SOURCE-AWARE TRAINING ENABLES KNOWLEDGE ATTRIBUTION IN LANGUAGE MODELS

Muhammad Khalifa[†*], **David Wadden**[¶], **Emma Strubell**[¶§],
Honglak Lee[†], **Lu Wang**[†], **Iz Beltagy**[¶], **Hao Peng**[‖*]
University of Michigan[†], Allen Institute for AI[¶], Carnegie Mellon University[§],
University of Illinois Urbana-Champaign[‖]
khalifam@umich.edu

ABSTRACT

Large language models (LLMs) learn a vast amount of knowledge during pretraining, but they are often oblivious to the source(s) of such knowledge. We investigate the problem of *intrinsic source citation*, where LLMs are required to cite the pretraining source supporting a generated response. Intrinsic source citation can enhance LLM transparency, interpretability, and verifiability. To give LLMs such ability, we explore *source-aware training*—a post pretraining recipe that involves **(i)** training the LLM to associate unique source document identifiers with the knowledge in each document, followed by **(ii)** an instruction-tuning to teach the LLM to cite a supporting pretraining source when prompted. Source-aware training can easily be applied to pretrained LLMs off the shelf, and diverges minimally from existing pretraining/fine-tuning frameworks. Through experiments on carefully curated data, we demonstrate that our training recipe can enable faithful attribution to the pretraining data without a substantial impact on the model's quality compared to standard pretraining. Our results also highlight the importance of data augmentation in achieving attribution.[1]

1 INTRODUCTION

Large language models (LLMs) often generate content that is not based on factual information (Ji et al., 2023; Ye et al., 2023a). As LLMs are pretrained over noisy web data that often contains inaccurate or outdated content, users should be able to verify LLM outputs by checking their sources. Moreover, concerns about copyright infringement (Min et al., 2023; Longpre et al., 2023), privacy violations (Kim et al., 2024), data contamination (Shi et al., 2023), and toxic content (Gehman et al., 2020) in LLMs emphasize the need for techniques to identify and trace the origins of information included in models' responses. It is therefore desirable if LLMs can provide supporting evidence for their responses by citing or attributing the outputs to the sources they draw upon (Rashkin et al., 2023; Huang & Chang, 2023; Li et al., 2023b). Beyond improving the models' transparency, attribution allows for a deeper understanding of the relationship between training data and model behaviors, thereby offering a pathway to refine the quality of pretraining data.

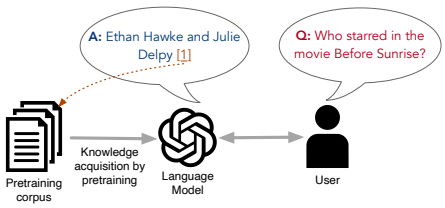

Figure 1: **Intrinsic source citation:** The language model cites pretraining document(s) from which it acquired its relevant parametric knowledge.

We focus on *intrinsic source citation*, where the LLM should cite source documents from the pretraining data from which it acquired its relevant parametric knowledge. Compared to retrieval-based approaches such as RAG (Lewis et al., 2020; Guu et al., 2020) or post-hoc techniques (He et al., 2023; Gao et al., 2023a), intrinsic source citation is inherently tied to the model itself, enables more

[*] Work partially done while these authors were at the Allen Institute for AI.

[1] Code and data available here: https://github.com/mukhal/intrinsic-source-citation.

faithful attribution to its parametric knowledge, thus opens up unique opportunities for improved interpretability (Alvarez Melis & Jaakkola, 2018; Marasovic et al., 2022).

To this end, we explore *source-aware training*—a post-pretraining recipe that enables a LLM to cite its pretraining data based on its parametric knowledge. Our motivation is three-fold. First, a significant portion of an LLM's knowledge is acquired during pretraining, therefore citing evidence for this parametric knowledge can greatly enhance the LLM trustworthiness. Second, the standard practice for LLM pretraining neglects the attribution angle, which explains why the current generation of LLMs fails to provide reliable citations (Agrawal et al., 2023; Zuccon et al., 2023). We aim to explore a training procedure that naturally facilitates citation of the pretraining data. Finally, from a scientific perspective, it is intriguing to investigate whether and how current language models can be trained to reference their pretraining data.

We inquire: Given an off-the-shelf LLM, can we train it to attribute its generations to the supporting sources from the pretraining data? Our goal is to cite the pretraining documents themselves (see Figure 1). Our setup mirrors existing frameworks for LLM pretraining and can be summarized as follows: We take an off-the-shelf LLM, continue pretraining it on a corpus associating each document with a unique identifier, then fine-tune it to answer questions about the acquired knowledge while providing citations. The citation is achieved by generating an identifier of a document supporting the answer. Continual pretraining is done as in prior work, with the main difference of injecting the document identifiers into the pretraining data—minimal changes in the model's architecture or implementation are needed.

To study the generalization over this task and simulate a realistic fine-tuning setting, we limit our instruction tuning stage to a subset of the pretraining documents (in-domain) and evaluate the model's attribution ability over the remaining (out-of-domain) documents. We run experiments over a synthetic pretraining corpus of fake biographies and show that LLMs can achieve reasonable attribution when answering a question about the out-of-domain documents.

Our contributions are summarized as follows:

- To the best of our knowledge, this work is the first to study intrinsic source citation and investigate the ability of current LLMs to cite the source of their parametric knowledge.
- We explore a source-aware training recipe that can be applied to off-the-shelf LLMs to give them the ability to attribute their outputs to the pretraining sources. On synthetic data, we show that such training can achieve reasonable attribution while maintaining a good balance with the LLM quality compared to standard pretraining.
- We examine the impact of various training decisions, such as data augmentation, on attribution and our findings can inform future efforts to train attribution-capable models at a large scale.

2 SOURCE-AWARE TRAINING

Our training framework is designed to easily integrate with existing pretraining pipelines. We minimize its deviations from established post-pretraining practice, and it involves almost no modifications to the model architecture or implementation. Each document in the pretraining corpus is assigned a unique document identifier (ID) and our goal is to train a language model that can respond to user prompts by providing both a response and an ID referring to the source document of the model's knowledge.

Setup. Our evaluation follows the attributed question answering setup (Bohnet et al., 2022), where given an input prompt z, the LLM output will consist of a tuple $\langle r, c \rangle$ where r is the response (e.g., the answer to a question) and c is the identifier of the document in the pretraining data that supports the answer. Following standard LLM training setups, our recipe has two stages: Continual pretraining (Section 2.1) and instruction tuning (Section 2.2). Instruction tuning trains the model to be able to attribute the generated responses to supporting documents it has seen during pretraining. The pretraining stage will involve all documents by nature, but the instruction tuning step is restricted to a subset of the pretraining documents. This restriction is due to the potential cost of curating instruction tuning data from all the pretraining documents, that is in addition to the training overhead incurred by instruction tuning (Zhou et al., 2024).

After training, we measure **out-of-domain (OOD) attribution**: whether the model can attribute knowledge to documents that are only included in the continual pretraining data but *not* in the instruction tuning data. We therefore split the pretraining corpus into in-domain and OOD subsets.

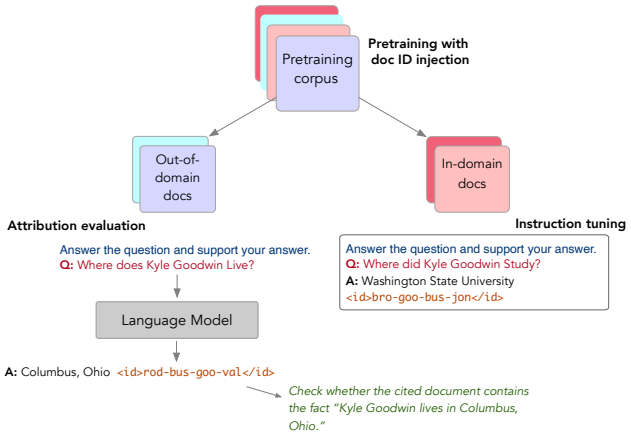

Figure 2: **Training and evaluation setup:** The pretraining corpus is split into in-domain and out-of-domain documents. The in-domain documents are used to create instruction tuning examples, and the out-of-domain documents are used for attribution evaluation.

The in-domain data is used to create attribution training examples, while the OOD documents are used for evaluation, as shown in Figure 2.

2.1 Continual Pretraining with Doc ID Injection

Doc ID injection. The continual pretraining phase has two goals: **(i)** memorizing knowledge via next-word prediction (same as established LLM pretraining), and **(ii)** associating knowledge within a source document with its ID to enable OOD attribution. We aim to achieve the second goal by *injecting* the document ID into the document before training. An important consideration is the location and frequency of injecting the document ID.

Formally, given a pretraining corpus of documents $\{x^{(1)}, x^{(2)}, ..., x^{(m)}\}$ and their corresponding IDs $\{c^{(1)}, c^{(2)}, ..., c^{(m)}\}$ where each $x^{(i)}$ is a sequence of tokens $x^{(i)} = (x_1^{(i)}, x_2^{(i)}, ..., x_{|x_i|}^{(i)})$, and each $c^{(i)} = (c_1^{(i)}, c_2^{(i)}, ..., c_{|c_i|}^{(i)})$ is a sequence of tokens of its identifier. Our pretraining aims to learn the language model parameters θ that maximize the objective $\max_\theta \sum_{i=1}^{m} \log P(\hat{x}^{(i)}; \theta)$, where $\hat{x}^{(i)}$ is the ID-injected version of the document $x^{(i)}$. We inject the doc ID into document a x with different strategies, each of which corresponds to a different $\hat{x}$.[2] Particularly, we experiment with the following strategies:

1. **NO-ID**: Standard pretraining without ID injection: $\hat{x} := x$.
2. **DOC-BEGIN**: Inject the ID once before the first token in the document: $\hat{x} := \{c_1, c_2, ..., c_{|c_i|}, x_1, x_2, ..., x_{|x_i|}\}$.
3. **DOC-END**: Inject once after the last token in the document. This is equivalent to $\hat{x} := \{x_1, x_2, ..., x_{|x_i|}, c_1, c_2, ..., c_{|c_i|}\}$.[3]
4. **REPEAT**: Inject the ID after *every* sentence in both in-domain and OOD documents. Here, $\hat{x} := \{X_{s_1}, c_1, ..., c_{|c_i|}, X_{s_2}, c_1, c_2, ..., c_{|c_i|}, ..., X_{s_k}, c_1, c_2, ..., c_{|c_i|}\}$, where X_{s_j} are the tokens in x corresponding to the j-th sentence in document x and assuming x has k sentences.

Attention masking to improve training efficiency. The doc ID tokens for a certain document will naturally attend to preceding tokens from other documents in same training sequence. Our initial experiments showed that this severely hurts attribution, since the model will associate the doc ID of a given document with tokens from other documents in the same training sequence. To avoid this, we modify the causal self-attention mask during pretraining such that the ID tokens for a given document only attend to tokens from within that document.

[2]We omit the superscript for brevity.

[3]DOC-END results in the same training objective as in DSI (Tay et al., 2022), where the model is trained to generate the ID given the full document. While this objective was shown to work for the information retrieval setup, we find that it fails to generalize in attribution.

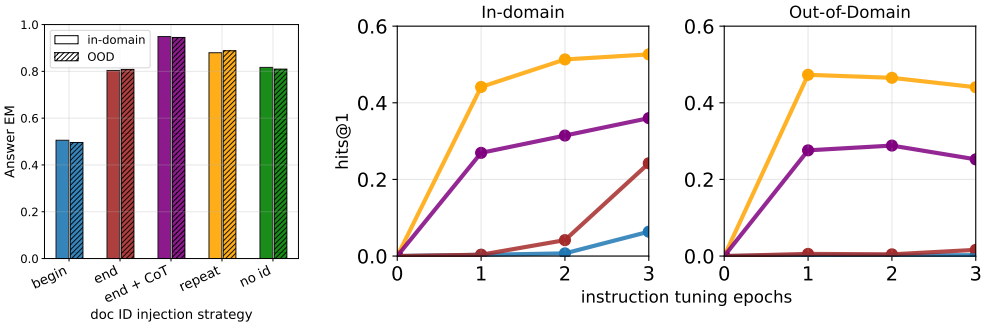

Figure 3: **Left:** Answer EM over questions from in-domain and OOD documents after 1 fine-tuning epoch with different ID injection strategies (Section 2.1). The LLM can generalize well to out-of-domain (OOD) questions in all document ID locations, although both in-domain and OOD answer EM scores degrade with DOC-BEGIN. **Right:** Hits@1 over in-domain and OOD questions during instruction tuning. Only REPEAT and DOC-END + CoT can achieve OOD attribution.

2.2 Instruction Tuning

In addition to pretraining, we further adapt the model to **(i)** recall the appropriate knowledge as a response to the prompt and **(ii)** cite the ID of the document supporting the response.[4] This stage does not teach the model any new knowledge, but merely aims at eliciting memorization of both knowledge and doc ID by instruction tuning (Wei et al., 2021; Zhang et al., 2023). Given l examples, the i-th example is a tuple $\langle z^{(i)}, r^{(i)}, c^{(i)} \rangle$, where $z^{(i)}$ is the prompt (instruction + query), $r^{(i)}$ is a ground-truth response, and $c^{(i)}$ is the ID of a document that supports the response. The model is trained with the objective $\max_\theta \sum_{i=1}^{l} \log P(r^{(i)}|z^{(i)};\theta)P(c^{(i)}|r^{(i)}, z^{(i)};\theta)$. The instruction-tuning examples only come from the in-domain documents, and we use the instruction *"Answer the following question and provide evidence for your answer."* Figure 2 shows a fine-tuning example from BIOCITE. During the standard LLM pretraining, i.e., with NO-ID, we remove the doc ID part from instruction tuning examples. Following Taylor et al. (2022), we surround document IDs with two learned special tokens `<id>` and `</id>` during both pretraining and fine-tuning.

2.3 Chain-of-Thought Attribution

The setup in the previous section train the model to recall the doc ID immediately after the answer. Another setup we explore is where the model is asked to cite the ground-truth document (or part of it) before generating the doc ID. This can be thought of as an instance of chain-of-thought (CoT) (Nye et al., 2021; Wei et al., 2022). In CoT, we inject the document using DOC-END, then the model is trained to cite the rest of the document after the answer up till the doc ID. Figure 5 in the Appendix shows an example of the CoT setup.

3 RESULTS AND DISCUSSION

To have a controlled experimental setting, we rely on pretraining knowledge in the form of *atomic synthetic facts*. We describe how we construct BIOCITE, our synthetic pretraining corpus, in Appendix B.2. We also provide the training and evaluation details are in Appendix D.

Downstream QA Performance. We start by evaluating the QA performance on OOD quuestions. Figure 3 (left) shows answer match over BIOCITE with different document ID injection strategies. The model can achieve OOD answer match $> 80\%$, showing that the model has well memorized the pretraining knowledge. We also note that DOC-BEGIN achieves much worse QA performance than other strategies, and we hypothesize that DOC-BEGIN conditions the model to expect the ID when citing knowledge, causing a mismatch during inference when the ID is absent.

OOD attribution depends on ID injection strategy. The ID injection strategy plays a major role in OOD attribution achieved by source-aware training. As shown in Figure 3 (right), placing the ID only once with DOC-BEGIN or DOC-END performs poorly. We hypothesize that both cases train the model to associate the *full* document—rather than individual facts—with the document ID. Precisely,

[4]The instruction tuning examples are curated from the pretraining data such that for a given prompt, we already have the reference document the model should cite. More details are in **??**.

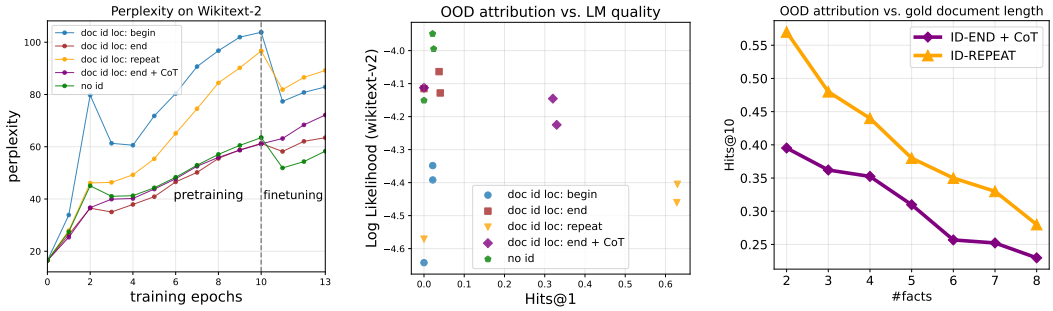

Figure 4: **Left:** LLM quality vs. OOD attribution. Higher is better for Hits@1 and Log Likelihood. Optimal is top-right corner. DOC-END + CoT is Pareto-optimal as it strikes the best balance between LLM quality and OOD attribution. **Right:** OOD attribution performance with different gold document lengths.

DOC-END conditions the model on the full document when generating the doc ID, but the evaluation requires the model to predict the ID given *individual* facts not full documents. This is an instance of LLM generalization failures in knowledge extraction discussed in prior work (Zhu & Li, 2023; Allen-Zhu & Li, 2023) and explains why REPEAT is substantially better, since it trains the model to predict the ID after each fact, making it easier for the model to associate individual facts with the ID.

Chain-of-thought attribution helps. REPEAT may be unfavorable, since the number of pretraining tokens will noticeably increase (by about 80% with REPEAT compared to NO-ID), bringing additional training overhead. Besides, the model quality will be negatively impacted since document IDs are not natural text, which is reflected in the perplexity over Wikitext-v2 shown in Figure 3 (right). The question here is whether source-aware training can yield OOD attribution while injecting the doc ID *once*. Interestingly, the chain-of-thought setup (Section 2.3) achieves reasonable OOD attribution without requiring repeating the doc ID within the document. The results above suggest that source-aware training can teach the model to attribute its parametric knowledge to their pretraining sources, with one key choice to consider: the doc ID injection strategy. Another key component is document augmentation, which we discuss next.

Importance of document augmentation. We compare two types of data augmentation methods: document and fact augmentation, and the goal is to assess which type of augmentation is necessary for OOD attribution. Document augmentation is done by permuting the facts within a document N_{aug} times while fact augmentation duplicates the facts in a document in N_{aug} different random documents. Figure 6 in Appendix shows OOD answer match and Hits@1 as N_{aug} is varied and where $N_{\text{aug}} = 1$ means no augmentation. While answer match improves using fact-level augmentation, Hits@1 remains the same and only improves when we apply document augmentation. Document augmentation appears necessary for the model to associate the doc ID with the facts in the document.

Impact on perplexity. Now we study the impact of different document ID injection strategies on the LLM quality measured in terms of perplexity over Wikitext-v2. Figure 3 (right) shows perplexity trends during both pretraining and instruction tuning over BIOCITE and Figure 4 (Left) visualizes the tradeoff between LLM quality and OOD attribution. We note that perplexity increases during training in all setups due to the domain shift incurred by training BIOCITE. Additionally, REPEAT exhibits the worst perplexity, since frequent ID injection means training on more non-natural text. Even though DOC-END + CoT leads to worse perplexity than NO-ID, it is still substantially better compared REPEAT and is Pareto-optimal as shown in Figure 4 (Left). These results that DOC-END + CoT strikes the best balance between OOD attribution and maintaining the model's quality.

OOD attribution vs. document complexity. We analyze how OOD attribution varies with the complexity of the document measured in terms of the number of facts when training with REPEAT and DOC-END + CoT. In Figure 4 (Right), we plot OOD attribution measured with Hits@10 changes as the number of facts in the gold document changes. We observe a consistent trend where documents with more facts are harder to cite. This can be explained by the limited representational capacity of the doc IDs: Documents with more facts require the doc ID to be associated with more knowledge.

REFERENCES

Ayush Agrawal, Lester Mackey, and Adam Tauman Kalai. Do language models know when they're hallucinating references? *CoRR*, abs/2305.18248, 2023. doi: 10.48550/ARXIV.2305.18248. URL https://doi.org/10.48550/arXiv.2305.18248.

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.2, knowledge manipulation. *CoRR*, abs/2309.14402, 2023. doi: 10.48550/ARXIV.2309.14402. URL https://doi.org/10.48550/arXiv.2309.14402.

David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. *Advances in neural information processing systems*, 31, 2018.

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. The reversal curse: Llms trained on" a is b" fail to learn" b is a". *arXiv preprint arXiv:2309.12288*, 2023.

Bernd Bohnet, Vinh Q Tran, Pat Verga, Roee Aharoni, Daniel Andor, Livio Baldini Soares, Massimiliano Ciaramita, Jacob Eisenstein, Kuzman Ganchev, Jonathan Herzig, et al. Attributed question answering: Evaluation and modeling for attributed large language models. *arXiv preprint arXiv:2212.08037*, 2022.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. Improving language models by retrieving from trillions of tokens. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 2206–2240. PMLR, 2022. URL https://proceedings.mlr.press/v162/borgeaud22a.html.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. Autoregressive entity retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=5k8F6UU39V.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Y. Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. RARR: researching and revising what language models say, using language models. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 16477–16508. Association for Computational Linguistics, 2023a. doi: 10.18653/V1/2023.ACL-LONG.910. URL https://doi.org/10.18653/v1/2023.acl-long.910.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling large language models to generate text with citations. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 6465–6488. Association for Computational Linguistics, 2023b. URL https://aclanthology.org/2023.emnlp-main.398.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *arXiv preprint arXiv:2009.11462*, 2020.

Roger Grosse, Juhan Bae, Cem Anil, Nelson Elhage, Alex Tamkin, Amirhossein Tajdini, Benoit Steiner, Dustin Li, Esin Durmus, Ethan Perez, et al. Studying large language model generalization with influence functions. *arXiv preprint arXiv:2308.03296*, 2023.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *International conference on machine learning*, pp. 3929–3938. PMLR, 2020.

Hangfeng He, Hongming Zhang, and Dan Roth. Rethinking with retrieval: Faithful large language model inference. *CoRR*, abs/2301.00303, 2023. doi: 10.48550/ARXIV.2301.00303. URL https://doi.org/10.48550/arXiv.2301.00303.

Jie Huang and Kevin Chen-Chuan Chang. Citation: A key to building responsible and accountable large language models. *CoRR*, abs/2307.02185, 2023. doi: 10.48550/ARXIV.2307.02185. URL https://doi.org/10.48550/arXiv.2307.02185.

Gautier Izacard, Patrick S. H. Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Atlas: Few-shot learning with retrieval augmented language models. *J. Mach. Learn. Res.*, 24:251:1–251:43, 2023. URL http://jmlr.org/papers/v24/23-0037.html.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, 2023.

Siwon Kim, Sangdoo Yun, Hwaran Lee, Martin Gubri, Sungroh Yoon, and Seong Joon Oh. Propile: Probing privacy leakage in large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pp. 1885–1894. PMLR, 2017.

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

Daliang Li, Ankit Singh Rawat, Manzil Zaheer, Xin Wang, Michal Lukasik, Andreas Veit, Felix X. Yu, and Sanjiv Kumar. Large language models with controllable working memory. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 1774–1793. Association for Computational Linguistics, 2023a. doi: 10.18653/V1/2023.FINDINGS-ACL.112. URL https://doi.org/10.18653/v1/2023.findings-acl.112.

Dongfang Li, Zetian Sun, Xinshuo Hu, Zhenyu Liu, Ziyang Chen, Baotian Hu, Aiguo Wu, and Min Zhang. A survey of large language models attribution. *arXiv preprint arXiv:2311.03731*, 2023b.

Nelson F. Liu, Tianyi Zhang, and Percy Liang. Evaluating verifiability in generative search engines. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pp. 7001–7025. Association for Computational Linguistics, 2023. URL https://aclanthology.org/2023.findings-emnlp.467.

Shayne Longpre, Robert Mahari, Anthony Chen, Naana Obeng-Marnu, Damien Sileo, William Brannon, Niklas Muennighoff, Nathan Khazam, Jad Kabbara, Kartik Perisetla, et al. The data provenance initiative: A large scale audit of dataset licensing & attribution in ai. *arXiv preprint arXiv:2310.16787*, 2023.

Kelvin Luu, Rik Koncel-Kedziorski, Kyle Lo, Isabel Cachola, and Noah A. Smith. Citation text generation. *ArXiv*, abs/2002.00317, 2020. URL https://api.semanticscholar.org/CorpusID:211010521.

Ana Marasovic, Iz Beltagy, Doug Downey, and Matthew E. Peters. Few-shot self-rationalization with natural language prompts. In Marine Carpuat, Marie-Catherine de Marneffe, and Iván Vladimir Meza Ruíz (eds.), *Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pp. 410–424. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.FINDINGS-NAACL.31. URL https://doi.org/10.18653/v1/2022.findings-naacl.31.

Sean M McNee, Istvan Albert, Dan Cosley, Prateep Gopalkrishnan, Shyong K Lam, Al Mamunur Rashid, Joseph A Konstan, and John Riedl. On the recommending of citations for research papers. In *Proceedings of the 2002 ACM conference on Computer supported cooperative work*, pp. 116–125, 2002.

Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, et al. Teaching language models to support answers with verified quotes. *arXiv preprint arXiv:2203.11147*, 2022.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Byj72udxe.

Sewon Min, Suchin Gururangan, Eric Wallace, Hannaneh Hajishirzi, Noah A Smith, and Luke Zettlemoyer. Silo language models: Isolating legal risk in a nonparametric datastore. *arXiv preprint arXiv:2308.04430*, 2023.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. Webgpt: Browser-assisted question-answering with human feedback. *CoRR*, abs/2112.09332, 2021. URL https://arxiv.org/abs/2112.09332.

Ramesh M Nallapati, Amr Ahmed, Eric P Xing, and William W Cohen. Joint latent topic models for text and citations. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 542–550, 2008.

Maxwell I. Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. Show your work: Scratchpads for intermediate computation with language models. *CoRR*, abs/2112.00114, 2021. URL https://arxiv.org/abs/2112.00114.

Fabio Petroni, Patrick S. H. Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. How context affects language models' factual predictions. In Dipanjan Das, Hannaneh Hajishirzi, Andrew McCallum, and Sameer Singh (eds.), *Conference on Automated Knowledge Base Construction, AKBC 2020, Virtual, June 22-24, 2020*, 2020. doi: 10.24432/C5201W. URL https://doi.org/10.24432/C5201W.

Ronak Pradeep, Kai Hui, Jai Gupta, Adam D Lelkes, Honglei Zhuang, Jimmy Lin, Donald Metzler, and Vinh Q Tran. How does generative retrieval scale to millions of passages? *arXiv preprint arXiv:2305.11841*, 2023.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Lora Aroyo, Michael Collins, Dipanjan Das, Slav Petrov, Gaurav Singh Tomar, Iulia Turc, and David Reitter. Measuring attribution in natural language generation models. *Computational Linguistics*, pp. 1–66, 2023.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models. *arXiv preprint arXiv:2310.16789*, 2023.

Weiwei Sun, Lingyong Yan, Zheng Chen, Shuaiqiang Wang, Haichao Zhu, Pengjie Ren, Zhumin Chen, Dawei Yin, Maarten Rijke, and Zhaochun Ren. Learning to tokenize for generative retrieval. *Advances in Neural Information Processing Systems*, 36, 2024.

Yi Tay, Vinh Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Prakash Gupta, Tal Schuster, William W. Cohen, and Donald Metzler. Transformer memory as a differentiable search index. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/892840a6123b5ec99ebaab8be1530fba-Abstract-Conference.html.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085*, 2022.

Yujing Wang, Yingyan Hou, Haonan Wang, Ziming Miao, Shibin Wu, Qi Chen, Yuqing Xia, Chengmin Chi, Guoshuai Zhao, Zheng Liu, et al. A neural corpus indexer for document retrieval. *Advances in Neural Information Processing Systems*, 35:25600–25614, 2022.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

Orion Weller, Marc Marone, Nathaniel Weir, Dawn Lawrie, Daniel Khashabi, and Benjamin Van Durme. " according to..." prompting language models improves quoting from pre-training data. *arXiv preprint arXiv:2305.13252*, 2023.

Xinyu Xing, Xiaosheng Fan, and Xiaojun Wan. Automatic generation of citation texts in scholarly papers: A pilot study. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6181–6190, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.550. URL https://aclanthology.org/2020.acl-main.550.

Hongbin Ye, Tong Liu, Aijia Zhang, Wei Hua, and Weiqiang Jia. Cognitive mirage: A review of hallucinations in large language models. *CoRR*, abs/2309.06794, 2023a. doi: 10.48550/ARXIV.2309.06794. URL https://doi.org/10.48550/arXiv.2309.06794.

Xi Ye, Ruoxi Sun, Sercan Ö Arik, and Tomas Pfister. Effective large language model adaptation for improved grounding. *arXiv preprint arXiv:2311.09533*, 2023b.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*, 2024.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*, 2023.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36, 2024.

Zeyuan Allen Zhu and Yuanzhi Li. Physics of language models: Part 3.1, knowledge storage and extraction. *CoRR*, abs/2309.14316, 2023. doi: 10.48550/ARXIV.2309.14316. URL https://doi.org/10.48550/arXiv.2309.14316.

Guido Zuccon, Bevan Koopman, and Razia Shaik. Chatgpt hallucinates when attributing answers. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, pp. 46–51, 2023.

A Related Work

Language Model Attribution. Attribution is gaining more attention recently as interpretability and grounding of language models become increasingly important. Generally speaking, approaches to achieve attribution can be classified as either **retrieval-based** or **model-based**. Retrieval-based approaches include retrieval augmentation (RAG) (Lewis et al., 2020; Guu et al., 2020; Borgeaud et al., 2022; Izacard et al., 2023) and post-hoc attribution (He et al., 2023; Gao et al., 2023a). RAG approaches enable attribution by providing a retrieved context for the LM to use, and teaching LM how to cite the retrieved context (Nakano et al., 2021; Menick et al., 2022). The major limitations of RAG approaches are the lack of guarantee that the model is relying on the retrieved data for generation (Petroni et al., 2020; Li et al., 2023a), and that they only work on non-parametric knowledge. Post-hoc approaches (He et al., 2023; Gao et al., 2023a) attribute the LM outputs by retrieving the supporting evidence given the model's response, but have been shown to produce non-accurate citations (Liu et al., 2023).

Model-based techniques involve prompting the model directly to generate citations for its parametric knowledge (Weller et al., 2023; Zuccon et al., 2023) or scaling techniques such as influence functions (Koh & Liang, 2017) to large models (Grosse et al., 2023). Model-based attribution is arguably more faithful than retrieval-based approaches as the citation mechanism is intrinsic to the model (Alvarez Melis & Jaakkola, 2018; Marasovic et al., 2022). However, standard approaches to pretraining LMs do not take into account the need for the language model to cite its pretraining data, which is where our work comes into play. Bohnet et al. (2022) proposed the task of attributed question-answering and evaluated the attribution performance of different systems using the AutoAIS metric (Rashkin et al., 2023; Gao et al., 2023a). In addition, they fine-tuned PaLM (Chowdhery et al., 2023) to generate both an answer and a URL pointing to Wikipedia page supporting the answer in generative retrieval style (Tay et al., 2022; Wang et al., 2022). Although this setup is similar to ours in that we require the LM to generate the document identifier as well, their setup is basically a variation of RAG where the LM acts as the retriever.

Citation Generation. There is a large body of work on the task of citation generation in the scientific domain, where the goal is to cite an appropriate article given a particular context (McNee et al., 2002; Nallapati et al., 2008) or to generate text citing one article in relation to another (Xing et al., 2020; Luu et al., 2020). A relevant work to ours is Galactica (Taylor et al., 2022), which leverages the underlying citation graph in the pretraining data to learn to predict citations given a context. Notably, Galactica is trained to leverage citations of scientific articles in the pretraining data, while our work explores citation of all the pretraining documents, extending beyond scientific articles. Gao et al. (2023b) introduced a benchmark for the automatic evaluation of LM citations and Ye et al. (2023b) proposed a method to improve language model grounding by fine-tuning the language model on responses that are well supported by their citations. However, their setup is restricted to citation of retrieved rather than parametric knowledge.

Generative Retrieval. Our work is somewhat related to generative retrieval, where an auto regressive model is trained to act as a retriever in an information retrieval (IR) system (Wang et al., 2022; Tay et al., 2022). Generative retrieval typically relies on a transformer model to map a given query to a document identifier that is likely to contain an answer to the query. While our task also requires the language model to generate a document identifier, we differ from generative retrieval in at least two ways. First, our goal is to generate an identifier pointing to a document containing the already generated answer rather than a document that is likely to contain the answer. Second, generative retrieval merely learns a mapping from query to document identifiers, while our setup is concerned with both acquiring knowledge via the next-word prediction objective over the documents and associating acquired knowledge with its source.

B Data details

B.1 Reproducing BioS

As BioS Zhu & Li (2023) has not been publicly released, we reproduced our own version as follows. Each biography lists six different facts about each person: birthdate, birth city, study major, university, employer and work city. To fill values for each fact, we start by compiling a list of first names, last names, company names, cities, universities, and majors. Examples and counts of the seed entities are shown in Table 1.

Then we construct a single biography by sampling a unique full name (first and last names could repeat) and then to make up each fact, we sample a random suitable entity. For example, for birth and work cities, we sample a random city and for employers, we sample a company name and so on. For birthdates, we generate a random date in the range: January 1st, 1900 – December 31st, 2099. Each fact is constructed via a corresponding template, shown in Table 2. To construct questions for instruction tuning with BIOCITE, each attribute is mapped to a corresponding template question as shown in Table 3.

Entity	Count	Examples
First Name	851	John, Mary, David, Aria,...
Last Name	557	Smith, Johnson, Williams,...
Company Name	284	Apple, Google, Microsoft,...
City	199	New York, Chicago, Tampa,...
University	178	University of Alabama, University of Michigan,...
Major	107	Industrial Design, Fashion Design, Psychology,...

Table 1: Counts of the compiled entities used to reproduce BioS Zhu & Li (2023).

Attribute	Fact template
Birthdate	<full name> was born on <birthdate>
Work place	<full name> works at <company>
Work city	<full name> lives in <city>
Birth city	<full name> was born in <city>
University	<full name> studied at <university>
Major	<full name> studied <major>

Table 2: Templates used to construct different biograohy facts for BioS.

Attribute	Question template
Birthdate	When was <full name> born?
Work place	Where does <full name> work?
Work city	Where does <full name> live?
Birth city	Where was <full name> born?
University	Where did <full name> study?
Major	What did <full name> study?

Table 3: Templates used to construct questions from each document in BIOCITE.

B.2 BIOCITE Construction

Sampling facts. BIOCITE is based on the BioS dataset (Zhu & Li, 2023), which is a collection of biographies of fake people where each biography lists six different facts about each person: birthdate, birth city, study major, university, employer, and work city.[5] Each attribute is described using a corresponding template. For example, the birth city fact is described by "<person name> was born in <birth city>." To avoid co-reference issues when sampling facts, the person's full name is mentioned in all the facts.

To simulate realistic pretraining data that often include facts about different entities, we construct each document in BIOCITE as a collection of facts from at least *two* different biographies in BioS. More particularly, to construct one document d, we first sample the number of biographies $n_d \sim \text{Uniform}([2, \cdots, N_{\text{MaxNBio}}])$. Then, we sample n_d biographies from BioS without replacement. Finally, we sample a random number of facts from each one in the n_d biographies and combine these to form the document. We allow the same combination of biographies to create a document only once and allow each fact to appear only once in BIOCITE.[6] In our experiments, we generate 100K documents in total using $N_{\text{MaxNBio}} = 4$.

[5]Details about reproducing BioS are in Appendix B.1.

[6]In this work, we assume each fact in BIOCITE is mentioned in exactly one document and leave the extension of this work to multi-doc citation to future work.

Document: Marleigh Austin works at SpaceX. Marleigh Austin studied at the University of Arkansas, Fayetteville. Isaiah Brown studied Graphic Design. Isaiah Brown was born on October 19, 1930. Lora Johnston was born on May 30, 1989. Lora Johnston works at Microsoft Teams. Kyle Goodwin studied at Washington State University. Kyle Goodwin works at Campari Group. **Doc ID:** `bro-goo-aus-joh`
Instruction tuning example: **Q:** Where does Lora Johnston work? **A:** Microsoft Teams ## `<id>bro-goo-aus-joh</id>`

Table 4: An example document in BIOCITE with its unique identifier and an example question. Documents in BIOCITE are constructed by sampling biographies of fake individuals (highlighted in color) from BioS (Zhu & Li, 2023) and then sampling several facts from each biography.

Creating questions. The input prompts for BIOCITE will take the form of factoid questions about the different facts such as "Where does Lora Jonhston Work?". Question generation is done by mapping each fact in the document to a corresponding question template. For example, a fact about a person's birth city is mapped to the question "Where was `<full name>` born?"

Doc ID format. It has been shown that the document ID design plays a role in generative retrieval performance (Tay et al., 2022; Pradeep et al., 2023; Sun et al., 2024) and we observed the same during our initial experiments. When designing a doc ID, we need to be careful not to make the task too easy, where the model can infer the doc ID from the input question without actually performing attribution. The design of our dataset allows us to use the last names of the individuals included in a document for two reasons. First, two facts from the same person will most likely exist in many different documents. Second, the same last name can be shared by many different biographies, whose individuals differ only in the first name. That means relying on the last name will not be sufficient to predict the correct doc ID. We choose to use a dash-separated concatenation of the 3-letter prefixes of the last names from the biographies that make up the document, shuffled randomly. We analyze the model predictions when prompted with inputs sharing the same person's last name in **??**. Table 4 shows an example document, its ID, and a question extracted from it. Exact dataset statistics are in Table 5 in the Appendix.

Data augmentation. LMs struggle to generalize at knowledge extraction over OOD documents (i.e., document that were not seen during fine-tuning) without a sufficient amount of redundancy where the LM will be exposed to the same fact in different formats/positions (Zhu & Li, 2023; Allen-Zhu & Li, 2023; Berglund et al., 2023). In large-scale pretraining setups, this is achieved by scaling the pretraining data but as we study attribution on a smaller scale, we achieve the same effect of redundancy via data augmentation. We mainly apply doc-level augmentation, by shuffling the sentences in each document $N_{\text{aug}} = 3$ times, where N_{aug} is the number of augmentation samples. Unless otherwise stated, our experiments will include document-level augmentation of the pretraining data, and we will explore the effect of augmentation on attribution in **??**.

B.3 Dataset Statistics

Table 5 shows statistics of training and instruction tuning data for both BIOCITE and Wikipedia.

	Size
Pretraining	
#documents	100K
#facts/sents	408K
#tokens	5.7M
avg. sents per doc	4.1
avg. tokens per doc	56.9
Instruction tuning	
#examples	186K
#tokens	3.1M

Table 5: Statistics for the BIOCITE before data augmentation. After data augmentation with $N_{\text{aug}} = 3$, the number of pretraining tokens goes up to 17.1M.

C METHOD DETAILS

C.1 Chain-of-Thought Attribution

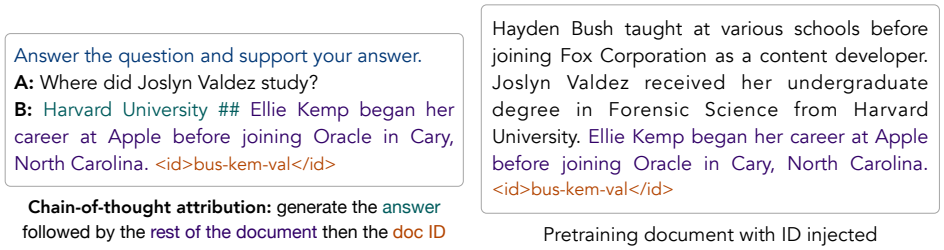

Figure 5: Example of chain-of-thought attribution. During both training and inference, the model cites the remaining part of the document before generating the doc ID.

D EXPERIMENTAL DETAILS

The pretraining corpus is split 50-50 into in-domain and OOD subsets, respectively. Training is done over 80% of the in-domain question, and we show performance in the remaining 20K. OOD evaluation is performed over 20K questions randomly sampled from the OOD documents. The QA performance is evaluated using the token exact match (EM) with the gold answer. During inference, we prompt the model and let it generate a response first, then append the special token `<id>` and continue decoding until the model generates the `</id>` token. We use constrained beam search Cao et al. (2021); Tay et al. (2022) to force the model to generate doc IDs that appeared in the pretraining data.

We evaluate attribution by measuring whether the cited document supports the question-answer pair. Precisely, we measure the gold document ID recall over cases where the answer is correct, where recall is evaluated using Hits@k with $k \in \{1, 10\}$, which measures whether the gold ID is in the top k beams. To monitor the impact of our attribution training on the model quality, we monitor the perplexity over Wikitext-v2 (Merity et al., 2017) during training, as done in previous work (Radford et al., 2019). The model we use for all experiments is TinyLLama 1.1B (Zhang et al., 2024),[7] which we pretrain for 10 epochs with a learning rate of 8×10^{-5} and instruction-tuning for 3 epochs with a learning rate of 1×10^{-5}. During both pretraining and fine-tuning, we apply a linear decay scheduler and use a batch size of 128, a weight decay of 0.02, and a learning rate warm-up of one epoch.

[7] huggingface.co/TinyLlama/TinyLlama-1.1B-intermediate-step-1431k-3T

E Additional Analysis

Figure 6 shows OOD answer EM and Hits@1 with different data augmentation strategies. Document-level augmentation seems necessary for OOD attribution.

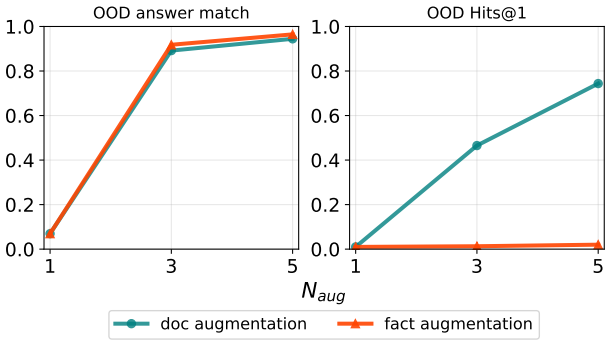

Figure 6: The effect of document- and fact-level augmentation on OOD answer match and Hits@1. $N_{\text{aug}} = 1$ means no augmentation applied. We show results when training with the doc ID injection strategy REPEAT. Document-level augmentation is necessary for OOD attribution.