

CiteFix: Enhancing RAG Accuracy Through Post-Processing Citation Correction

Harsh Maheshwari
mahhars@amazon.com

Srikanth Tenneti
stenneti@amazon.com

Alwarappan Nakkiran
nakkiran@amazon.com

Abstract

Retrieval Augmented Generation (RAG) has emerged as a powerful application of Large Language Models (LLMs), revolutionizing information search and consumption. RAG systems combine traditional search capabilities with LLMs to generate comprehensive answers to user queries, ideally with accurate citations. However, in our experience of developing a RAG product, LLMs often struggle with source attribution, aligning with other industry studies reporting citation accuracy rates of only about 74% for popular generative search engines. To address this, we present efficient post-processing algorithms to improve citation accuracy in LLM-generated responses, with minimal impact on latency and cost. Our approaches cross-check generated citations against retrieved articles using methods including keyword + semantic matching, fine tuned model with BERTScore, and a lightweight LLM-based technique. Our experimental results demonstrate a relative improvement of 15.46% in the overall accuracy metrics of our RAG system. This significant enhancement potentially enables a shift from our current larger language model to a relatively smaller model that is approximately 12x more cost-effective and 3x faster in inference time, while maintaining comparable performance. This research contributes to enhancing the reliability and trustworthiness of AI-generated content in information retrieval and summarization tasks which is critical to gain customer trust especially in commercial products.

1 Introduction

Recent advancements in AI infrastructure and methodologies have enabled training Large Language Models (LLMs) over internet-scale data. These models demonstrate impressive competence in answering a wide range of general queries. However, when applied to specialized domains such as addressing questions based on internal company

documents, off-the-shelf LLMs exhibit significant limitations. They often lack access to latest information, have difficulty interpreting domain specific language, struggle with source attribution, are prone to hallucinations (Ji et al., 2023), and are prone to overly broad responses.

To overcome these challenges, two broad strategies have emerged. The first involves fine-tuning LLMs on domain-specific data. However, this approach is not only resource-intensive and requires frequent updates, but also risks unintended consequences such as catastrophic forgetting, where the model loses previously acquired general knowledge, thereby increasing the overall system complexity. The second, often more practical method is Retrieval-Augmented Generation (RAG). RAG is a process that combines information retrieval with text generation. It typically involves the following steps: (1) indexing a knowledge base of relevant information, (2) using a retrieval system to find content specifically relevant to a given user query, (3) providing the user query and the retrieved content to an LLM, instructing it to generate a response based on the retrieved content. RAG offers numerous benefits, including real-time access to up-to-date information, improved token generation (Khandelwal et al., 2019), reduced hallucinations, better source attribution (Gao et al., 2023a; Hsu et al., 2024) and overall superior response generation (Shuster et al., 2021; Béchar and Ayala, 2024). Additionally, RAG tends to be more cost-effective and transparent than full model fine-tuning. Examples of RAG-based products include Perplexity.ai (Perplexity AI, 2024), Bing search, GPT Search etc.

Despite enabling a novel information retrieval experience for users, RAG systems today face key limitations. Table 1 illustrates results of a Subject Matter Expert based auditing of a RAG based system. Shown is a metric "Relative Mean Question Level Accuracy", which captures relevancy of cited chunks, correctness and completeness of the

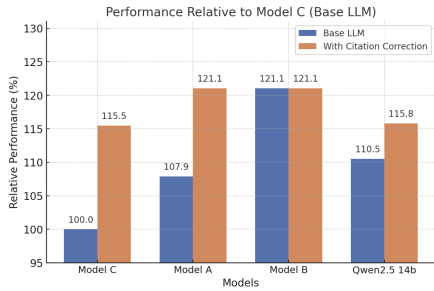


Figure 1: Improvements in RAG accuracy for various LLMs after employing our proposed citation correction methods. Results are shown as percentage improvements in Mean Question Level Accuracy (MQLA) over Model C baseline performance without citation correction. MQLA is a metric designed to capture relevancy, correctness and completeness (see Sec. 4.1).

answer (Sec. 4.1), relative to Model C¹ accuracy. A prevalent form of error that contributes to lower performance is that of unverifiable facts in LLMs’ responses. Unverifiable facts are the facts in LLM response which cannot be validated by cited reference. In our analysis for Model C, notably around 80% of these unverifiable facts were not pure hallucinations, but rather errors in the model’s ability to cite the correct reference from which it generated the given factual point. These observations align with industry studies (Liu et al., 2023) reporting citation accuracy rates of only about 74% for popular generative search engines. Incorrect citations not only reduce the actionability of the responses, but also dent customer trust, especially for commercial products. This paper focuses on this issue and proposes methods to address it.

While previous studies have explored attributable text generation ((Nakano et al., 2022); (Gao et al., 2023b)) and simple prompting techniques for citation incorporation ((Malaviya et al., 2024; Sun et al., 2024; Li et al., 2024)), systematic evaluations reveal significant performance gaps (Gao et al., 2023b). Recent work (Huang et al., 2024) has only scratched the surface by demonstrating attribution quality degradation from ChatGPT to Llama 2 7B, leaving a critical need for deeper analysis and practical solutions.

This paper offers two contributions:

¹Model names are anonymized following standard practice for proprietary/pre-release models. Publicly available models retain their original names. Model A, Model B and Model C are sufficiently large and powerful language model. With number of parameters in decreasing order for A, B and C. Model B however is the model trained on latest data with better methodologies

Model	Cents per 1K O/P tokens	Relative Mean Question Level Accuracy	% of factual points unverifiable	% of factual points incorrectly cited	% of factual points purely hallucinated
Model A	+1100%	+7.9% (+12%)	Base (Base)	90.8% (65%)	9.1% (35%)
Model B	+220%	+21.1% (+21.1%)	Base (Base)	66.6% (66.6%)	34.4% (-33.4%)
Model C	Base	Base (+15.4%)	Base (Base)	80.6% (33.3%)	19.4% (-66.6%)
Qwen 1.4-B	open source	+10.5% (+15.8%)	Base (Base)	76.2% (70.8%)	-13.8% (29.2%)
Qwen 2-B	open source	-39% (NA)	NA	NA	NA

Table 1: *Motivating the need for CiteFix*: This table shows the prevalence of incorrect citations across LLMs and our method’s impact. Model C is the baseline for cost and accuracy columns. For the last three columns, the baseline is each model’s total percentage of unverifiable factual points. Numbers outside (inside) parentheses show performance before (after) CiteFix. Initially, incorrect citations significantly outnumber hallucinations. CiteFix balances this ratio and in absolute terms it drastically reduces incorrect citations. Qwen 2-B was excluded from detailed audit due to inconsistent citation generation.

1. Demonstrating the existence and extent of the incorrect citations issue across multiple LLMs, and highlighting the need to address the same.
2. Proposing six computationally light weight methods to address this issue, ranging from simple heuristic methods to more sophisticated learning-based solutions. Through extensive experimentation, we show that different citation correction approaches may be optimal for different LLMs - for instance, hybrid (lexical + semantic) matching works best with Model A, while fine-tuned BERTScore performs better with Model B. We provide detailed comparisons of their effectiveness and practical applicability. As seen in Fig 1 and Table 1, our method resulted in an improvement of upto 15.46% relative improvement in accuracy when tested across four different LLMs.

Through this work, we aim to not only advance the understanding of citation accuracy challenges in LLMs, but also provide practical low cost solutions for improving attribution in real-world applications. Sec. 2 presents an overview of related work. Sec. 3 details our proposed algorithms. Sec. 4 presents evaluation results. Sec. 5 concludes, along with a discussion of the limitations of our work and plans for addressing them going forward.

2 Related Work

Accurate attribution of information to sources remains a critical challenge in building trustworthy AI systems, particularly for Large Language

Models (LLMs) and Retrieval-Augmented Generation (RAG) systems. The challenge of accurate attribution in AI-generated content has been approached from multiple angles in the literature. Some researchers have focused on developing models specifically designed for attributable text generation (Nakano et al., 2022), while others have explored the effectiveness of prompt engineering techniques for citation accuracy (Malaviya et al., 2024; Li et al., 2024). However, a comprehensive study (Gao et al., 2023b) has highlighted that significant challenges remain, particularly in maintaining consistent attribution accuracy across different types of queries and document structures. These findings underscore the need for more robust and versatile approaches to citation/attribution in AI systems.

Recent work has focused on the automatic evaluation of attribution by LLMs (Yue et al., 2023) and factual entailment for hallucination detection (Rawte et al., 2024), primarily assessing whether generated content is present in cited references. However, there is a notable gap in research specifically addressing citation correction.

Many existing methods, including those fine-tuning T5 models (Gao et al., 2023c; Song et al., 2024; Honovich et al., 2022), are limited by context lengths of around 512 tokens. This constraint poses significant challenges when dealing with longer documents or multiple sources, which is often the case in practical RAG systems. Our proposed solution for citation correction is designed to handle larger context lengths, addressing a critical limitation in current approaches.

Furthermore, our research distinguishes itself by focusing on not just detecting citation errors but actively working towards correcting them. This shift from identification to correction represents a significant step forward in improving the usefulness of AI-generated content in RAG systems. We introduce a range of citation correction methods, including lexical matching, hybrid (lexical + semantic) approaches, and lightweight LLM-based attribution. One method builds on BERT Score (Zhang et al., 2020), leveraging pre-trained contextual embeddings from BERT (Devlin et al., 2019). Initial experiments with an off-the-shelf model (Beltagy et al., 2020) showed improvements, but fine-tuning on in-domain data yielded better results. This led us to explore ColBERT (Khattab, 2020), a neural retrieval model designed for fine-grained context-

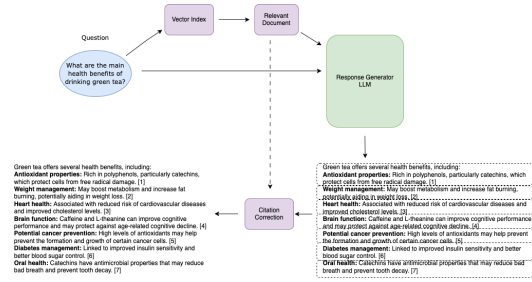


Figure 2: Overview of the workflow of the proposed methods using a sample question. Once the RAG system’s response generating LLM generates an answer, we split the answer into distinct factual points (shown in dotted lines above). For each factual point, we use its similarity scores with the retrieved documents to detect citation errors and correct them. See Section 3 for details. Question used is for illustration purpose only

tual late interaction. By combining BERT Score’s semantic similarity assessment with ColBERT’s fine-tuning capabilities, we developed a more robust and accurate citation correction method, which we detail in the next section. We detail these methods in the next section.

3 Proposed Methodology

Our goal is to improve the overall citation accuracy, while having minimal impact on latency and costs. Towards this, we propose a suite of algorithms that leverage various techniques, ranging from simple heuristics to sophisticated machine learning models. Our algorithms are streaming-compatible post-processing techniques, meaning that they operate on an LLM’s response as it is being generated.

The general framework of our proposed methods is depicted in Figure 2. We will now go into its details. Let us denote the query that the user asks the RAG system as q . Let the set of documents retrieved by the Retriever module in RAG be $\{\hat{x}_i\}_{i=0}^{R-1}$. Let A denote the answer generated by the LLM. Our algorithms involve the following steps:

1. We first split the LLM’s response A into distinct "factual points" $\{x_i\}_{i=0}^{L-1}$. A factual point is defined as a section within A that the LLM attributes to a particular set of retrieved documents via citations. In our use case, the LLMs were instructed to include citations at the end of each factual statement in their response. We use simple regular expressions to segment the LLM’s response into "factual points", delimited by citations. See Fig. 2 for example.

- Let C_i be the number of citations in the LLM’s generated response A for the factual point x_i . Our algorithms will estimate the "corrected citations" to be the top C_i retrieved documents among $\{\hat{x}_i\}_{i=0}^{R-1}$ that maximize the following similarity metric with the factual point x_i :

$$s_{ij} = f(x_i, \hat{x}_j) \quad (1)$$

In the next sections, we will discuss various choices for the function f in Eq. 1. We will use the following notation: Let us denote each factual point x_i as list of its individual tokens t_{ij} . Namely, $x_i = [t_{i0}, t_{i1}, \dots, t_{ik}]$. Let us also denote each retrieved document \hat{x}_i as a list of its tokens $\hat{x}_i = [\hat{t}_{i0}, \hat{t}_{i1}, \dots, \hat{t}_{il}]$.

3.1 Keyword based matching

We define f in Eq. 1 as the size of the intersection between the tokens in x_i and \hat{x}_j . We also tried a term-frequency (TF) by inverse-document-frequency type of scoring, such as done in traditional document ranking (Rousseau and Vazirgianis, 2013; Trotman et al., 2014), but it did not yield good results. We noticed regular IDF being particularly noisy with domain specific keywords such as "yield" which have different meaning in agriculture and financial context or "drill" which have different meaning in mining and military context etc.

3.2 Keyword + Semantic Context based matching

In this approach, we combine the above keyword match score with a mild contribution from the semantic similarity between the user query q and the retrieved document \hat{x}_i . The motivation is to mildly prefer retrievals that are more relevant to the user query:

$$f(x_i, \hat{x}_j) = \lambda \cdot f_{keyword}(x_i, \hat{x}_j) + (1 - \lambda) \cdot r(q, \hat{x}_j) \quad (2)$$

Where $f_{keyword}(x_i, \hat{x}_j)$ is the keyword based matching score and $r(q, \hat{x}_j)$ is the retrieval score for document \hat{x}_j given query q . We empirically found $\lambda = 0.8$ to perform well in our experiments.

3.3 BERT Score

In the previously discussed approaches, contextual meaning of the words in x_i and \hat{x}_j was not fully utilized. They also do not differentiate between cases where word matches occur in close proximity within the reference versus where they are scattered

across unrelated positions. Additionally, keyword-based methods struggle to handle scenarios where the language model or response generator paraphrases the words, as these methods rely on exact word matches.

BERT Score (Zhang et al., 2020) addresses these limitations by leveraging contextual embeddings to represent the tokens in the factual point x_i and the reference \hat{x}_j . These embeddings are generated using the LongFormer model (Beltagy et al., 2020), which incorporates bi-directional attention to capture not only the token but also its surrounding context.

Once the embeddings are computed, the similarity between the factual point and a retrieved document is calculated as follows: For each token in the factual point x_i , we compute its maximum similarity among all tokens in the retrieved document. The mean of these maximum similarity scores among all tokens in x_i is used as the final score in Eq 1:

$$f(x_i, \hat{x}_j) = \frac{1}{|x_i|} \sum_{t_{il} \in x_i} \max_{\hat{t}_{jk} \in \hat{x}_j} e(t_{il})^\top e(\hat{t}_{jk}) \quad (3)$$

where $e(t)$ denotes the embedding of a token t .

3.4 Fine-tuned Model with Bert Score

While off-the-shelf BERTScore models provide a good starting point for incorporating contextual similarity into the citation correction process, we hypothesize that fine-tuning these models specifically for this task on an in-domain dataset can further improve their performance. The key limitation of the off-the-shelf models is that they are not explicitly trained to capture the nuances of citation attribution & factual entailment. Our methodology is motivated by ColBERT (Khattab, 2020).

During training, the input to the model is a factual point (x), a positive reference ($\hat{x}+$) that validates the point, and a negative reference ($\hat{x}-$) that does not validate the factual point. BERTScore for the factual point, calculated using Eq. 3, is maximized for the positive reference compared to the negative reference. We used cross-entropy loss to increase the score with the positive reference compared to the negative reference.

Dataset Preparation: To train the model, we need factual points, and corresponding positive and negative references. We employed an LLM for this, using two strategies: First, for each document in

the corpus, we determine the n^{th} most similar document using (Amazon-Titan-V2, 2024). We then prompt LLM to provide a factual point present in the former document, but not in the latter. By varying $n \in \{14, 11, 8, 5, 4, 3\}$, we get progressively hard positive and negative pairs for training. Secondly, for a list of questions, we generate answers from our RAG-based system. For each factual point present in the answer and for each retrieved document, we employ an LLM to check for if the former is grounded in the latter. We then use this information to create multiple pairs of positive-negative for a given factual point. This allows us to tune the model specifically for the citations issue for the specific LLM used within the RAG system.

3.5 LLM Based Matching

An alternative approach for citation correction is to employ an LLM directly. Table 1 presents results using our best-performing prompt instructions for citation-aware response generation. Here, we explore a secondary LLM that identifies the most relevant reference for each factual point.

To balance accuracy with efficiency, we use a simple prompt that requests only the reference number, avoiding complex techniques like Chain of Thought (CoT) (Wei et al., 2023), which would increase token usage, latency, and cost. This approach leverages the LLM’s ability to capture contextual and semantic nuances beyond keyword-based or rule-based methods, enabling adaptability across domains without explicit rule-crafting or fine-tuning.

However, the effectiveness of this method depends on the LLM’s quality, training data, and prompt design. Additionally, processing each factual point individually introduces computational overhead, requiring a careful trade-off between cost, latency, and accuracy.

3.6 Reusing Attention Maps of the Base LLM

The main idea here is, can we look at the attention maps of the response generating LLM itself to check which retrieved documents were used in generating each factual point in the response. We did not have enough time to fully experiment with this idea, but in Appendix 6.1, we show a simple proof of concept that demonstrates this idea. We will explore this further in our future work.

4 Results

In this section, we will present evaluation results of all the proposed methods on top of RAG based system. The evaluations were done by human auditors, who have prior knowledge on the topic for which RAG is used.

4.1 Metrics

We developed the following metrics to evaluate RAG system performance. The uber level metric we track is called "Mean Question-Level Accuracy" (MQLA). It combines the following:

- **Relevancy URL:** Checks if the set of citations referenced to by the LLM are relevant to the question. Calculated as the fraction of cited URLs that are relevant.
- **Relevancy Keywords:** Checks if keywords in the LLM’s response are relevant to the question. Calculated as the ratio of keywords which are relevant by the total number of keywords present in the query. The keywords in the response are identified by humans.
- **Relevancy Facts:** Checks if facts present in the LLM’s response are relevant to the question. Calculated as the ratio of facts which are relevant to query by the total number of facts present in the response. The facts in the response are identified by humans.
- **Correctness:** Checks if the facts present in the LLM’s response can be verified in the citations provided. Calculated as the ratio of number of facts supported by cited references and the total number of facts. **Note:** The facts not supported by cited referenced can be divided into two categories 1) Hallucinated facts and 2) Incorrectly cited facts, based on whether the fact was present in any of the retrieved documents or not.
- **Completeness:** Checks if all aspects (possible sub-questions) of the original questions are addressed in the response. The possible sub-questions are identified by the humans.

We calculate MQLA as described in Algorithm 1.

4.2 Comparing Different Citation Correction Methods

In Table 2, we compare different citation correction algorithms proposed in this paper on Model

Table 2: Comparing Citation Correction Methods. All columns except p90 latency show relative performance

Citation Correction Method	Response Generating LLM	Mean Question Level Accuracy	Relevancy URL	% of Facts Correctly Cited	p90 latency per factual point (in sec)
None	Model C	Base	Base	Base	-
Keyword	Model C	+12.7%	-0.9%	+12%	0.014
Keyword + Semantic Context	Model C	+15.5%	-0.9%	+13.6%	0.015
BERT Score	Model C	+2.6%	-1%	+3.2%	0.389
Finetuned BERT Score	Model C	+15.8%	+1.5%	+13.7%	0.389
LLM Based Matching (Model C)	Model C	+1.9%	+0.9%	+7%	1.586
None (Baseline)	Model A	+7.8%	+2%	+5.4%	-

Algorithm 1 Mean Question Level Accuracy

```

1: Initialize totalAccuracy= 0, n = number of
   questions
2: for q in questions do
3:   Initialize accuracy= 0
4:   if all(relevancyUrl, relevancyKeyword, rel-
   evancyFacts, correctness, completeness  $\geq$ 
   0.8) and hallucinatedFacts  $\leq$  1 then
5:     accuracy= 1
6:   end if
7:   totalAccuracy + = accuracy
8: end for
9: meanAccuracy = totalAccuracy / n
10: return meanAccuracy

```

C’s responses. We used a set of 50 representative questions for evaluation, incurring an audit time of 2.5 days by 2 humans per row of Table 2. The table includes p90 latency per factual point for each citation correction method, which adds negligible overhead (except LLM method) to our system’s time to first token p90 latency. The latency is computed on g5.4xlarge instance. Results for Model A, a model that is 12x more expensive and about 3x slower are also shown for reference. The impact of our techniques Keyword + Semantic Context based and Fine-tuned BERT Score is evident, taking Model C’s MQLA higher than Model A.

4.3 Evaluating Impact Across Different LLMs

In Table 3, we evaluated the two best performing citation correction methods from Table 2 for four different LLMs (using the same dataset as in Sec. 4.2). Interestingly, different LLMs may pair optimally with different citation correction strategies. The impact of our methods is strongly evident for Model C, Model A and Qwen 2.5 14-B. Model B seems to be inherently much better at citations, but we see some mild improvements in the relevancy of cited URLs when paired with our fine-tuned BERT Score method. These results demonstrate potentially wide applicability of our proposed methods.

Table 3: This table shows the effectiveness of our two best citation correction approaches with various LLMs. KSC represents Keyword+Semantic context and FBS represents Finetuned BERT Score

Response Generator	Citation Corrector	MQLA	Relevancy URL	% of facts Correctly Cited
Model C	None	base	base	base
Model C	KSC	+15.5%	-0.9%	+13.6%
Model C	FBS	+15.8%	+1.5%	+13.7%
Model B	None	+21%	+1.5%	+14.9%
Model B	KSC	+10.5%	+1.5%	+10.7%
Model B	FBS	+21%	+2%	+15%
Model A	None	+7.9%	+2%	+5.4%
Model A	KSC	+21%	-1.3%	+16%
Model A	FBS	+10.5%	+2%	+9.8%
Qwen 2.5 14b	None	+10.5%	+2%	+8.4%
Qwen 2.5 14b	KSC	15.8%	+2%	+9.7%
Qwen 2.5 14b	FBS	13.1%	+1.3%	+8.7%

5 Conclusion

This paper addresses the critical challenge of citation accuracy in RAG systems, demonstrating its impact across multiple LLMs and its effect on AI-generated content trustworthiness. Our key contribution is the development of efficient post-processing algorithms for citation correction, improving relative accuracy by up to 15.46% while maintaining minimal computational overhead. Notably, we found that optimal citation correction methods vary across LLMs, emphasizing the importance of model-specific approach selection.

Our findings, while promising, represent early steps in addressing this challenge. Future research areas include exploring attention-map-based methods for more precise attributions and developing sophisticated dataset preparation techniques. While newer LLMs (Model B) have improved citation accuracy, attribution issues persist to a lesser extent, suggesting the need for more sophisticated correction algorithms. Additionally, our framework’s ability to establish relationships between factual points and source documents opens up interesting applications, such as determining appropriate contexts for content insertion (e.g., advertisement placement) based on document similarity and factual relevance.

References

- Amazon-Titan-V2. 2024. [Amazon titan foundation models](#). Accessed: 2025-01-15.
- Patrice Bécard and Orlando Marquez Ayala. 2024. Reducing hallucination in structured outputs via retrieval-augmented generation. *arXiv preprint arXiv:2404.08189*.
- Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. [Longformer: The long-document transformer](#).
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [Bert: Pre-training of deep bidirectional transformers for language understanding](#).
- Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023a. Enabling large language models to generate text with citations. *arXiv preprint arXiv:2305.14627*.
- Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023b. [Enabling large language models to generate text with citations](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6465–6488, Singapore. Association for Computational Linguistics.
- Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023c. [Enabling large language models to generate text with citations](#).
- Or Honovich, Roei Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. [True: Re-evaluating factual consistency evaluation](#).
- I Hsu, Zifeng Wang, Long T Le, Lesly Miculicich, Nanyun Peng, Chen-Yu Lee, Tomas Pfister, et al. 2024. [Calm: Contrasting large and small language models to verify grounded generation](#). *arXiv preprint arXiv:2406.05365*.
- Chengyu Huang, Zeqiu Wu, Yushi Hu, and Wenya Wang. 2024. [Training language models to generate text with citations via fine-grained rewards](#).
- Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.
- Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2019. Generalization through memorization: Nearest neighbor language models. *arXiv preprint arXiv:1911.00172*.
- Omar Khattab. 2020. [Colbert: Efficient and effective passage search via contextualized late interaction over bert](#).
- Xinze Li, Yixin Cao, Liangming Pan, Yubo Ma, and Aixin Sun. 2024. [Towards verifiable generation: A benchmark for knowledge-aware language model attribution](#).
- Nelson F. Liu, Tianyi Zhang, and Percy Liang. 2023. [Evaluating verifiability in generative search engines](#).
- Chaitanya Malaviya, Subin Lee, Sihao Chen, Elizabeth Sieber, Mark Yatskar, and Dan Roth. 2024. [Expertqa: Expert-curated questions and attributed answers](#).
- 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. 2022. [Webgpt: Browser-assisted question-answering with human feedback](#).
- Perplexity AI. 2024. [Perplexity AI: AI-powered search engine](#). Accessed: November 21, 2024.
- Vipula Rawte, S. M Towhidul Islam Tonmoy, Krishnav Rajbangshi, Shravani Nag, Aman Chadha, Amit P. Sheth, and Amitava Das. 2024. [Factoid: Factual entailment for hallucination detection](#).
- François Rousseau and Michalis Vazirgiannis. 2013. Composition of tf normalizations: new insights on scoring functions for ad hoc ir. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 917–920.
- Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. *arXiv preprint arXiv:2104.07567*.
- Maojia Song, Shang Hong Sim, Rishabh Bhardwaj, Hai Leong Chieu, Navonil Majumder, and Soujanya Poria. 2024. [Measuring and enhancing trustworthiness of llms in rag through grounded attributions and learning to refuse](#).
- Hao Sun, Hengyi Cai, Bo Wang, Yingyan Hou, Xiaochi Wei, Shuaiqiang Wang, Yan Zhang, and Dawei Yin. 2024. [Towards verifiable text generation with evolving memory and self-reflection](#).
- Andrew Trotman, Antti Puurula, and Blake Burgess. 2014. Improvements to bm25 and language models examined. In *Proceedings of the 19th Australasian Document Computing Symposium*, pages 58–65.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. [Chain-of-thought prompting elicits reasoning in large language models](#).
- Xiang Yue, Boshi Wang, Ziru Chen, Kai Zhang, Yu Su, and Huan Sun. 2023. [Automatic evaluation of attribution by large language models](#).
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. [Bertscore: Evaluating text generation with bert](#).

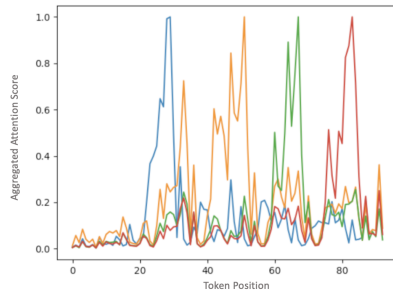


Figure 3: Visualisation of Attention Score. See Appendix 6.1 for details.

6 Appendix

6.1 Using Attention map for attribution

In a RAG based system, the response generating LLM is given a set of relevant document in response to a user query. It then understands information from these different documents to answer the question at hand. Here, we want to explore if can we leverage attention scores within the LLM to understand which document in the prompt it is focusing on while generating a particular fact in its response. We did a small *toy experiment* with Qwen 2.5B - 2B to test the same. We use the below prompt:

```
Hi, you are an assistant who has access to the
following <documents> about cricket.
Please answer the <user query> at the end
using only the information provided in
the <documents>. Do not output any information
not contained in the <documents>.
Do not output any information that is not
relevant to answering the <user query>.
If the <user query> cannot be answered with
the given <documents>, please say so.

<documents>

<doc> Axx is a tall batsman. </doc>

<doc> Byy can bat with a broken bat as well.
</doc>

<doc> Czz is a very funny umpire. </doc>

<doc> Dii is a fast bowler from Mumbai. </doc>

</documents>

<user query>
QUESTION
</user query>
```

We asked the following questions to the LLM:

- Name a batsman who is not particularly short
- Name a batsman who can bat with a damaged bat
- Name an umpire who makes people smile

- Who is a player from Mumbai?

and visualised the attention scores in 3 (Blue, Orange, Green and Red lines for the above four questions respectively). The x-axis in the figure is the token position within the prompt. The y-axis is the sum of the attentions scores for all tokens in the output, across all layers of the LLM at that particular input token location. A higher value of this sum at a particular location of the input token indicates that that input token was taken into account by the LLM in generating the response.

You will see that for first question the peak of attention score is before the second question which is in line with where the necessary information is present in the prompt. Likewise, the peak of attention for second question is before the third one, and so on. This small proof of concept shows that we may be able to leverage the LLM's internal attention maps to correct citations.