# 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

**ACM Reference Format:**
Ehsan Kamalloo, Shivani Upadhyay, and Jimmy Lin. 2024. Towards Robust QA Evaluation via Open LLMs. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), July 14–18, 2024, Washington, DC, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3626772.3657675

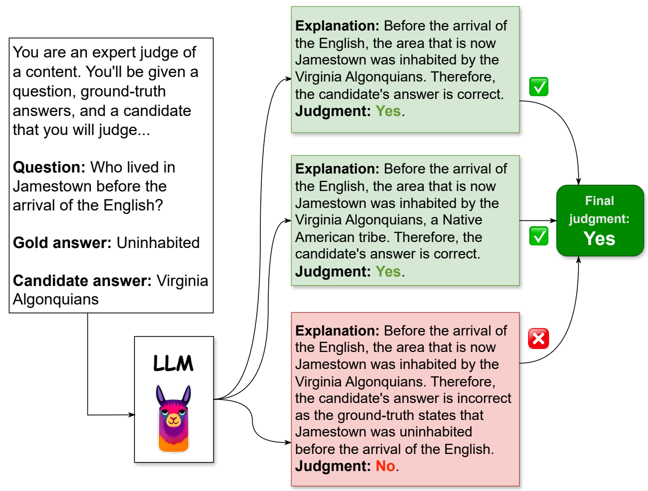


**Figure 1: An illustration of our evaluation framework, consisting of two steps: (1) Prompt preparation using the inputs (question, gold answers, and candidate answer for evaluation). We use a few examples with chain-of-thought reasoning to guide the LLM. (2) Generating multiple responses and carrying 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.**

## 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 discontinuation[1] 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_eval`[2] in information retrieval, our main goal is to unify evaluation by presenting our

*Equal Contribution

[1]A family of GPT-3 models including `text-davinci-003` was deprecated in Jan'24: https://openai.com/blog/gpt-4-api-general-availability.
[2]https://github.com/usnistgov/trec_eval


*SIGIR '24, July 14–18, 2024, Washington, DC, USA*

ACM ISBN 979-8-4007-0431-4/24/07
https://doi.org/10.1145/3626772.3657675

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:

```
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 groundtruth 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".
```

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], RocketQAv2 [32], FiD-KD [17], ANCE [43], GAR [26], R2-D2 [12], EMDR$^2$ [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-large`[3] (750M) [11], `Mistral-7B-Instruct`[4] [19], and `Zephyr-7B`[5] [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$_{\text{turbo}}$ (`gpt-3.5-1106`) and GPT-4$_{\text{turbo}}$ (`gpt-4-1106-preview`), and GPT-4 (`gpt-4-0314`) results from Kamalloo et al. [20] as a reference. We do not use

[3]https://huggingface.co/google/flan-t5-large
[4]https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.1
[5]https://huggingface.co/HuggingFaceH4/zephyr-7b-beta

| Models | Human† | Lexical EM | Lexical $F_1$ | zero-shot GPT-4† | zero-shot GPT-3.5$_{turbo}$ | zero-shot GPT-4$_{turbo}$ | few-shot GPT-3.5$_{turbo}$ | few-shot GPT-4$_{turbo}$ | few-shot FLAN$_{large}$ | few-shot Mistral | few-shot Zephyr |
|---|---|---|---|---|---|---|---|---|---|---|---|
| InstructGPT (zero-shot) | 71.4 | 12.6 | 27.5 | 68.8 | 60.5 | 65.1 | 62.8 | 67.4 | 58.5 | **70.1** | 69.8 |
| InstructGPT (few-shot) | **75.8** | 33.9 | 50.5 | 68.8 | 68.1 | 71.1 | **67.1** | 72.1 | **70.8** | 68.1 | **71.1** |
| DPR | 58.8 | 45.9 | 52.3 | 56.5 | 56.5 | 58.1 | 53.5 | 58.1 | 50.2 | 57.8 | 56.5 |
| FiD | 64.8 | 47.8 | 55.4 | 61.8 | 59.5 | 61.5 | 57.1 | 61.8 | 56.8 | 63.1 | 63.5 |
| ANCE+ & FiD | 65.8 | 48.2 | 55.9 | 62.5 | 60.5 | 62.8 | 58.8 | 63.5 | 57.1 | 63.5 | 63.8 |
| RocketQAv2 & FiD | 70.1 | 49.8 | 58.7 | 67.1 | 64.5 | 67.8 | 61.1 | 68.1 | 59.5 | 65.1 | 65.8 |
| Contriever & FiD | 66.5 | 46.5 | 55.9 | 64.8 | 61.8 | 64.1 | 58.5 | 65.5 | 56.8 | 61.1 | 64.1 |
| FiD-KD | 73.1 | 50.8 | 61.2 | **69.4** | 67.4 | 69.4 | 65.5 | 69.8 | 61.5 | 67.4 | 69.4 |
| GAR+ & FiD | 69.4 | 50.8 | 59.7 | 67.4 | 64.1 | 65.5 | 61.5 | 65.8 | 60.1 | 64.8 | 66.8 |
| EviGen | 67.1 | 51.8 | 59.5 | 66.1 | 62.8 | 64.8 | 61.1 | 65.5 | 57.5 | 63.5 | 63.8 |
| EMDR$^2$ | 73.1 | **53.2** | **62.6** | 68.4 | **73.8** | **71.8** | **67.1** | **72.8** | 58.8 | 68.1 | 67.8 |
| R2-D2 | 71.4 | 52.8 | 61.4 | 65.8 | 64.5 | 69.1 | 64.1 | 68.1 | 61.8 | 68.1 | 69.4 |

**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].**

| | Evaluator | Spearman | Kendall |
|---|---|---|---|
| Lexical | EM | 22.0 | 23.3 |
| | $F_1$ | 30.2 | 36.9 |
| zero-shot | GPT-4 | 90.2 | 79.1 |
| | GPT-4$_{turbo}$ | 95.8 | 89.2 |
| | Zephyr | 86.5 | 69.8 |
| few-shot | GPT-3.5$_{turbo}$ | 97.4 | 90.6 |
| | GPT-4$_{turbo}$ | 97.0 | 90.6 |
| | FLAN$_{large}$ | 86.5 | 72.9 |
| | Mistral | 88.5 | 76.2 |
| | Zephyr | 93.0 | 81.2 |

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

self-consistency for proprietary LLMs because their results are acceptable without self-consistency.

For the experiments, we first rank the QA models utilizing different LLM evaluators. These rankings are then compared with human annotations [20] to compute Spearman and Kendall $\tau$'s correlations to quantify the quality of the evaluation model.

## 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

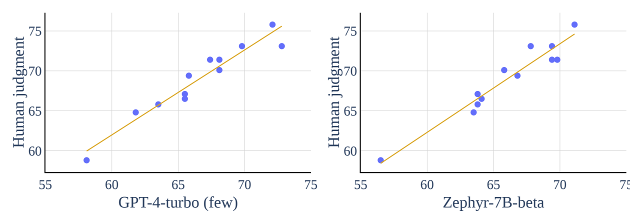


**Figure 2: Scatter plots visualizing the correlation of evaluation methods: GPT-4$_{turbo}$ (left), and Zephyr (right), both few-shot v.s. human judgment.**

with the open-source Zephyr model exhibit a strong correlation with human judgment,[6] 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?*"

[6] Based on the rule-of-thumb that correlation greater than 0.8 is typically considered "very strong" in the statistics literature.

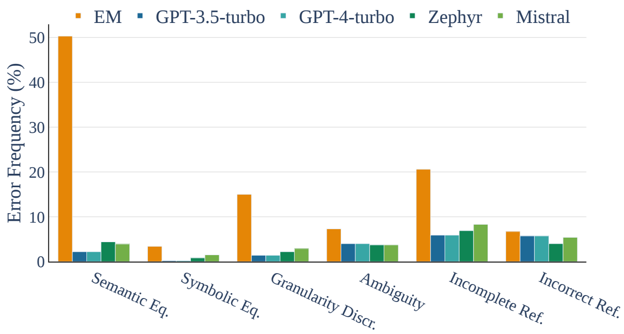


**Figure 3: Frequency of lexical matching failure modes for each evaluation method. All LLM-based evaluation methods rectify most errors for three modes: semantic equivalence, symbolic equivalence, and granularity discrepancy.**

- **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:

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

| | Decoding Alg. | Spearman | | Kendall | |
|---|---|---|---|---|---|
| CoT + Self-C. | Beam $n = 3$ | 93.0 | | 81.2 | |
| No CoT + No Self-C. | Beam $n = 1$ | 81.6 | -11.4↓ | 64.3 | -16.9↓ |
| No CoT + Self-C. | Beam $n = 3$ | 86.5 | -6.5↓ | 69.8 | -11.4↓ |
| CoT + No Self-C. | Beam $n = 1$ | 87.4 | -5.6↓ | 73.9 | -8.3↓ |
| CoT + Self-C. | Beam $n = 9$ | 93.8 | +0.8↑ | 82.2 | +1.0↑ |
| CoT + Self-C. | Nucleus $n = 3$ | 89.8 | -3.2↓ | 79.1 | -2.1↓ |

**Table 3: Ablation analysis of our evaluation framework. Spearman and Kendall's $\tau$ correlations of Zephyr judgments under different variants v.s. 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:

```
$ 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:

```
{
    "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.

# REFERENCES


[1] Vaibhav Adlakha, Parishad BehnamGhader, Xing Han Lu, Nicholas Meade, and Siva Reddy. 2023. Evaluating Correctness and Faithfulness of Instruction-Following Models for Question Answering. (2023). arXiv:2307.16877

[2] Akari Asai, Matt Gardner, and Hannaneh Hajishirzi. 2022. Evidentiality-guided Generation for Knowledge-Intensive NLP Tasks. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Seattle, United States, 2226–2243.

[3] Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Association for Computational Linguistics, Ann Arbor, Michigan, 65–72.

[4] Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Usvsn Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. 2023. Pythia: A Suite for Analyzing Large Language Models across Training and Scaling. In *Proceedings of the 40th International Conference on Machine Learning*, Vol. 202. PMLR, 2397–2430.

[5] Sidney Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, Usvsn Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. 2022. GPT-NeoX-20B: An Open-Source Autoregressive Language Model. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*. Association for Computational Linguistics, virtual+Dublin, 95–136.

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, Vol. 33. Curran Associates, Inc., 1877–1901.

[7] Jannis Bulian, Christian Buck, Wojciech Gajewski, Benjamin Börschinger, and Tal Schuster. 2022. Tomayto, Tomahto. Beyond Token-level Answer Equivalence for Question Answering Evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 291–305.

[8] Anthony Chen, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Evaluating Question Answering Evaluation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*. Association for Computational Linguistics, Hong Kong, China, 119–124.

[9] Anthony Chen, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2020. MOCHA: A Dataset for Training and Evaluating Generative Reading Comprehension Metrics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 6521–6532.

[10] Lingjiao Chen, Matei Zaharia, and James Zou. 2023. How is ChatGPT's Behavior Changing Over Time? (2023). arXiv:2307.09009

[11] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V Le, and Jason Wei. 2022. Scaling Instruction-Finetuned Language Models. (2022). arXiv:2210.11416

[12] 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*. Association for Computational Linguistics, Punta Cana, Dominican Republic, 854–870.

[13] Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. The False Promise of Imitating Proprietary LLMs. (2023). arXiv:2305.15717

[14] Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B Brown, Prafulla Dhariwal, Scott Gray, Chris Hallacy, Benjamin Mann, Alec Radford, Aditya Ramesh, Nick Ryder, Daniel M Ziegler, John Schulman, Dario Amodei, and Sam McCandlish. 2020. Scaling Laws for Autoregressive Generative Modeling. (2020). arXiv:2010.14701

[15] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The Curious Case of Neural Text Degeneration. In *International Conference on Learning Representations*.

[16] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised Dense Information Retrieval with Contrastive Learning. *Transactions on Machine Learning Research* (2022).

[17] Gautier Izacard and Edouard Grave. 2021. Distilling Knowledge from Reader to Retriever for Question Answering. In *International Conference on Learning Representations*.

[18] Gautier Izacard and Edouard Grave. 2021. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Association for Computational Linguistics, Online, 874–880.

[19] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. (2023). arXiv:2310.06825

[20] Ehsan Kamalloo, Nouha Dziri, Charles Clarke, and Davood Rafiei. 2023. Evaluating Open-Domain Question Answering in the Era of Large Language Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Toronto, Canada, 5591–5606.

[21] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling Laws for Neural Language Models. (2020). arXiv:2001.08361

[22] 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)*. Association for Computational Linguistics, Online, 6769–6781.

[23] 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*. Association for Computational Linguistics, Florence, Italy, 6086–6096.

[24] Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. AlpacaEval: An Automatic Evaluator of Instruction-following Models. https://github.com/tatsu-lab/alpaca_eval.

[25] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*. Association for Computational Linguistics, Barcelona, Spain, 74–81.

[26] Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021. Generation-Augmented Retrieval 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)*. Association for Computational Linguistics, Online, 4089–4100.

[27] Sewon Min, Jordan Boyd-Graber, Chris Alberti, Danqi Chen, Eunsol Choi, Michael Collins, Kelvin Guu, Hannaneh Hajishirzi, Kenton Lee, Jennimaria Palomaki, Colin Raffel, Adam Roberts, Tom Kwiatkowski, Patrick Lewis, Yuxiang Wu, Heinrich Küttler, Linqing Liu, Pasquale Minervini, Pontus Stenetorp, Sebastian Riedel, Sohee Yang, Minjoon Seo, Gautier Izacard, Fabio Petroni, Lucas Hosseini, Nicola De Cao, Edouard Grave, Ikuya Yamada, Sonse Shimaoka, Masatoshi Suzuki, Shumpei Miyawaki, Shun Sato, Ryo Takahashi, Jun Suzuki, Martin Fajcik, Martin Docekal, Karel Ondrej, Pavel Smrz, Hao Cheng, Yelong Shen, Xiaodong Liu, Pengcheng He, Weizhu Chen, Jianfeng Gao, Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Wen-tau Yih. 2021. NeurIPS 2020 EfficientQA Competition: Systems, Analyses and Lessons Learned *(Proceedings of Machine Learning Research, Vol. 133)*. PMLR, 86–111.

[28] OpenAI. 2023. *GPT-4 Technical Report*. Technical Report. arXiv:2303.08774

[29] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training Language Models to Follow Instructions with Human Feedback. In *Advances in Neural Information Processing Systems*, Alice H Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (Eds.). Curran Associates, Inc., 27730–27744.

[30] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, 311–318.

[31] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, 2383–2392.

[32] Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, QiaoQiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021. RocketQAv2: A Joint Training Method for Dense Passage Retrieval and Passage Re-ranking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic,

2825–2835.

[33] Julian Risch, Timo Möller, Julian Gutsch, and Malte Pietsch. 2021. Semantic Answer Similarity for Evaluating Question Answering Models. In *Proceedings of the 3rd Workshop on Machine Reading for Question Answering*. Association for Computational Linguistics, Punta Cana, Dominican Republic, 149–157.

[34] Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How Much Knowledge Can You Pack Into the Parameters of a Language Model?. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 5418–5426.

[35] Chenglei Si, Chen Zhao, and Jordan Boyd-Graber. 2021. What's in a Name? Answer Equivalence For Open-Domain Question Answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 9623–9629.

[36] Devendra Singh, Siva Reddy, Will Hamilton, Chris Dyer, and Dani Yogatama. 2021. End-to-end Training of Multi-document Reader and Retriever for Open-domain Question Answering. In *Advances in Neural Information Processing Systems*, Vol. 34. 25968–25981.

[37] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. (2023). arXiv:2307.09288

[38] Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M Rush, and Thomas Wolf. 2023. Zephyr: Direct Distillation of LM Alignment. (2023). arXiv:2310.16944

[39] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *International Conference on Learning Representations*.

[40] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent Abilities of Large Language Models. *Transactions on Machine Learning Research* (2022).

[41] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*, Vol. 35. Curran Associates, Inc., 24824–24837.

[42] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Online, 38–45.

[43] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *International Conference on Learning Representations*.

[44] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: Open Pre-Trained Transformer Language Models. (2022). arXiv:2205.01068

[45] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations*.