 15 shows that SEQA with $T = 0.1$ achieved the best performance on the development set of COPA. And then we evaluated SEQA with $T = 0.1$ on the test set of COPA as well as the development sets of SCT, SocialIQA and CosmosQA.

E Analysis on Sample Size

Figure 5,6,7 shows the effect of the sample size K on SEQA. For comparison, these figures also include the results of baselines in the settings of before- and after-attack, respectively. On the overall trend, the performance of SEQA improved as

[2]https://people.ict.usc.edu/ gordon/copa.html

[3]https://www.cs.rochester.edu/nlp/rocstories/

[4]https://leaderboard.allenai.org/socialiqa/submissions/get-started

[5]https://leaderboard.allenai.org/cosmosqa/submissions/get-started

Original Question	**Rewrite**
What was the cause of this?	because
What happened as a result?	so
Original Example	**Rewrite**
I saw my breath when I exhaled. What was the cause of this? The weather was chilly.	I saw my breath when I exhaled because the weather was chilly.

Table 11: Templates and a rewritten example of COPA. The templates are inherited from Tamborrino et al. (2020).

Original Question	**Rewrite 1**	**Rewrite 2**
What will [SUBJ] want to do next?	As a result, [SUBJ] wanted to	<xwant>
How would [SUBJ] feel as a result?	As a result, [SUBJ] felt	<xeffect>
What will [SUBJ] do next?	[SUBJ] then	<xreact>
How would you describe [SUBJ]?	[SUBJ] is seen as	<xattr>
Why did [SUBJ] do that?	Before, [SUBJ] wanted	<xintent>
What does [SUBJ] need to do before?	Before, [SUBJ] needed to	<xneed>
Original Example	**Rewrite 1**	**Rewrite 2**
Sydney went trick or treating and the others joined him happily. What will Others want to do next? get candy	Sydney went trick or treating and the others joined him happily. As a result, Others wanted to get candy.	Sydney went trick or treating and the others joined him happily. <xwant> get candy.

Table 12: Some templates and a rewritten example of SocialIQA. [SUBJ] refers to a subject. There are two groups of templates, Rewrite1 for GPT-2 and Rewrite2 for COMET (Bosselut et al., 2019). The relations in Rewrite2 are defined in Sap et al. (2019a) and used for training COMET. These templates are inherited from Shwartz et al. (2020). More details can be found in Shwartz et al. (2020) and https://github.com/vered1986/self_talk.

the sample size increased. Another observation is that a smaller sample size can already make SEQA outperform most baseline methods.

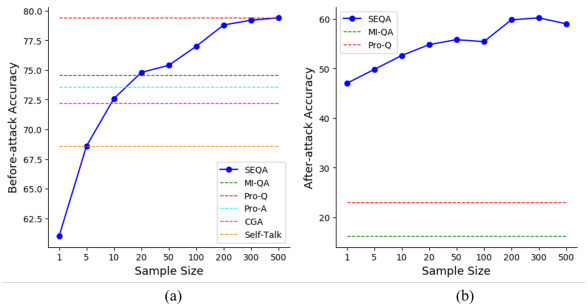


Figure 5: The before-attack (a) and after-attack accuracy (b) of methods with different sample sizes on COPA. The after-attack accuracy of Pro-A, CGA and Self-Talk is below 10.0%, and thus omitted in (b).

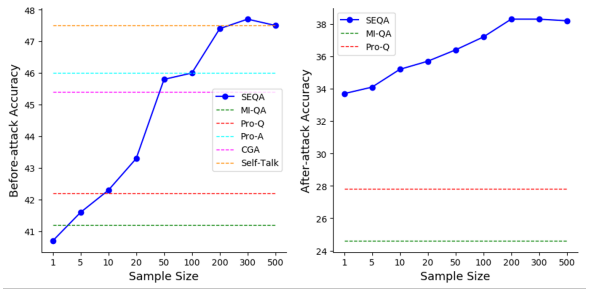


Figure 6: The before-attack (a) and after-attack accuracy (b) of methods with different sample sizes on SocialIQA. The after-attack accuracy of Pro-A, CGA and Self-Talk is below 20.0%, and thus omitted in (b).

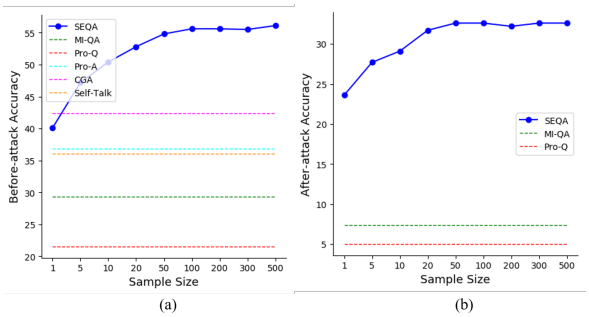


Figure 7: The before-attack (a) and after-attack accuracy (b) of methods with different sample sizes on CosmosQA. The after-attack accuracy of Pro-A, CGA and Self-Talk is below 2.0%, and thus omitted in (b).

Original Question	**Rewrite**
Why [SENTENCE] [CLAUSE] ?	[CLAUSE] [SENTENCE] because
What [NOUN] [SENTENCE] [CLAUSE] ?	[CLAUSE] the [NOUN] [SENTENCE] is that
What [SENTENCE] [CLAUSE] ?	[CLAUSE] it [SENTENCE] that
Original Example	**Rewrite**
... He was conscious but seemed dazed and probably intoxicated . Nearby there was a young man dialing his cell phone . What may happen after the young man makes his call ? An ambulance would likely come to the scene .	... He was conscious but seemed dazed and probably intoxicated . Nearby there was a young man dialing his cell phone . After the young man makes his call , it may happen that an ambulance would likely come to the scene .

Table 13: Templates and a rewritten example of CosmosQA. [NOUN], [SENTENCE] and [CLAUSE] refer to a noun, a sentence fragment and an adverbial clause, respectively.

Method	Set of Candidate Pre-trained Models
Pro-A	LM as QA model: (**GPT2-xlarge**, COMET, RoBERTa-large, SentenceRoBERTa-large)
Pro-Q	LM as QA model: (**RoBERTa-large**, SentenceRoBERTa-large)
MI-QA	LM as QA model: (**GPT2-xlarge**, COMET, RoBERTa-large, SentenceRoBERTa-large)
Self-talk	LM as generator: (GPT2-xlarge, **COMET**) LM as QA model: (**GPT2-xlarge**, COMET, RoBERTa-large, SentenceRoBERTa-large)
CGA	LM as QA model and generator: (**GPT2-xlarge**, **COMET**)
SEQA	LM as generator: (**GPT2-xlarge**, COMET) Feature Extractor: **SentenceRoBERTa-large**

Table 14: The set of candidate pre-trained models. The selected pre-trained models for each method are marked in bold. Note that CGA achieved its best performance with COMET on SocialIQA and with GPT2-xlarge on the other datasets.

T	Dev	Test
10	70.0	75.6
1	70.4	76.4
0.2	71.8	77.0
0.1	**75.4**	79.4
0.05	74.4	80.2

Table 15: Hyperparameter Search of SEQA. The temperature is selected according to the accuracy on the development set of COPA.