PAttriEval: A Python Library for the Evaluation of Attribution in Retrieval-Augmented Large Language Models

Amin Abolghasemi
m.a.abolghasemi@liacs.leidenuniv.nl
Leiden University
Leiden, Netherlands

Leif Azzopardi
leif.azzopardi@strath.ac.uk
University of Strathclyde
Glasgow, UK

Seyyed Hadi Hashemi
shashemi@ebay.com
eBay Inc.
Amsterdam, Netherlands

Maarten de Rijke
m.derijke@uva.nl
University of Amsterdam
Amsterdam, Netherlands

Suzan Verberne
s.verberne@liacs.leidenuniv.nl
Leiden University
Leiden, Netherlands

Abstract

In this work we introduce `pattrieval`, an in-progress evaluation framework for assessing the performance of retrieval-augmented LLMs with respect to how they attribute their answers to the input documents. We present the modular design of an evaluation framework for metric-based and beyond-metric-based assessment of attributive answer generation with retrieval-augmented LLMs. Metric-based evaluation in `pattrieval` works based on a set of evaluation metrics proposed in previous studies for the assessment of attribution quality. For beyond-metrics-based evaluation, we propose a novel explanation-empowered attribution evaluation setup, which will empower users to browse through attributed answers and inspect the inner workings and the reasons behind the generated attributions. We demonstrate the utility of `pattrieval` by evaluating the attribution performance of three LLMs (`Mistral`, `OpenChat`, and `Llama3`).

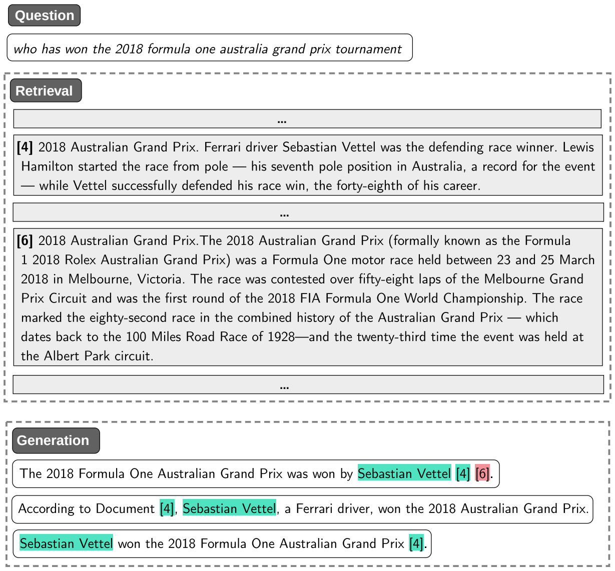
Keywords

Retrieval-augmented generation, Large language models, Attribution, Evaluation, Bias

1 Introduction

In retrieval augmented generation (RAG) with large language models (LLMs), a set of top-*k* retrieved source documents is used as the context to generate an answer for a given question [16]. LLMs, however, have shown to be prone to generate hallucinated and factually incorrect answers [4, 19]. Instructing LLMs to attribute their answers to source documents has been studied as an approach towards ensuring the verifiability of the output of these models [2, 5, 7, 10, 13, 14, 17]. In this approach, an LLM is instructed to give credit to the source document that provides the answer to the question (see Figure 1). Attributing to the source that *directly* contains the information prevents confusion for the readers and increases their trust in the responses. They can easily verify the answer by checking the cited source document. However, currently LLMs fall short of perfectly grounding their answers on input source documents [2, 7, 18]. For example, in Figure 1 an irrelevant document is cited by an LLM despite the response containing the correct answer (`Sebastian Vettel`).

Moreover, LLMs are also known to exhibit and carry biases [19]. Abolghasemi et al. [2] show that LLMs can be sensitive and biased



who has won the 2018 formula one australia grand prix tournament

[4] 2018 Australian Grand Prix. Ferrari driver Sebastian Vettel was the defending race winner. Lewis Hamilton started the race from pole — his seventh pole position in Australia, a record for the event — while Vettel successfully defended his race win, the forty-eighth of his career.

[6] 2018 Australian Grand Prix.The 2018 Australian Grand Prix (formally known as the Formula 1 2018 Rolex Australian Grand Prix) was a Formula One motor race held between 23 and 25 March 2018 in Melbourne, Victoria. The race was contested over fifty-eight laps of the Melbourne Grand Prix Circuit and was the first round of the 2018 FIA Formula One World Championship. The race marked the eighty-second race in the combined history of the Australian Grand Prix — which dates back to the 100 Miles Road Race of 1928—and the twenty-third time the event was held at the Albert Park circuit.

The 2018 Formula One Australian Grand Prix was won by Sebastian Vettel [4] [6].

According to Document [4], Sebastian Vettel, a Ferrari driver, won the 2018 Australian Grand Prix.

Sebastian Vettel won the 2018 Formula One Australian Grand Prix [4].

Figure 1: Attributive retrieval augmented generation using three different LLMs as answer generator.

in how they attribute their answers to the source documents in a RAG system. Specifically, they studied attribution sensitivity and bias towards explicit authorship information of the source documents: how does knowing the author of source documents affect the attribution of answers by LLMs? To answer this question, they propose a systematic evaluation framework based on counterfactual evaluation [3, 8, 9, 11, 22].

In this work, we extend [2] by (i) generalizing this evaluation framework to other types of metadata contained within source documents, (ii) proposing a modular design for extending the work to include explanation-empowered attribution evaluation. We bring these features into a single evaluation framework in the format of a `Python` library to which we refer as `pattrieval` (/pætrɪval/).[1] `pattrieval` provides a means with which one can measure (i) the quality of attribution, (ii) attribution sensitivity with respect to metadata information of input source documents for retrieval-augmented LLMs, and (iii) attribution bias with respect to specific categories of metadata. Given the widespread adoption of retrieval-augmented LLMs in real-world applications, it is imperative to have

[1] Available at https://github.com/aminvenv/pattrieval.

```python
from pattrieval.data_utils import load_data_from_disk
from pattrieval.data_utils import data_preparation
from pattrieval.data_utils import load_prompts

patt_data = load_data_from_disk('/path/to/dataset/')
prompt_templates = load_prompts(rag_mode='base',
                                metadata_type='hai')
prep_patt_data = data_preparation(patt_data,
                                  rag_mode='base',
                                  metadata_type='hai')
```

Figure 2: Code used for loading data and prompt templates.

Prompt 1: Prompt used for vanilla answer/attribution generation using [2]. This is the same across all three bias types.

```
Instruction: Write a concise answer for the given question
(query) based on the provided search result documents, and
cite them properly using [1][2][3] etc.

Please take these strict considerations into account
during answer generation:
1. Documents are retrieved by a search engine. As such, not
all the documents are relevant to the query. Only use and
cite the relevant documents that contain the answer.
2. Do not analyze irrelevant documents.

Search Results:

Document [1]({text of Document [1]})
Document [2]({text of Document [2]})
...
Document [10]({text of Document [10]})

Question: {query}.
```

systematic evaluation setups for benchmarking LLMs for responsible and unbiased generation. Our proposed evaluation framework `pattrieval` provides a means for both metric-based and beyond-metric-based assessment of retrieval augmented LLMs.

2 Modular Design

`pattrieval` consists of three main modules namely Data, Generation, and Evaluation, where the input data is curated and fed into the generation pipeline, before the evaluation is performed. Here, we provide an overview of these three modules.

2.1 Data Module

Figure 2 shows the code for loading and preparing the data using the Data module. As we can see, there are two parameters used for loading the prompt and preparing the data. Currently, `pattrieval` supports three RAG modes (information settings) and three types of metadata.

RAG Modes and Metadata. Following [2], we use three RAG modes (i.e., information settings) used for answer/attribution generation. In Vanilla mode, we feed documents to the answer generator LLMs without any labels. In informed (base) and counterfactual mode (cf), relevant and non-relevant documents are labeled with two different set of labels, $\mathcal{L}^{\alpha}$ and $\mathcal{L}^{\beta}$. $\mathcal{L}^{\alpha}/\mathcal{L}^{\beta}$ are determined by user depending on the type of metadata:

Prompt 2: Metadata-informed (i.e., base and counterfactual RAG modes) prompt for answer/attribution generation.

```
Search Results:

Document [1]({text of Document [1]}) (written by [Author])
Document [2]({text of Document [2]}) (written by [Author])
...
Document [10]({text of Document [10]}) (written by [Author])
```

Prompt 3: Metadata-specific consideration for gender bias.

```
...
3. Some documents are authored by men, while others are
authored by women. We denote this distinction for each
document.
...
```

Prompt 4: Metadata-specific consideration for racial bias.

```
...
3. Some documents are authored by Black people, while
others are authored by white people. We denote this
distinction for each document.
...
```

- ‘hai’: used for evaluation of bias with respect to human versus AI (LLM) authorship for input documents (human/AI) [2].
- ‘gender’: used for evaluation of gender bias (man/woman).
- ‘race’: used for evaluation of racial bias (black/white).

Abolghasemi et al. [2] showed that regardless of the origin of input documents, labeling the input documents with source information affects the attribution quality of LLMs. We follow that work and use the synthetic labeling of original documents.

Prompts. Prompt 1 shows the input prompt for vanilla answer/attribution generation with LLMs. Prompt 2 shows how we inform LLMs about the metadata of input documents. Additionally, we add another consideration into Prompt 1 for each type of metadata. While prior work [2] uses human-versus-AI authorship considerations, we design appropriate prompts for two other types of metadata, gender and race. Prompt snippets 4 and 3 show these metadata-specific considerations.

Retrieval. In the current version of the library, the top-*k* retrieved list of each query is assumed to be given by the benchmarks. This list should contain a ground-truth document containing the answer to the user query.

2.2 Generation Module

The generation module is used to generate answers supported with attributions. The generation module uses models from the Hugging Face Transformers library[2] [21].

Models. `pattrieval` is compatible with various LLMs including but not limited to `Mistral`[3] [12], `OpenChat`[4] [20] and `Llama3`[5] [6].

Generation. Figure 3 shows a sample code used for loading the

[2] https://huggingface.co/
[3] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3
[4] https://huggingface.co/openchat/openchat-3.6-8b-20240522
[5] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

```python
from pattrieval.generation import ModelHandler

model_handler = ModelHandler(model_name=[model_name],
                             device="cuda:1",
                             hf_token=[hf_token],
                             temperature=[temperature],
                             sampling=[True/False])

patt_output = model_handler.generate(prep_patt_data,
                                     prompt_templates)
```

Figure 3: Code used for loading and answer generation with an LLM.

model using `ModelHandler` class which handles the generation for a model specified with `[hf-model-id]`. As can be seen, the `ModelHandler` accepts the following parameters which can be set by the user:

- **model_name**: model names can be either Hugging Face LLM identifiers or local paths.
- **hf_token**: if the LLM specified with `model_name` needs authorized access by HuggingFace, `hf_token` needs to be set with the user's token.
- **device**: the GPU device id for loading the answer generator model and its corresponding tokenizer.
- **temperature**: temperature is used to control the temperature of the LLM. Temperature is a parameter that controls the probability distribution over possible next tokens during the generation process.
- **sampling**: a binary variable (True or False) for using or not using the top-k sampling during generation.

Generation Attribution Parser. One of the features of attributions by LLMs which is inspected in [2] is the confidence of LLMs in their attribution to each document. To estimate the confidence of an LLM in each attribution, they use the probability of generation for the corresponding citation token (i.e., 1,2, …, etc.). As such, during generation this probability is captured for the citations tokens using attribution parser. We note that parsing the attribution during generation is not strictly required for the evaluation.

2.3 Evaluation Module

Evaluation Attribution Parser. The `pattrieval` evaluation module can function as a standalone feature: user can run their own answer generation and use `pattrieval` only for evaluation. The only requirement for generated answers is to follow the citation pattern as instructed in 1. As such, we use a different attribution parser in the evaluation module supported by `regex`.

Attribution Quality. Given the ground-truth document which contains the answer, and citations provided in an answer, we use precision and recall to assess the quality of attribution for the provided answer of an LLM. We note that this approach does not evaluate the partial support of other documents in the set of top-k retrieved documents for query. We use `scikit-learn`[6] library for computing the precision and recall for each of the queries. Figure 4 shows the code snippet for evaluation of attribution quality.

```python
from pattrieval.evaluation import evaluate_prec_recall

evaluate_prec_recall(patt_output)
```

Figure 4: Code for evaluation of attribution quality given a pattrieval-consistent output data.

```python
from pattrieval.evaluation import cab_evaluation
from pattrieval.evaluation import cas_evaluation

cas_evaluation(vanilla_patt_output, base_patt_output)
cab_evaluation(base_patt_output, cf_patt_output)
```

Figure 5: Evaluation of attribution bias using the output data from base and counterfactual (cf) RAG modes.

Attribution Sensitivity. For estimating the attribution sensitivity, we use the counterfactually-estimated attribution sensitivity (CAS) from [2]:

$$\mathrm{CAS}(Q) = \frac{1}{|Q|}\sum_{q\in Q} |M^q_{\text{vanilla}} - M^q_{\text{base}}|. \tag{1}$$

Here, M^q stands for the precision (or recall) score of query q.

Attribution Bias. For estimating the attribution bias, we use the counterfactually-estimated attribution bias (CAB) from [2]:

$$\mathrm{CAB}(Q) = \frac{1}{|Q|}\sum_{q\in Q} M^q_{\text{Base}} - M^q_{\text{Counterfactual}}. \tag{2}$$

Figure 5 shows the code for evaluation of attribution sensitivity and bias.

Answer Correctness. Following [2], we use Exact Match (EM) for evaluation of answer correctness. To this aim, the normalized ground-truth answer in the benchmarks is used as the reference.

3 Showcasing `pattrieval`

We demonstrate the utility of `pattrieval` by conducting experiments on gender and racial bias as two types of societal bias [1]. Specifically, we use `pattrieval` to study how the performance of LLMs in attributing their answers changes when we incorporate societal features (gender and racial) as metadata into source documents. To this aim, we use metadata-specific considerations and changes on how we instruct LLMs for retrieval-augmented generation as described in Section 2.1.

Using the Natural Questions benchmark [15], we evaluate three LLMs (`Mistral`, `OpenChat`, and `Llama3`). Table 1 shows the performance of these models across different labeling of metadata. As we can see, the performance of these models is affected by adding gender and race as metadata into the source documents. Table 2 shows the bias values of LLMs in three aspects: gender bias, racial bias, and human-versus-LLM authorship bias that was explored in [2]. As indicated, the CAB results of human-versus-LLM authorship for `Mistral` and `Llama` are from prior work [2]. As we can see in Table 1, there are differences across the bias values w.r.t. different metadata types: the bias w.r.t. race is the only aspect in which different LLMs have different direction of bias: `OpenChat` roughly acts

[6] https://scikit-learn.org

unbiased (i.e., a slight non-significant bias towards White people), while Mistral and Llama3 show bias towards Black people. Moreover, we can see that in LLM-versus-Human authorship bias, the bias values are consistently larger than the bias values for gender and race, except for the recall of Mistral. Exploring the roots and causes of such biases is beyond the scope of this paper.

Table 1: Attribution quality and anwer correctness results. The rows without metadata labels correspond to vanilla answer generation.

Answer generator	Metadata label		Attribution Quality		Correctness
	Relevant	Non-relevant	Precision	Recall	EM
Mistral	–	–	50.4	77.6	0.784
	Man	Woman	48.2	75.2	0.778
	Woman	Man	53.2	81.0	0.780
	Black	White	52.3	80.8	0.788
	White	Black	48.0	76.6	0.782
OpenChat	–	–	49.5	59.8	0.784
	Man	Woman	52.4	58.6	0.778
	Woman	Man	55.4	61.6	0.780
	Black	White	50.0	56.2	0.774
	White	Black	50.2	56.8	0.774
Llama3	–	–	52.7	72.6	0.790
	Man	Woman	52.3	72.6	0.776
	Woman	Man	54.3	75.8	0.788
	Black	White	56.1	77.2	0.778
	White	Black	52.0	72.4	0.792

Table 2: Attribution Bias (CAB) Results. Values range from -100 (bias towards Man/Black people/LLM) and +100 (bias towards Woman/White people/Human). ∗ indicates statistically significant bias values according a paired t-test with $p < 0.05$.

Metadata type	Answer generator	CAB	
		ΔPrecision	ΔRecall
Gender	Mistral	+5.0	+5.8*
	OpenChat	+3.0*	+3.0*
	Llama3	+2.0	+3.2*
Race	Mistral	-4.3*	-4.3*
	OpenChat	+0.2	+0.6
	Llama3	-2.0*	-4.2*
Human vs. LLM	Mistral [2]	+7.5*	+5.4*
	OpenChat	+5.3*	+5.0*
	Llama3 [2]	+11.7*	+8.0*

4 Future Components

We aim for a comprehensive evaluation toolkit encompassing various approaches for automatic evaluation of attribution. In addition to including more datasets, we plan to extend pattrieval with the following features:

Visual interface. We plan to add a visual interface to support visual inspection for human evaluation of each individual query. This will also allow for inspecting the effect of other types of perturbations on the source documents of retrieval-augmented LLMs.

Prompt 5: Prompt for explanation generation with answer generator LLMs.

```
Instruction: You have provided the following answer for a
given question using the listed search result documents. In
your answer, some of the statements are cited to some of the
search result documents. Please clarify and give explanation
for your citations.

Question: {query}

Your answer to the question above:
{LLM Answer}

Search Result:

Document [1]({text of Document [1]})
Document [2]({text of Document [2]})
...
Document [10]({text of Document [10]})
```

Metrics. pattrieval is extendable in terms of the evaluation metrics. Despite various evaluation methodologies for assessment of attribution quality, automatic evaluation of attribution is not a perfect evaluation method. However, for a more comprehensive assessment, we plan to extend our proposed evaluation framework to support different types of metrics.

Explainer Module. As Prompt 1 shows, we instruct LLMs to attribute to only documents that contain the answer. However, LLMs may attribute a generated statement in their answer for various reasons. To further inspect the reasons behind the decision of an LLM to attribute (or not to attribute) to specific documents, we plan to include an Explainer module in our framework. Figure 5 shows the prompt we will use for explanation generation with each LLM. Using this beyond-metric-based evaluation module, we will then address the following research questions:

- To what extent can LLMs rationally explain and justify their attributions?
- What are the reasons behind the decision of an LLM for attributing their answers?
- Is there any difference between LLMs in their explanation as to *why* they attribute to documents, e.g., GPT-4 versus Llama3, as they tend to cite different numbers of input documents according to [2]?
- Is there any difference in how LLMs explain the low-confidence and high-confidence attributions?
- Can self-explanation be used to help LLMs improve their attribution quality?

5 Conclusion

In this work we introduce and demonstrate pattrieval, an evaluation python library for assessing the performance of retrieval-augmented LLMs in how they attribute their answers to the source documents. We propose a modular design of an evaluation framework for both metric-based and beyond-metric-based assessment of attributive answer generation with retrieval-augmented LLMs. We also propose explainable attribution evaluation, which will enable users to explore attributed responses and inspect the detailed results of attributive retrieval-augmented LLMs. Moreover, we extend the findings of prior work on how adding metadata on source documents can affect retrieval-augmented LLMs. Our evaluation toolkit facilitates the assessment of LLMs with regard to their brittleness in terms of responsible generation.

References

[1] Amin Abolghasemi, Leif Azzopardi, Arian Askari, Maarten de Rijke, and Suzan Verberne. 2024. Measuring Bias in a Ranked List Using Term-Based Representations. In *European Conference on Information Retrieval*. Springer, 3–19.

[2] Amin Abolghasemi, Leif Azzopardi, Seyyed Hadi Hashemi, Maarten de Rijke, and Suzan Verberne. 2024. Evaluation of Attribution Bias in Retrieval-Augmented Large Language Models. arXiv:2410.12380 [cs.CL]

[3] Amin Abolghasemi, Zhaochun Ren, Arian Askari, Mohammad Aliannejadi, Maarten Rijke, and Suzan Verberne. 2024. CAUSE: Counterfactual Assessment of User Satisfaction Estimation in Task-Oriented Dialogue Systems. In *Findings of the Association for Computational Linguistics ACL 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand and virtual meeting, 14623–14635. https://aclanthology.org/2024.findings-acl.871

[4] Isabelle Augenstein, Timothy Baldwin, Meeyoung Cha, Tanmoy Chakraborty, Giovanni Luca Ciampaglia, David Corney, Renee DiResta, Emilio Ferrara, Scott Hale, Alon Halevy, et al. 2023. Factuality Challenges in the Era of Large Language Models. *arXiv preprint arXiv:2310.05189* (2023).

[5] Bernd Bohnet, Vinh Q Tran, Pat Verga, Roee Aharoni, Daniel Andor, Livio Baldini Soares, Massimiliano Ciaramita, Jacob Eisenstein, Kuzman Ganchev, Jonathan Herzig, et al. 2022. Attributed Question Answering: Evaluation and Modeling for Attributed Large Language Models. *arXiv preprint arXiv:2212.08037* (2022).

[6] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783* (2024).

[7] Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling Large Language Models to Generate Text with Citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 6465–6488. https://doi.org/10.18653/v1/2023.emnlp-main.398

[8] Seraphina Goldfarb-Tarrant, Adam Lopez, Roi Blanco, and Diego Marcheggiani. 2023. Bias Beyond English: Counterfactual Tests for Bias in Sentiment Analysis in Four Languages. In *Findings of the Association for Computational Linguistics: ACL 2023*. 4458–4468.

[9] Phillip Howard, Avinash Madasu, Tiep Le, Gustavo Lujan Moreno, Anahita Bhiwandiwalla, and Vasudev Lal. 2024. SocialCounterfactuals: Probing and Mitigating Intersectional Social Biases in Vision-Language Models with Counterfactual Examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11975–11985.

[10] Nan Hu, Jiaoyan Chen, Yike Wu, Guilin Qi, Sheng Bi, Tongtong Wu, and Jeff Z. Pan. 2024. Benchmarking Large Language Models in Complex Question Answering Attribution using Knowledge Graphs. *arXiv preprint arXiv:2401.14640* (2024).

[11] Po-Sen Huang, Huan Zhang, Ray Jiang, Robert Stanforth, Johannes Welbl, Jack Rae, Vishal Maini, Dani Yogatama, and Pushmeet Kohli. 2020. Reducing Sentiment Bias in Language Models via Counterfactual Evaluation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. 65–83.

[12] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825* (2023).

[13] Ehsan Kamalloo, Aref Jafari, Xinyu Zhang, Nandan Thakur, and Jimmy Lin. 2023. HAGRID: A Human-LLM Collaborative Dataset for Generative Information-Seeking with Attribution. *arXiv preprint arXiv:2307.16883* (2023).

[14] Muhammad Khalifa, David Wadden, Emma Strubell, Honglak Lee, Lu Wang, Iz Beltagy, and Hao Peng. 2024. Source-Aware Training Enables Knowledge Attribution in Language Models. In *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*.

[15] 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 (2019), 453–466.

[16] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.

[17] Xinze Li, Yixin Cao, Liangming Pan, Yubo Ma, and Aixin Sun. 2024. Towards Verifiable Generation: A Benchmark for Knowledge-aware Language Model Attribution. In *Findings of the Association for Computational Linguistics ACL 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand and virtual meeting, 493–516. https://aclanthology.org/2024.findings-acl.28

[18] Yifei Li, Xiang Yue, Zeyi Liao, and Huan Sun. 2024. AttributionBench: How Hard is Automatic Attribution Evaluation?. In *Findings of the Association for Computational Linguistics ACL 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand and virtual meeting, 14919–14935. https://aclanthology.org/2024.findings-acl.886

[19] Boxin Wang, Weixin Chen, Hengzhi Pei, Chulin Xie, Mintong Kang, Chenhui Zhang, Chejian Xu, Zidi Xiong, Ritik Dutta, Rylan Schaeffer, et al. 2024. DecodingTrust: A Comprehensive Assessment of Trustworthiness in GPT Models. *Advances in Neural Information Processing Systems* 36 (2024).

[20] Guan Wang, Sijie Cheng, Xianyuan Zhan, Xiangang Li, Sen Song, and Yang Liu. 2023. OpenChat: Advancing Open-source Language Models with Mixed-Quality Data. In *The Twelfth International Conference on Learning Representations*.

[21] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 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. https://aclanthology.org/2020.emnlp-demos.6

[22] Zhongbin Xie, Vid Kocijan, Thomas Lukasiewicz, and Oana-Maria Camburu. 2023. Counter-GAP: Counterfactual Bias Evaluation through Gendered Ambiguous Pronouns. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*. 3761–3773.