When to Read Documents or QA History: On Unified and Selective Open-domain QA

Kyungjae Lee[1*] **Sang-eun Han**[2,3*] **Seung-won Hwang**[2,3†] **Moontae Lee**[1,4]
[1]LG AI Research [2]SNU-LG AI Research Center
[3]Seoul National University [4]University of Illinois at Chicago

Abstract

This paper studies the problem of open-domain question answering, with the aim of answering a diverse range of questions leveraging knowledge resources. Two types of sources, QA-pair and document corpora, have been actively leveraged with the following complementary strength. The former is highly precise when the paraphrase of given question q was seen and answered during training, often posed as a retrieval problem, while the latter generalizes better for unseen questions. A natural follow-up is thus leveraging both models, while a naive pipelining or integration approaches have failed to bring additional gains over either model alone. Our distinction is interpreting the problem as calibration, which estimates the confidence of predicted answers as an indicator to decide when to use a document or QA-pair corpus. The effectiveness of our method was validated on widely adopted benchmarks such as Natural Questions and TriviaQA.

1 Introduction

Open-domain question answering is a well-known task in natural language processing, aiming to answer factoid questions from an open set of domains. One commonly used approach for this task is the retrieve-then-read pipeline (also known as *Open-book QA*) to retrieve relevant knowledge, then reason answers over the knowledge. Given the wide range of topics that open-domain questions can cover, a key to a successful answering model is: to access and utilize diverse knowledge sources effectively.

Toward this goal, existing work can be categorized by the knowledge source used:

- Document Corpus-based QA (**Doc-QA**): This type of work utilizes a general-domain **Document Corpus** (*e.g.*, Wikipedia) (Karpukhin et al., 2020; Guu et al., 2020; Liu et al., 2021; Izacard and Grave, 2021) for reading then answering questions (*i.e.*, $\{Q, D\} \rightarrow A$).
- QA as Retrieval (**QR**): This type of work utilizes a collection of already answered questions (or QA-pair) as knowledge, typically leveraging nonparametric approaches, such as a retriever for closest QA-pairs, to extract the top-1 QA pair that is most similar to a target question and is considered as a final answer (Lewis et al., 2021b; Xiao et al., 2021; Lewis et al., 2021a). *i.e.*, $Q \rightarrow \{\text{paraphrase } Q', A\}$.

In an effort to leverage complementary strengths of existing models, previous work has attempted to build a pipeline of individual models (Lewis et al., 2021b). However, their approach has not resulted in significant gains over using either model alone. In this paper, we propose a novel approach of leveraging the strengths of both document and QA pairs as contexts for a **Unified Reader**-based QA (or **UR-QA**).[1] Figure 1 illustrates the distinction of our approach providing both knowledge to a unified reader as context. We retrieve a list of relevant QA-pairs (called as **QA-history**), then treat the few retrieved QA examples, as if it is a relevant document passage.

Meanwhile, the closest approach to use multiple knowledge sources is concatenating the multi-sources uniformly into a single decoder (Oguz et al., 2020), but we argue **knowledge selection** is critically missing. To motivate, Figure 1 shows the QA-history, from which answer 'Eric Liddell' is explicitly identified, while it is more implicit in the document such that another name such as 'Hugh Hudson' is known to often confuse QA models. It is critical for the QA model to **calibrate** prediction quality as an indicator to decide when to use a

[*]First two authors equally contributed to this work.
[†]correspond to seungwonh@snu.ac.kr

[1]We stress that our focus is a unified framework, and orthogonal to optimizing readers or retrievers, which is beyond the scope of this paper.

Findings of the Association for Computational Linguistics: ACL 2023, pages 6420–6432
July 9-14, 2023

Figure 1: An overview of our Unified Reader QA. We retrieve contexts from document and QA-pair corpus, infer answers from each source, then select the final answer by comparing the calibrated confidences.

document corpus or QA-history.

Toward the goal, we propose Selective QA, where a more reliable answer among candidates can be identified through the calibration of the QA model. Existing calibration (Kamath et al., 2020; Zhang et al., 2021; Si et al., 2022) has focused on the ability of models to "know when they don't know" and abstain from answering if they are uncertain. A naive approach would be simply prioritizing more confident predictions for answer selection.

As a known measure of confidence, LM likelihood of generated tokens has been found to often miscalibrate (Jiang et al., 2021; Kumar and Sarawagi, 2019), tending to prefer short outputs (Murray and Chiang, 2018), or being biased towards more frequent words (Ott et al., 2018). We also observed similar issues in our setting, which we refer to as **calibration overfitting** – LM likelihoods are biased towards increasing confidence on both correct and wrong answers.

Our distinction is to overcome this limitation, by proposing two new objectives, for lowering confidence when the given context cannot answer the question (*i.e.*, answerability), or when sampling uncertainty from decoder is high (*i.e.*, sampling consistency). Finally, building upon improved calibration, we carefully select among answer candidates inferred from document and QA-pairs.

To summarize, we make the following contributions: a) We propose an open-domain QA model complementing document corpus with QA-pair corpus, and decide the selective usage between a document or QA-pair corpus through calibration. b) We evaluate our approach on Natural Questions (Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017), and our method can improve QA performance of existing models. c) We analyze how our method improves calibration and how it helps to select better answers.

2 Related Works

Doc-QA has been a dominant paradigm in open-domain QA (Karpukhin et al., 2020; Guu et al., 2020; Liu et al., 2021; Izacard and Grave, 2021), where the relevant passages are first fetched by the *retriever* model and then processed by the *reader* model to produce the answer. *Reader* models are typically categorized as an *extractive* or *generative* model, where the former locates the answer span in the given context and the latter generates the answer in token-by-token manner. In our work, we focus on a *generative* model, which can transfer knowledge from generative LMs such as T5 (Raffel et al., 2020) and GPT-3 (Brown et al., 2020). Meanwhile, while most works for open-domain QA use Wikipedia as context, some works (Oguz et al., 2020; Ma et al., 2022) leverage various knowledge including Tables and Knowledge Graphs.

QR retrieving relevant QA pairs over a large collection of QA pairs is a more efficient alternative to Doc-QA. Lewis et al. (2021b) build PAQ (for Probably Asked Questions) – 65M QA pairs: automatically-generated resources by using question generation techniques, and learn RePAQ retriever to efficiently extract the top-1 QA pair that is most similar to a target question, and uses its answer for answering the question. Xiao et al. (2021) use answer aggregation heuristic to combine retrieved candidates of QA-pairs with candidates from other sources. Chen et al. (2022) also leverage retrieved QA-pairs, by fusing representations of

the QA-pairs into language models. Despite some gains, their generalizability for unseen questions is limited, compared to Doc-QA, which motivates our approach of selectively combining with other knowledge.

Our Distinction is to analyze and utilize the complementarity of Doc-QA and QR, carefully selecting knowledge sources via calibration, while the previous work (Oguz et al., 2020) blindly concatenates all types of data into a single context.

Calibration has been studied for abstaining from answering when the model does not know. Sources of calibration have been LM's likelihoods (Si et al., 2022), classifier (Kamath et al., 2020), and linguistic expressions (Lin et al., 2022; Mielke et al., 2022; Kadavath et al., 2022; Tian et al., 2023). Our distinction is exploring the use of calibration for selective QA, and overcoming the calibration overfitting we observed from existing methods, by proposing new likelihoods based on answerability and consistency.

3 Proposed Method

In this section, we formally describe Doc-QA as backbones (Section 3.1) and our unification baseline (Section 3.2), followed by our proposed calibration for selective QA (Section 3.3).

3.1 Backbone: Doc-QA

Open-book QA requires to answer question q given context $\mathbf{c}$, *i.e.*, optimizing $\mathrm{P}_{LM}(\mathbf{a}|\mathbf{q},\mathbf{c})$. Doc-QA (Lee et al., 2019; Karpukhin et al., 2020) typically uses Wikipedia documents as knowledge $\mathbf{c}$.

In this paper, for implementing a Doc-QA backbone, we use a state-of-the-art generative reader: Fusion-in-Decoder (Izacard and Grave, 2021), based on a pretrained language model – T5 (Raffel et al., 2020). This approach separately encodes top-n passages in an encoder, and fuses them in a decoder. The final answer A is obtained as follows:

$$\begin{aligned}\mathrm{Fuse}(\mathbf{q},\mathbf{d}_{1:n}) &= [\mathrm{Enc}(\mathbf{q},\mathbf{d}_1);,...,;\mathrm{Enc}(\mathbf{q},\mathbf{d}_n)] \\ \mathrm{A} &= \mathrm{Dec}\left(\mathrm{Fuse}(\mathbf{q},\mathbf{d}_{1:n})\right)\end{aligned} \tag{1}$$

where Enc and Dec indicate Encoder and Decoder modules in transformer (Vaswani et al., 2017), and $[\ ;\ ]$ indicates the concatenation of encoder's outputs. Let $\mathbf{x}$ denote the input sequence, and $\mathbf{y} = (y_1,...,y_T)$ the output sequence. The language model based QA model is trained with maximum likelihood estimation (MLE) to optimize the following objective for a given $(\mathbf{x},\mathbf{y})$:

$$\mathcal{L}(\mathbf{x},\mathbf{y}) = -\sum_{t=1}^{T}\log \mathrm{P}_{LM}(y_t|\mathbf{y}_{<t},\mathbf{x}) \tag{2}$$

where $\mathbf{x}$ is a pair of question/document $(\mathbf{q},\mathbf{d}_{1:n})$, and $\mathbf{y}$ is the ground-truth answer $\mathbf{a}^*$ in our setting. Meanwhile, at inference time, we use Greedy Decoding,[2] which is commonly used for QA tasks. A decoded sequence is $\hat{\mathbf{a}} = (\hat{a}_1,\hat{a}_2,...,\hat{a}_T)$, where each token is selected as follows:

$$\hat{a}_t = \underset{y\in V}{\mathrm{argmax}}\ \mathrm{P}_{LM}(y|\hat{\mathbf{a}}_{<t},\mathbf{q},\mathbf{d}_{1:n}) \tag{3}$$

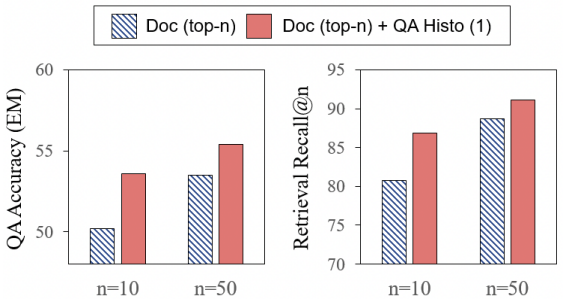


Figure 2: The results of QA (Left) and Retrieval (Right) on NQ.

3.2 Unified Reader: UR-QA

While traditional methods rely on high-efficiency retrievers to match questions with QA history, our work is inspired by *in-context learning* (Brown et al., 2020) for closed-book QA: We propose using the QA-history retrieved as a hypothetical document with few-shot examples and reading it to answer the question

As shown in Figure 1, we retrieve top-n QA pairs from QA corpus as in-context examples, and finetune a QA model with the in-context examples. Specifically, as QA corpus and QR, we used PAQ and a dense retrieval of RePAQ (See Experimental Section for more details), as proposed in Lewis et al. (2021b). Given a target question, we extract top-m QA-pairs from PAQ and the top-m retrieved QA-pairs, as they are short, can be concatenated into one document passage as below:

```
Question: {target q},    Answer:                \n
Question: {example q1}, Answer: {example a1} \n
Question: {example q2}, Answer: {example a2} \n
Question:     ...        Answer:     ...
```

[2] As a decoding method, we can choose beam search or temperature-based sampling, but greedy decoding empirically outperformed others in our QA tasks.

To motivate this approach, Figure 2 shows QA accuracy of our UR-QA and Recall of retrieved knowledge (recall@n) on the following variants of knowledge: (1) Document-only (n passages); (2) Doc + QA history ($n+1$ passages). Gains from adding one passage (concatenating $m = 50$ QA history) suggest the complementary nature of QA history to documents, in terms of both QA and retrieval performances, regardless of the size of retrieved passages n.

Inspired, we propose to combine $\mathbf{d}_{1:n}$ and $\mathbf{k}$ as context, and a baseline (Oguz et al., 2020) concatenates all knowledge – texts, tables, and knowledge graphs in the decoder. Through this "concat" baseline, we can consider $\mathbf{k}$ of QA-pairs as (n+1)th passage in Doc-QA, so that the final answer A_{base} is obtained as follows:

$$A_{base}(\mathbf{q}, \mathbf{d}_{1:n}, \mathbf{k}) = \\ \text{Dec}([\text{Enc}(\mathbf{q}, \mathbf{d}_1); ...; \text{Enc}(\mathbf{q}, \mathbf{d}_n); \text{Enc}(\mathbf{q}, \mathbf{k})]) \quad (4)$$

where $[\,;]$ indicates the concatenation of encoder's outputs. However, due to unreliable inputs from the concatenation, the performance may degrade with increasing noisy context, as reported in Oguz et al. (2020). We hypothesize this as a cause of combining multi-knowledge underperforming a single model and propose selective QA.

3.3 Selective UR-QA via Calibration

Our distinction from concat baseline is that we compare the confidence of each answer from documents and QA history, then select the final answer A_{ours} as follows:

$$A_{ours} = \begin{cases} \hat{\mathbf{a}}_k & \text{if Conf}(\hat{\mathbf{a}}_k|\mathbf{q}, \mathbf{k}) \geq \text{Conf}(\hat{\mathbf{a}}_d|\mathbf{q}, \mathbf{d}) \\ \hat{\mathbf{a}}_d & \text{if Conf}(\hat{\mathbf{a}}_k|\mathbf{q}, \mathbf{k}) < \text{Conf}(\hat{\mathbf{a}}_d|\mathbf{q}, \mathbf{d}) \end{cases} \quad (5)$$

where $\hat{\mathbf{a}}_k$ and $\hat{\mathbf{a}}_d$ are the decoded answers over QA pairs $\mathbf{k}$ and documents $\mathbf{d}$, respectively. While the existing methods for confidence estimation adopt the likelihoods of language models, to overcome its overfitting (Section 3.3.1), we propose two new measures, answerability (Section 3.3.2) and consistency (Section 3.3.3), to eventually ensemble these confidence estimates into a score.

3.3.1 Sequence Likelihood of LM

The key point of our method is to find the effective measurement of the answer confidence, which is essentially the calibration problem. The confidence score $\text{P}(\hat{\mathbf{a}}|\cdot)$ should be able to discern the accurate answer, by comparing the reliability of each knowledge. We propose the way to find such $\text{P}(\hat{\mathbf{a}}|\cdot)$ in the next paragraph, based on our analysis of the important factors on documents and QA-pairs.

Prior work (Hendrycks and Gimpel, 2016) has proposed MaxProb – a method that uses the maximum probability of a classifier as the confidence estimator for selective prediction. For extractive QA, existing works (Zhang et al., 2021; Si et al., 2022) adopt MaxProb as a baseline, by using the sum of the maximum logits of the start and end of the answer span. Meanwhile, we focus on calibrating generative language models, where its output is a token sequence. To apply MaxProb for generative LMs, we select the maximum probability at each step by the argmax function in Eq. (3), which can be viewed as greedy decoding. The scores of decoded tokens are aggregate by product, as follows:

$$\text{P}_{LM}(\hat{\mathbf{a}}|\mathbf{q}, \mathbf{c}) = \prod_{t=1}^{|\hat{\mathbf{a}}|} \text{P}_{LM}(\hat{a_t}|\hat{\mathbf{a}}_{<t}, \mathbf{q}, \mathbf{c}) \quad (6)$$

where $\text{P}_{LM}(*)$ is the token probabilities obtained from LM head. Since LM tends to underestimate the likelihood of longer texts, length normalization is essential as in (Adiwardana et al., 2020). To normalize as sequence lengths,[3] we take the geometric mean of the multiplicative terms, *i.e.*, $\{\text{P}_{LM}(\hat{\mathbf{a}}|\mathbf{q}, \mathbf{c})\}^{1/|\hat{\mathbf{a}}|}$.

However, this LM likelihood obtained by MaxProb has an inevitable problem. MLE loss in Eq. (2) enforces to train LM solely towards maximizing the likelihoods of observed sequences. Because the observed sequences (or labeled answers) can have diverse surface forms, MLE training inevitably leads to miscalibration. In QA tasks, the sequence likelihood of QA models is reported to be often miscalibrated, or overconfident (Jiang et al., 2021; Kumar and Sarawagi, 2019).

In Figure 3, we also observe a consistent tendency in our open-domain QA task, where each line indicates the average confidence score of three estimates on correct predictions (solid line) and incorrect predictions (dashed line). As the training steps increase, the scores of LM likelihood (red lines) increases monotonically, and even the gap between correct and incorrect predictions decreases. We denote this problem as **calibration overfitting**, and hypothesize two causes (C1 and C2).

[3] We empirically found that length normalization slightly improves the performance of Selective QA.

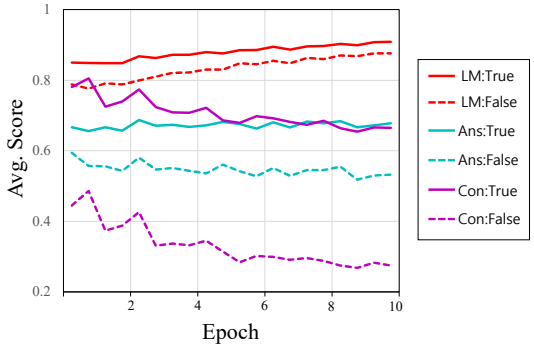


Figure 3: Average scores of three confidence estimates over training epochs. While the solid line is confidence on correct answers, the dashed line is confidence on incorrect answers. Red: LM likelihood, Aqua: Answerability, Purple: Consistency.

- C1: LM's objective maximizes the probabilities on answers regardless if the retrieved context is answerable or not, such that it is overconfident on unanswerable contexts.
- C2: LM likelihood of a decoded output alone does not represent their uncertainty, while candidates unselected by greedy decoding can be a meaningful indicator of uncertainty.

To deal with the above issues, we propose a new calibration approach of learning two measures: **Answerability** and **Consistency**, which are robust to calibration overfitting, as shown in Figure 3.

3.3.2 Answerability

For **C1**, we learn an answerability score, "P(Answerable)", the probability that the passage can answer the given question, which has been studied in Machine Reading Comprehension tasks (Rajpurkar et al., 2018). Our contribution is to train to predict answerability for the question/context pair $(\mathbf{q}, \mathbf{c})$ for the purpose of detecting the low confidence when the given context $\mathbf{c}$ cannot answer question $\mathbf{q}$, *i.e.*, unanswerable. Training signals can be straightforwardly collected by whether $\mathbf{q}$ is answerable in $\mathbf{c}$, or not.

$$\text{P(Answerable)} = \begin{cases} 1, & \text{if } \mathbf{q} \text{ is answerable in } \mathbf{c} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

3.3.3 Consistency

For **C2**, we learn a consistency score, "P(Consistent)", the probability of whether samples consistently match a correct answer. The same decoded answer $\hat{\mathbf{a}}$ may have a high uncertainty, if a discarded candidate from the decoder is also highly plausible. In contrast, the same answer has low uncertainty, if discarded candidates from the decoder are not plausible.

To estimate such sampling uncertainty, we apply sampling-based decoding (temperature=1) generating a set of samples of size N, and measure sampling consistency. More formally, our supervision signal for uncertainty can be collected as:

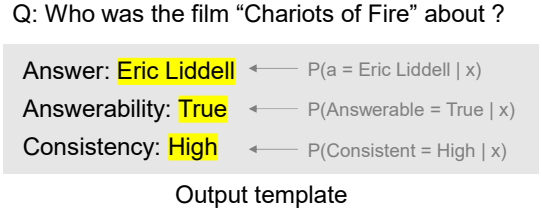
$$\text{P(Consistent)} = \frac{\sum_{i=1}^{N} \mathbb{1}(\hat{\mathbf{a}}_\mathbf{i} = \mathbf{a}^*)}{N} \tag{8}$$

where $\mathbb{1}()$ is 1 if the condition holds (0 otherwise). $\hat{\mathbf{a}}_i$ and $\mathbf{a}^*$ are i-th sampled output and the ground-truth, respectively. N is the number of samples, and we set $N = 30$ in our experiment.

3.3.4 Prompted Calibration

We then proceed to discuss the process of aggregating calibration components into a score, using LM for weak supervision. LM has been used as a means of estimating scores by verbally expressing to estimate a score as an output sequence, as adopted in diverse cases, *e.g.*, sensibleness and safety (Thoppilan et al., 2022) and uncertainty as question types (Lin et al., 2022). The advantages of using a LM-based verbal estimator are twofold: (1) it eliminates the need to construct separate networks for scoring and (2) it captures the interdependency between answer prediction and its uncertainty within the same LM head.

To learn S_{ans} and S_{con} via verbal estimator, we convert the scores into discrete words. Specifically, S_{ans} is expressed as either *True* or *False*. The continuous values S_{con} in training data are sorted and partitioned into equally sized quantiles (*i.e.*, *High*, *Medium*, and *Low*). Then, we train UR to generate the output template, prompted with the verbalized scores, as follows:



Output template

After training with the prompt, we can estimate S_{ans} and S_{con} on test examples, through the likeli-

hood of token "True" or "High", as follows:

$$\begin{aligned}\text{P(Answerable)} &= \text{P}_{LM}(\text{True}|\mathbf{y}_{<True}, \mathbf{q}, \mathbf{c}) \\ \text{P(Consistent)} &= 1 \cdot \text{P}_{LM}(\text{High}|\mathbf{y}_{<High}, \mathbf{q}, \mathbf{c}) \\ &+0.5 \cdot \text{P}_{LM}(\text{Medium}|\mathbf{y}_{<Medium}, \mathbf{q}, \mathbf{c})\end{aligned} \tag{9}$$

where $\mathbf{x}$ is the context with the above prompt. At inference time, we can use a calibration ensemble by averaging the three scores:

$$\begin{aligned}\text{Conf}(\hat{\mathbf{a}}|\mathbf{q}, \mathbf{c}) = \frac{1}{3}\Big(&\text{P}_{LM}(\hat{\mathbf{a}}|\mathbf{q}, \mathbf{c}) \\ &+ \text{P(Answerable)} + \text{P(Consistent)}\Big)\end{aligned} \tag{10}$$

This final confidence is used in Eq. (5) for comparing two candidates, then it decides the final answer.

4 Experiment

In our experiments, we first demonstrate that our proposed confidence scores effectively improve the calibration for question answering. We then examine how these scores contribute to an overall improvement in question answering performance. Finally, we provide qualitative analysis to gain a deeper understanding and insight on our method.

Datasets We use the open-domain QA version of Natural Questions (Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017), following the previous setting (Karpukhin et al., 2020; Izacard and Grave, 2021).[4] The details of the benchmarks are as follows:

- **Natural Questions (NQ)** contains real user questions from Google search engine. We use training/dev/testing splits for open-domain question answering, consisting of 79K train, 8.7k dev, 3.6K test examples.
- **TriviaQA (TQA)** is constructed from web-scraped trivia questions. We use TriviaQA open-domain training/dev/testing splits, consisting of 79K train, 8.8k dev, and 11K test examples.

Implementation We implement our models upon T5 with the size of 770M (or 'Large') and 3B 3B (or 'XL'), and fine-tune them on NQ and TQA. To retrieve the contexts ($\mathbf{d}$ and $\mathbf{k}$), we use the same off-the-shelf retrieval as used by baselines: FiD-KD (Izacard and Grave, 2020) for DR, and RePAQ (Lewis et al., 2021b) for QR. While FiD-KD set the number of passages to 100, we used top-50 passages for DR-QA due to GPU limitations, which is the reason why our DR-QA performed lower than FiD-KD. For QA-history, we concatenate top-50 QA-pairs into a single passage. We use 8 Tesla A100 40GB GPUs for all experiments.

[4]https://github.com/facebookresearch/FiD

Metric	Documents		QA-Pairs	
	NQ	TQA	NQ	TQA
Top-1	50.9	56.9	41.7	41.3
Top-5	75.1	80.2	53.5	51.2
Top-10	80.8	84.8	58.5	55.7
Top-30	86.8	88.6	64.5	61.4
Top-50	88.7	89.7	67.2	64.0

Table 1: Retrieval accuracy on test sets in NQ and TQA.

To retrieve the contexts ($\mathbf{d}$ and $\mathbf{k}$), we use the same off-the-shelf retrieval as used by baselines: FiD-KD (Izacard and Grave, 2020) for Doc-QA, and RePAQ (Lewis et al., 2021b) for QR. For a collection of knowledge, we also use PAQ database for QA pairs (Lewis et al., 2021b), and Wikipedia for documents (Karpukhin et al., 2020). Table 1 shows the accuracy of retrievals from documents and QA-pairs. If a correct answer is included in the top-*K* contexts, the retrieval is assumed to succeed. While this measure calculated by naive string matching is commonly used in (Karpukhin et al., 2020; Izacard and Grave, 2021, 2020), it is not perfect as false negative examples can be counted as true positive.

Baselines To show the effectiveness of our method, we compare previous models over a single source – FiD (Izacard and Grave, 2021), FiD-KD (Izacard and Grave, 2020), UnitedQA (Cheng et al., 2021), and R2-D2 (Fajcik et al., 2021) over documents, and RePAQ (Lewis et al., 2021b) over QA-pairs. "Our backbone" is reimplemented from FiD-KD, while the difference is the number of retrieved documents. To validate the complementary of documents and QA-history, we compare UR-QA on a single source without our selection: "Document Only" and "QA-History Only". As baselines over multiple sources, we compare our method with "Base1: Pipeline" consisting of RePAQ and FiD (Lewis et al., 2021b), and "Base2: Concat" in Eq. (4), inspired by (Oguz et al., 2020).

Main results Table 2 shows the performance of our models, with comparable other models in NQ and TQA. We evaluate the performance of our models by Exact Match (EM) score, which is a stan-

Method	NQ	TQA
Document-based QA		
RAG	44.5	56.8
UnitedQA	54.7	70.5
R2-D2	55.9	69.9
FiD (n=100, Large)	51.4	67.6
FiD-KD (n=100, Large)	54.4	72.5
Our backbone (n=50, Large)	53.4	71.4
QA as Retrieval		
TF-IDF	22.2	23.5
RePAQ (Retriever only)	41.7	41.3
RePAQ (Reranker)	47.6	52.1
UR-QA (on a single source)		
Document Only (n=10, Large)	50.7	69.2
Document Only (n=50, Large)	53.5	71.3
Document Only (n=50, XL)	56.0	73.5
QA-History Only (Large)	46.6	54.3
QA-History Only (XL)	47.7	56.8
UR-QA (Document + QA-History)		
Base1: Pipeline (Large)	52.3	67.3
Base2: Concat (Large)	53.9	72.0
Base2: Concat (XL)	56.7	74.2
Ours: SelectiveQA (n=10+1, Large)	53.6	70.6
Ours: SelectiveQA (n=50+1, Large)	55.4	72.6
Ours: SelectiveQA (n=50+1, XL)	**58.2**	**74.5**

Table 2: Comparison to open-domain QA models on NQ and TQA. Note that while FiD and FiD-KD use 100 documents, we use 10 or 50 documents for ours.

dard metric for open domain question answering (Izacard and Grave, 2021). Our models outperform the baseline models for both datasets and in both model sizes (Large and XL readers). In NQ, we observe that our selective UR-QA achieved the performance gain of 1.9 EM over UR-QA ("Document Only"), and 8.8 over UR-QA ("QA-History Only"), on T5-Large. Our method (Large-NQ) also outperforms Base1: Pipeline (Lewis et al., 2021b) by 2.9 and Base2: Concat by 1.5, respectively. Our best model with larger size (XL) shows **58.2** EM in NQ, which is the highest among the compared models. Meanwhile, our model trained on TQA (Large-TQA) increases EM score by 0.9 over UR-QA ("Document Only") baseline, and 17.9 over UR-QA ("QA-History Only"). Our best performing model in TriviaQA (XL-TQA) achieves the highest score as well, recording **74.5** EM.

Does our method improve calibration for open-domain QA? We use two metrics for the evaluation of the calibration performance: Expected Calibration Error (ECE) and Area Under Curve (AUC) of the risk-coverage graph. ECE is one of the most commonly used metric in previous works

Method	NQ		TQA	
	ECE$_\downarrow$	AUC$_\downarrow$	ECE$_\downarrow$	AUC$_\downarrow$
FiD-KD (LM likeli)	0.310	0.251	0.186	0.103
+Temp Scaling	0.246	0.247	0.063	0.098
UR (DOC-ONLY)				
(1) LM likelihood	0.305	0.290	0.182	0.091
(2) Answerability	0.154	0.307	0.185	0.116
(3) Consistency	**0.134**	0.244	**0.154**	0.099
(1+2+3) Ours	0.163	**0.240**	0.168	**0.088**
UR (QA-ONLY)				
(1) LM likelihood	0.396	0.390	0.326	0.209
(2) Answerability	**0.126**	0.293	0.174	0.188
(3) Consistency	0.153	0.298	**0.074**	0.174
(1+2+3) Ours	0.147	**0.289**	0.170	**0.171**

Table 3: Calibration Evaluation: ECE & AUC of our methods, compared to FiD-KD. ↓ means the lower the metric, the better the calibration is.

(Guo et al., 2017; Minderer et al., 2021; Si et al., 2022; Jiang et al., 2021), which indicates how much the expected accuracy deviates from the expected confidence score. We use the density-based ECE from Minderer et al. (2021), defined as below:

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{1}{M} |\mathrm{Acc}(B_m) - \mathrm{Conf}(B_m)|, \quad (11)$$

where M is the total number of bins (we use $M = 10$), B_m denotes m-th bucket, Acc(B_m) is the mean accuracy of B_m, and Conf(B_m) is the mean confidence. In density-based ECE, an equal number of predictions are assigned to each bin.

On the other hand, the risk-coverage plot (Wang et al., 2017) shows the trade-off between the coverage and risk, where the former is measured as the fraction of test cases that model makes prediction on, and the latter is the error rate (or $1-$accuracy) at that coverage. Specifically, the risk is reportedly high when the coverage increases (El-Yaniv et al., 2010), since the less confident examples come into consideration. Lower AUC of risk-coverage plot indicates the lower average risk, which means more chance of retaining correct answers in selectiveQA.

Table 3 shows that our method (1+2+3) has the lowest AUC in all observed cases. Ours robustly outperforms individual measures in AUC, while there is no 'all-time winner' among individual measures. The robustness of our method is observed in ECE as well – ours is the second-lowest in all cases, while the ranking of others shifts with the change of the dataset or knowledge source. Meanwhile, we attempted temperature scaling (Guo et al., 2017) by

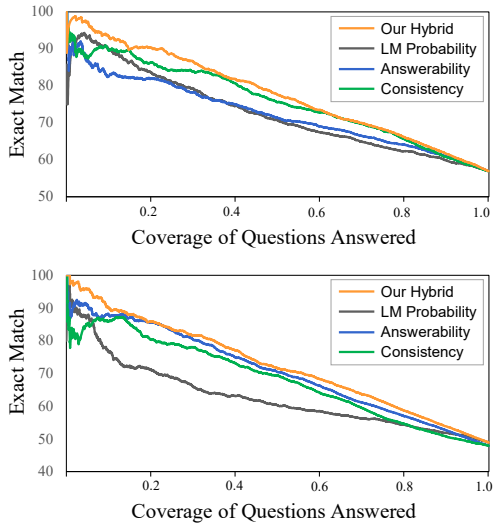


Figure 4: Accuracy-coverage plot (NQ Large). Ours retains the highest accuracy for all coverage. Top: UR-QA (Doc Only), Bottom: UR-QA (QA-History Only).

optimizing a scaling factor in [0,10], but observed no significant improvement on AUC.

Figure 4 provide a finer-grained illustration of this situation, where our hybrid (1+2+3) has the best accuracy (Exact Match) for all coverage in both documents and QA pairs, while the accuracy of other measures fluctuates beneath it.

Does better calibration improve the complementarity of two knowledge sources? Our goal is to enhance the complementarity of documents and QA-history through better calibration, leading to improved QA performance. We investigate if improved calibration truly contributes to the utilization of complementarity. As seen in Table 4, our hybrid (1+2+3) method, which exhibits the best calibration performance in Figure 4, proves to be the most effective criterion for selection, while language model likelihood often fails to improve QA performance beyond the baseline. To examine the upper bound of our approach, we also report the ideal QA performance ('Oracle') which is attainable with the perfect selection. The results indicate that there is a significant potential for complementarity to further enhance QA performance, and that the selection method plays a crucial role in realizing this potential gain.

Is ours robust under domain shifts? To ensure that our model is robust under domain shifts, we conducted cross-evaluation by out-of-domain evaluations: evaluating our QA model (trained on the NQ dataset) on the TQA test set and our QA model (trained on the TQA dataset) on the NQ test set. As shown in Table 5, we found that utilizing both knowledge sources is more beneficial than using a single source, even under domain shifts. Our proposed selective UR achieved gains of 3.8 EM on the NQ dataset and 2.1 EM on the TQA dataset, compared to baselines that used a single source.

Size	Method	NQ	TQA
Large	Ours: (1) LM likelihood	52.2	70.4
	(1) + Temp Scaling	52.1	70.5
	Ours: (2) Answerability	55.1	71.7
	Ours: (3) Consistency	54.9	72.4
	Ours: (1+2+3)	**56.0**	**72.8**
	Oracle - Upper Bound	62.7	75.5
Xlarge	Ours: (1) LM likelihood	54.5	73.5
	(1) + Temp Scaling	54.5	73.6
	Ours: (2) Answerability	57.6	74.2
	Ours: (3) Consistency	57.0	74.3
	Ours: (1+2+3)	**58.1**	**74.7**
	Oracle - Upper Bound	64.6	77.5

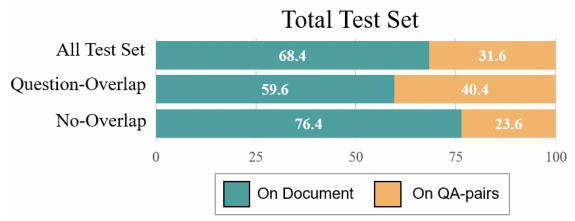
Table 4: Ablation Study

Method	Train on TQA Eval on NQ	Train on NQ Eval on TQA
UR (Doc-only)	34.1	59.9
UR (QA-only)	35.2	49.0
Selective UR	39.0	62.0

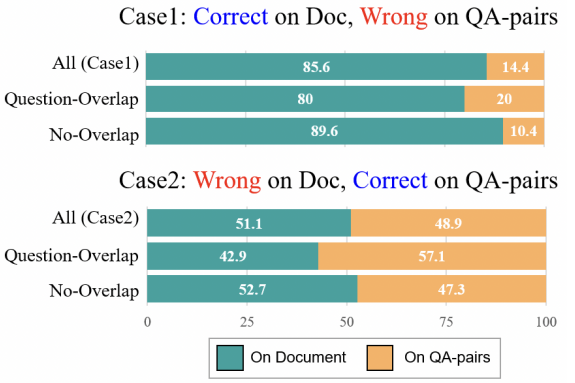
Table 5: Results under domain shift

Model's Selection Ratio We remark our model's behavior that is related to the generalization. Previous work (Lewis et al., 2021a) splits test set into paraphrased questions in training set ("Question-Overlap"), and unseen questions ("No-overlap"). On the divided sub sets, we observe which knowledge (either documents or QA pairs) our method selected. Figure 5 shows the selection ratio of on total test set and Question-overlap/No-overlap sets. As shown in Figure 5 (a), our method tends to select document knowledge (68.4% on all test set). On the question-overlap set, the ratio of selecting QA-pair knowledge increased on the Question-overlap set (31.6% $\rightarrow$ 40.4%). This means the tendency of selecting QA pair knowledge increased when knowledge matching with questions in training set. In contrast, on the no-overlap set, the tendency of selecting documents increased (68.4% $\rightarrow$ 76.4%), which means reading documents is more preferred for generalization on unseen questions.

For a closer look, we select only *critical cases*



(a) The results on all test set



(b) The results on critical cases 1&2

Figure 5: Selection ratio of each knowledge source, from the result of NQ large model.

where only one of the candidate answers is correct – Case1: the answer from documents is correct, but one from QA-history is wrong, and Case2: one from documents is wrong, but one from QA-history is correct. As shown in Figure 5(b), in Case1, document is the majority of the selection, which increases the complementarity of the two knowledge. Meanwhile, in Case2, the ratio of selecting documents (51.1% on all Case2) is the error rate, which is potential room for improvement in our selection.

5 Acknowledgement

This work was supported by LG AI Research. This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) [NO.2021-0-01343, Artificial Intelligence Graduate School Program (Seoul National University)].

6 Conclusion

This paper studies the selective QA system leveraging both document and QA-pair corpus. For careful selection, we propose a novel and effective calibration method based on Answerability and Sampling Consistency and leverage them for comparing and selecting two knowledge sources. On two benchmarks: NQ and TQA, we empirically show our proposed methods outperform existing approaches for open-domain question answering tasks.

7 Limitations

We have identified several limitations in our work and propose future directions to improve them:

(i) The sources for UR-QA in this paper are limited to the document corpus and QA-history, but our unified reader is not restricted to take specific sources. Further research can explore the generalizability of UR-QA to more diverse sources, such as linearized knowledge sources as proposed in (Oguz et al., 2022). Future work can also explore the optimal method for considering LM likelihood, answerability, and consistency together.

(ii) Though it is not the focus of this work to optimize readers, our proposed UR-QA can orthogonally benefit from improvement in retrieval. Further study on the retrieval for UR-QA can be conducted, including the direction to co-optimize the reader and retriever as proposed in (Izacard and Grave, 2020).

References

Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. 2020. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Wenhu Chen, Pat Verga, Michiel de Jong, John Wieting, and William Cohen. 2022. Augmenting pre-trained language models with qa-memory for open-domain question answering. *arXiv preprint arXiv:2204.04581*.

Hao Cheng, Yelong Shen, Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2021. Unitedqa: A hybrid approach for open domain 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 3080–3090.

Ran El-Yaniv et al. 2010. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11(5).

Martin Fajcik, Martin Docekal, Karel Ondrej, and Pavel Smrz. 2021. R2-d2: A modular baseline for open-domain question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 854–870.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *ICML*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International Conference on Machine Learning*, pages 3929–3938. PMLR.

Dan Hendrycks and Kevin Gimpel. 2016. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*.

Gautier Izacard and Edouard Grave. 2020. Distilling knowledge from reader to retriever for question answering. *arXiv preprint arXiv:2012.04584*.

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

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield Dodds, Nova DasSarma, Eli Tran-Johnson, et al. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

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

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

Aviral Kumar and Sunita Sarawagi. 2019. Calibration of encoder decoder models for neural machine translation. *arXiv preprint arXiv:1903.00802*.

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

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

Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. 2021a. Question and answer test-train overlap in open-domain question answering datasets. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1000–1008.

Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021b. Paq: 65 million probably-asked questions and what you can do with them. *Transactions of the Association for Computational Linguistics*, 9:1098–1115.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *arXiv preprint arXiv:2205.14334*.

Ye Liu, Kazuma Hashimoto, Yingbo Zhou, Semih Yavuz, Caiming Xiong, and S Yu Philip. 2021. Dense hierarchical retrieval for open-domain question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 188–200.

Kaixin Ma, Hao Cheng, Xiaodong Liu, Eric Nyberg, and Jianfeng Gao. 2022. Open domain question answering with a unified knowledge interface. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1605–1620.

Sabrina Mielke, Arthur Szlam, Emily Dinan, and Y-Lan Boureau. 2022. Reducing conversational agents' overconfidence through linguistic calibration. *Transactions of the Association for Computational Linguistics*, 10:857–872.

Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. 2021. Revisiting the calibration of modern neural networks. *Advances in Neural Information Processing Systems*, 34:15682–15694.

Kenton Murray and David Chiang. 2018. Correcting length bias in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 212–223.

Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Scott Yih. 2020. Unik-qa: Unified representations of structured and unstructured knowledge for open-domain question answering. *arXiv preprint arXiv:2012.14610*.

Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Scott Yih. 2022. UniK-QA: Unified representations of structured and unstructured knowledge for open-domain question answering. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1535–1546, Seattle, United States. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 1–9.

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

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.

Chenglei Si, Chen Zhao, Sewon Min, and Jordan Boyd-Graber. 2022. Re-examining calibration: The case of question answering. *Findings of Empirical Methods in Natural Language Processing*.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D. Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. *arXiv:2305.14975*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

William Wang, Angelina Wang, Aviv Tamar, Xi Chen, and Pieter Abbeel. 2017. Safer classification by synthesis. *arXiv preprint arXiv:1711.08534*.

Jinfeng Xiao, Lidan Wang, Franck Dernoncourt, Trung Bui, Tong Sun, and Jiawei Han. 2021. Open-domain question answering with pre-constructed question spaces. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 61–67.

Shujian Zhang, Chengyue Gong, and Eunsol Choi. 2021. Knowing more about questions can help: Improving calibration in question answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1958–1970.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
7

☒ A2. Did you discuss any potential risks of your work?
Left blank.

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

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

B ☑ Did you use or create scientific artifacts?

4

☑ B1. Did you cite the creators of artifacts you used?
4

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
4

☑ 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)?
4

☒ 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?
We used well-known public benchmarks, NQ and TQA. In the data, there exist the names of public people as answers , but it does not violate privacy.

☒ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Because NQ and TQA are famous benchmarks, we refer to citation information.

☑ 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.
4

C ☑ Did you run computational experiments?

4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
4

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?
4

☒ 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?
We follow the convention of the previous works.

☐ 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.)?
Not applicable. Left blank.

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

Left blank.

☐ 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.?
No response.

☐ 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)?
No response.

☐ 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?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.