Towards Robust QA Evaluation via Open LLMs

Ehsan Kamalloo∗
ekamalloo@uwaterloo.ca
University of Waterloo
Waterloo, Canada

Shivani Upadhyay∗
sjupadhyay@uwaterloo.ca
University of Waterloo
Waterloo, Canada

Jimmy Lin
jimmylin@uwaterloo.ca
University of Waterloo
Waterloo, Canada

ABSTRACT

Instruction-tuned large language models (LLMs) have been shown to be viable surrogates for the widely used, albeit overly rigid, lexical matching metrics in evaluating question answering (QA) models. However, these LLM-based evaluation methods are invariably based on proprietary LLMs. Despite their remarkable capabilities, proprietary LLMs are costly and subject to internal changes that can affect their output, which inhibits the reproducibility of their results and limits the widespread adoption of LLM-based evaluation. In this demo, we aim to use publicly available LLMs for standardizing LLM-based QA evaluation. However, open-source LLMs lag behind their proprietary counterparts. We overcome this gap by adopting chain-of-thought prompting with self-consistency to build a reliable evaluation framework. We demonstrate that our evaluation framework, based on 750M and 7B open LLMs, correlates competitively with human judgment, compared to most recent GPT-3 and GPT-4 models. Our codebase and data are available at https://github.com/castorini/qa-eval.

CCS CONCEPTS
• Information systems → Question answering; Evaluation of retrieval results.

KEYWORDS
Question Answering, Evaluation, Large language models

1 INTRODUCTION

Evaluating question answering (QA) models requires matching candidate answers with a set of predefined gold answers. This type of answer equivalence matching is often done based on lexical matching [31]. Despite its widespread adoption and simplicity, lexical matching suffers from fundamental flaws, mostly rooted in diverse forms of plausible answers not present in the gold answers [7, 8, 27]. For instance, if the gold answer is the year “1689”, “17th century” may also be acceptable, but cannot be captured by lexical matching. These flaws substantially undermine evaluation reliability [20]. Luckily, instruction-tuned large language models (LLMs) are found to be promising alternatives for the evaluation of QA models [1, 20]. Yet, this success is all centered around proprietary LLMs such as OpenAI’s GPT-3 [6, 29] and GPT-4 [28]. Notwithstanding their remarkable capabilities, regular opaque changes to proprietary LLMs [10] or the possibility of their discontinuation1 inhibits the reproducibility of these findings. Furthermore, proprietary LLMs come with associated expenses, thus impeding their broad adoption in evaluation, especially on large-scale datasets. Therefore, achieving trustworthy and deterministic evaluation demands the use of fully open-source models that are widely accessible.

In this demo, we aim to bridge this gap by introducing a standardized QA evaluation framework to make LLM-based automated evaluation widely accessible. Inspired by trec_eval2 in information retrieval, our main goal is to unify evaluation by presenting our LLM evaluation widely accessible. Inspired by trec_eval.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGIR '24, July 14–18, 2024, Washington, DC, USA
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0431-4/24/07
https://doi.org/10.1145/3626772.3657675

∗Equal Contribution

1A family of GPT-3 models including text-davinci-003 was deprecated in Jan’24: https://openai.com/blog/gpt-4-api-general-availability.

2https://github.com/usnistgov/trec_eval

Figure 1: An illustration of our evaluation framework, consisting of two steps: (1) Prompt preparation using the inputs (question, gold answers, and candidate answer for evaluation). We use a few examples with chain-of-thought reasoning to guide the LLM. (2) Generating multiple responses and carrying out majority voting over the judgments to obtain the final judgment (self-consistency). In the example above, although the gold answer is not accurate, our LLM evaluator was able to provide a correct evaluation.
evaluation tool to the community. One major obstacle in achieving this goal is that open LLMs [4, 5, 37, 44] are known to lag behind proprietary LLMs on many benchmarks [13, 24]. Moreover, the downsized scale of LLMs that can be run on commodity hardware may not be strong enough in that LLM capabilities become more powerful at a larger scale [14, 21, 40]. We overcome these challenges via two simple strategies in prompting and generation (Figure 1). First, we follow chain-of-thought (CoT) prompting [41] to guide the model to explain its output before making its judgment. However, it is challenging to convey all the intricacies of evaluation through explanation in a few examples, which contributes to reasoning errors in LLM evaluation. To fix this issue, we adopt self-consistency [39] to sample multiple explanations and obtain the final evaluation based on a majority vote.

To test the reliability of our proposed method, we examine several open instruction-tuned LLMs for QA evaluation and measure their correlation with human judgment. We find that despite their smaller size, open-source LLMs demonstrate competitive effectiveness, compared to their proprietary counterparts.

Our evaluation framework aims to standardize QA evaluation using open-source LLMs. We hope our effort fosters robust evaluation and provides the essential means to reliably gauge progress in QA. Our key contributions can be summarized as follows:

- We introduce a fully open-source QA evaluation tool to unify the evaluation of QA models.
- We develop LLM-based evaluation techniques based on CoT prompting and self-consistency to bolster reliability.
- Our framework, based on smaller-scale LLMs that can be run on one GPU, is competitive with GPT-3 and GPT-4.

2 QA EVALUATION USING LLMS

The task of answer equivalence in QA evaluation is usually done using lexical matching metrics: Exact-Match (EM) and $F_1$ [31]. Different variants of n-gram matching [3, 25, 30] have also been used in QA. More recently, evaluation is framed as semantic similarity, either supervised [7, 9, 33] or unsupervised [45]. Another line of work focuses on augmenting QA datasets using external sources to enrich the list of gold answers [35]. With the rise of LLMs [6], evaluation can be done by simply eliciting a prompt from an LLM [1, 20]. Many studies [27, 34] employ humans for accurate and reliable evaluation. However, human judgment is not cost-effective and difficult to scale for large datasets. This work builds on using LLMs for automated evaluation, aiming to standardize QA evaluation using open-source LLMs.

Our main idea to use LLMs for QA evaluation is to insert both gold answers and candidate answers in the prompt and instruct the model to verify whether candidate answers are acceptable. While this approach is previously shown to be effective using proprietary LLMs such as GPT-3 and GPT-4 [1, 20], providing only detailed instructions does not work well for smaller open-source LLMs. To address this gap, we propose two simple strategies, depicted in Figure 1, to make open LLMs robust in QA evaluation. Note that our focus in this paper is on factoid questions where answers are typically expected to be short.

CoT prompting. Judging for QA evaluation can be non-trivial in numerous cases and LLMs may not be able to understand all the nuances of the task solely from the instructions. For this purpose, we provide carefully crafted examples, derived from lexical matching failures in the prompt [1, 20]. However, the final judgment could be confusing without additional explanations. Thus, we use a CoT-style [41] prompting approach to provide explanations for the in-context examples. CoT prompting guides the model to explain its reasoning before concluding its judgment. Our prompt is as follows:

<table>
<thead>
<tr>
<th>Prompt</th>
<th>CoT Prompting</th>
</tr>
</thead>
<tbody>
<tr>
<td>You are an expert judge of a content. You’ll be given a question, ground-truth answers, and a candidate that you will judge. Using your internal knowledge and simple commonsense reasoning, and given the ground-truth answers, try to verify if the candidate is correct or not. The contestant may provide a candidate answer that isn’t an exact match. Your job is to determine if the candidate is correct or not. Provide explanation for the comparison and give your judgment with a “yes” or “no” in a new line. Here, “yes” represents that the candidate answer is relevant and correct based on either inbuilt knowledge or ground-truth answers. If not, the judgment based on the explanation should be “no”.</td>
<td></td>
</tr>
</tbody>
</table>

Examples in prompts are sampled from the NQ-OPEN [23] dev set.

Self-consistency. Even with the provided examples in the prompt, models may still be prone to reasoning errors. For instance, LLMs may generate correct explanations, but arrive at a wrong judgment. These types of errors suggest that the model is already equipped with sufficient knowledge and capabilities to reason about the correctness of a candidate answer but under some circumstances such as an “unlucky” sample during decoding, it fails. As a remedy, we use self-consistency [39] by sampling multiple responses from an LLM, followed by taking a majority vote to determine the outcome. This simple approach ensures that the model selects the most consistent answer, thereby reducing the likelihood of reasoning errors.

3 EXPERIMENTAL SETUP

Our experiments are performed on a subset of 301 questions, derived from NQ-OPEN [23], following Kamalloo et al. [20]. We take generated answers from 12 QA models that fall into two paradigms: closed-book and retriever-reader. In total, we examined 12 QA models taken from Kamalloo et al. [20]: DPR [22], Fusion-In-Decoder (FiD; [18]), Contriever [16], RocketQA v2 [32], FiD-KD [17], ANCE [43], GAR [26], R2-D2 [12], EMDR² [36], EviGen [2], and InstructGPT [29] in two settings: zero-shot and few-shot.

For evaluating these QA models, we experimented with multiple instruction-tuned LLMs:

- Open-source LLMs. We use FLAN-T5-large³ (750M) [11], Mistral-7B-Instruct⁴ [19], and Zephyr-7B⁵ [38]. Our initial experiments showed that LLMs without instruction-tuning such as Llama-2 [37] do not work well here.

- Proprietary LLMs. We use GPT-3.5-turbo (gpt-3.5-1106) and GPT-4-turbo (gpt-4-1106-preview), and GPT-4 (gpt-4-0314) results from Kamalloo et al. [20] as a reference. We do not use

³https://huggingface.co/google/flan-t5-large
⁴https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.1
⁵https://huggingface.co/HuggingFaceH4/zephyr-7b-beta
Table 1: Accuracy of 12 QA models on 301 sampled questions from NQ-open using different evaluation methods: human, lexical matching, zero-shot LLMs, and few-shot LLMs. GPT models are proprietary, whereas FLAN-T5, Mistral, and Zephyr are open-source. Different shades of blue indicate the best, second best, and third best under each evaluation method. * denotes a result taken from Kamalloo et al. [20].

<table>
<thead>
<tr>
<th>Models</th>
<th>Human</th>
<th>Lexical</th>
<th>GPT-4</th>
<th>GPT-4_turbo</th>
<th>GPT-3.5_turbo</th>
<th>Few-shot</th>
</tr>
</thead>
<tbody>
<tr>
<td>InstructGPT (zero-shot)</td>
<td>71.4</td>
<td>12.6</td>
<td>27.5</td>
<td>68.8</td>
<td>60.5</td>
<td>65.1</td>
</tr>
<tr>
<td>InstructGPT (few-shot)</td>
<td>75.8</td>
<td>33.9</td>
<td>50.5</td>
<td>68.8</td>
<td>68.1</td>
<td>71.1</td>
</tr>
<tr>
<td>DPR</td>
<td>58.8</td>
<td>45.9</td>
<td>52.3</td>
<td>56.5</td>
<td>56.5</td>
<td>58.1</td>
</tr>
<tr>
<td>FiD</td>
<td>64.8</td>
<td>47.8</td>
<td>55.4</td>
<td>61.8</td>
<td>60.4</td>
<td>61.5</td>
</tr>
<tr>
<td>ANCE+ &amp; FiD</td>
<td>65.8</td>
<td>48.2</td>
<td>55.9</td>
<td>62.5</td>
<td>60.6</td>
<td>62.8</td>
</tr>
<tr>
<td>RocketQA v2 &amp; FiD</td>
<td>70.1</td>
<td>49.8</td>
<td>58.7</td>
<td>67.1</td>
<td>64.5</td>
<td>67.8</td>
</tr>
<tr>
<td>Contriever &amp; FiD</td>
<td>66.5</td>
<td>46.5</td>
<td>55.9</td>
<td>64.8</td>
<td>61.9</td>
<td>64.1</td>
</tr>
<tr>
<td>FiD-KD</td>
<td>73.1</td>
<td>50.8</td>
<td>61.2</td>
<td>69.4</td>
<td>67.4</td>
<td>69.4</td>
</tr>
<tr>
<td>GAR+ &amp; FiD</td>
<td>69.4</td>
<td>50.8</td>
<td>59.7</td>
<td>67.4</td>
<td>64.1</td>
<td>65.5</td>
</tr>
<tr>
<td>EviGen</td>
<td>67.1</td>
<td>51.8</td>
<td>59.5</td>
<td>66.1</td>
<td>62.8</td>
<td>64.8</td>
</tr>
<tr>
<td>EMDR2</td>
<td>73.1</td>
<td>53.2</td>
<td>62.6</td>
<td>68.4</td>
<td>73.8</td>
<td>71.8</td>
</tr>
<tr>
<td>R2-D2</td>
<td>71.4</td>
<td>52.8</td>
<td>61.4</td>
<td>65.8</td>
<td>64.5</td>
<td>69.1</td>
</tr>
</tbody>
</table>

Table 2: Spearman and Kendall’s $\tau$ correlations between open-source and proprietary LLMs and human judgment.

Table 2 presents Spearman and Kendall’s $\tau$ correlations with the open-source Zephyr model exhibit a strong correlation with human judgment, although the correlations of the proprietary LLMs are near-perfect. These strong correlations suggest that LLMs are reliable for comparing the effectiveness of QA models.

4 RESULTS

Correlation Results. In Table 1, the accuracy of QA models is reported based on human judgment as well as automated methods, i.e., lexical metrics, zero-shot LLMs, and few-shot LLMs. All QA models consistently demonstrate an increase in accuracy under human judgment and LLM-based evaluation, compared to lexical metrics. Notably, the average absolute error across the QA models is only slightly different between GPT-4_turbo (few-shot) with 2.4%, and the open-source model, Zephyr, with 3.0%.

Table 2 presents Spearman and Kendall’s $\tau$ correlations with human judgment in ranking the QA models. Figure 2 visualizes the correlation for the automated evaluation methods. The few-shot LLM-based evaluation using GPT-3.5_turbo and GPT-4_turbo along with the open-source Zephyr model exhibit a strong correlation with human judgment, although the correlations of the proprietary LLMs are near-perfect. These strong correlations suggest that LLMs are reliable for comparing the effectiveness of QA models.

Error Analysis. We analyze to what extent LLMs are able to amend lexical matching errors. We follow the lexical matching failure modes specified in Kamalloo et al. [20]:

- **Semantic Equivalence**: Model predictions and gold answers express similar meanings without using identical wording, e.g., "3" vs. "three" or "USA" vs. "America".
- **Symbolic Equivalence**: For numerical answers, gold answers and predicted ones could be the same, either precisely or approximately, even though their surface texts are different, e.g., "about 3.99 degrees" vs. "3.97 degrees".
- **Granularity Discrepancies**: When answers include temporal/spatial references, predicted and reference answers may differ in granularity, e.g., "2000" vs. "8 Nov, 2000".
- **Intrinsic Ambiguity in Questions**: Ambiguous questions can be interpreted in several ways, each potentially resulting in different answers, e.g., "When does the new episode of Scorpion come on?"

---

*Based on the rule-of-thumb that correlation greater than 0.8 is typically considered "very strong" in the statistics literature.*
We also evaluate the impact of the decoding algorithm as well as are showcased in Figure 3. We see that LLM-based evaluation methods rectify most errors for three modes: semantic equivalence, symbolic equivalence, and granularity discrepancy.

- **Incomplete Reference Answers**: Acceptable answers consist of a range of plausible answers that are not completely provided in the list of gold answers.
- **Incorrect Reference Answers**: In QA datasets, reference answers are sometimes erroneously labelled, leading to the rejection of actually correct predicted answers.

Results using the failure mode categorizations from Kamalloo et al. [20] are showcased in Figure 3. We see that LLM-based evaluation methods successfully fix most of the failures corresponding to semantic equivalence, granularity discrepancy, symbolic equivalence, and incomplete reference answers, while having limited impact on failures stemming from data quality issues. The gap between proprietary LLMs and open-source LLMs is negligible in all error categories except for semantic equivalence and granularity discrepancy, where the difference is nearly 2%.

**Ablation Study**. To investigate the impact of CoT prompting and self-consistency in evaluating QA models, we conduct an ablation study of our evaluation framework, considering three variants:

1. **No CoT + No Self-Consistency**: Zero-shot prompting and generate \( n = 1 \) response using beam search (beam size=10).
2. **No CoT + Self-Consistency**: Zero-shot prompting and generate \( n \) responses using beam search.
3. **CoT + No Self-Consistency**: Few-shot prompting and generate \( n = 1 \) response using beam search.

We also evaluate the impact of the decoding algorithm as well as the number of generated responses \( (n) \). We compute the ranking correlation of each variant with human judgment.

The results, presented in Table 3, highlight the importance of both CoT and self-consistency in achieving robust evaluation. Another interesting observation is that increasing the number of generated responses \( (n = 9) \) yields modest improvements but at the expense of slower run-time; hence, we opt for \( n = 3 \) by default.

### 5 PACKAGE OVERVIEW

Our evaluation framework is shipped as a Python package and also hosted on GitHub. It can be easily installed as follows:

```bash
$ pip install git+github.com/castorini/QA-eval
```

![Figure 3: Frequency of lexical matching failure modes for each evaluation method. All LLM-based evaluation methods rectify most errors for three modes: semantic equivalence, symbolic equivalence, and granularity discrepancy.](image)

<table>
<thead>
<tr>
<th>Decoding Alg.</th>
<th>Spearman</th>
<th>Kendall</th>
</tr>
</thead>
<tbody>
<tr>
<td>CoT + Self-C.</td>
<td>Beam ( n = 3 )</td>
<td>93.0</td>
</tr>
<tr>
<td>No CoT + No Self-C.</td>
<td>Beam ( n = 1 )</td>
<td>81.6</td>
</tr>
<tr>
<td>No CoT + Self-C.</td>
<td>Beam ( n = 3 )</td>
<td>86.5</td>
</tr>
<tr>
<td>CoT + No Self-C.</td>
<td>Beam ( n = 1 )</td>
<td>87.4</td>
</tr>
<tr>
<td>CoT + Self-C.</td>
<td>Beam ( n = 9 )</td>
<td>93.8</td>
</tr>
<tr>
<td>CoT + Self-C.</td>
<td>Nucleus ( n = 3 )</td>
<td>89.8</td>
</tr>
</tbody>
</table>

Table 3: Ablation analysis of our evaluation framework. Spearman and Kendall’s \( \tau \) correlations of Zephyr judgments under different variants vs. human judgment. Self-C. refers to self-consistency. Decoding algorithms are beam search and nucleus sampling [15]. \( n \) denotes the number of responses, sampled from Zephyr during generation.

We support OpenAI APIs for GPT-3 and GPT-4 models as well as the open-source LLMs we examined in this paper via Huggingface [42]. Our tool offers a simple, unified interface for running QA evaluation via a simple invocation:

```bash
$ python -m qaeval /path/to/prediction.jsonl --model MODEL
```

where MODEL refers to the evaluation model name that can either be a proprietary GPT model or a Huggingface model. Also, it is possible to adjust generation parameters, including the maximum number of tokens to generate, temperature, greedy decoding or sampling, and the number of generated samples. Details are provided in our documentation. System outputs to be evaluated are passed as a jsonl file with the following structure:

```json
{
    "question": "what is the boiling temperature for water",
    "answer": ["212 °F (100 °C)", "prediction": "100 degrees C"
}
```

This package allows researchers to reproduce the results in this paper and to evaluate their own QA systems.

### 6 CONCLUSION

For QA evaluation, the widely used lexical matching technique inherently fails to match semantically similar answers that do not exist within the gold answers. Luckily, instruction-tuned LLMs have proven to be promising alternatives for lexical matching. Nonetheless, existing efforts to leverage LLMs for QA evaluation overwhelmingly rely on opaque, proprietary LLMs. In this work, we introduce an evaluation framework using open LLMs to standardize LLM-based QA evaluation. Our recipe is simple, building on CoT prompting and self-consistency. Our proposed framework, captured in a tool we share with the community, performs competitively with opaque and substantially larger proprietary models.

### ACKNOWLEDGEMENTS

This research was supported in part by the Canada First Research Excellence Fund and the Natural Sciences and Engineering Research Council (NSERC) of Canada. We’d also like to thank Microsoft for providing access to OpenAI LLMs via Azure.
2825–2835.


