
Can backward models improve citation?

Ivan Wiryadi

Carnegie Mellon University
Pittsburgh, PA 15213
iwiryadi@andrew.cmu.edu

Janbol Jangaby

Carnegie Mellon University
Pittsburgh, PA 15213
jjangaby@andrew.cmu.edu

Jihu Hwang

Carnegie Mellon University
Pittsburgh, PA 15213
jihuh@andrew.cmu.edu

Xuanyi Shen

Carnegie Mellon University
Pittsburgh, PA 15213
xuanyis@andrew.cmu.edu

Abstract

Large Language Models (LLMs) have shown remarkable capabilities in text generation, however it may still be prone to hallucination or producing inaccurate results. To alleviate this, one possible solution is to use citation. Time-Reversed Language Model (TRLM) by Varun et al. [2025] proposes a backward prediction framework, where models infer queries from responses or trained with a reversed token order. It is shown to have better performance in citation attribution. In this work, our objective is to validate the improvements in citation attribution performance of TRLM using alternative but similar settings. We used Pythia-160m and its reversed variant, performed fine tuning using FLAN dataset, and implemented three citation search algorithms on the CNN-Daily Mail dataset. We evaluated attribution performance using embedding similarity, TF-IDF, and ROUGE. Our results confirm that inference direction produced substantially better performance gains; however, contrary to the original findings, the reverse-trained model in our experiment did not perform significantly better than the forward-trained model. We conclude by noting our limitations and suggesting several future work areas.

1 Overview

Language models have demonstrated remarkable capabilities in text generation across various domains. However, their probabilistic nature often leads to the generation of content that are coherent but may lack factual accuracy. This phenomenon is commonly referred as "hallucination". More formally, "hallucination" is defined as a condition where generated information is either in conflict with the existing resource (intrinsic hallucination) or cannot be verified with the available resource (extrinsic hallucination) [Ji et al., 2022, Zhao et al., 2025].

In certain use cases, for example in healthcare, law, or research, accuracy and reliability are paramount. Without these traits, users risk producing inaccurate work based on fabricated information. Therefore, having a language model that is able to be factually accurate is important. This raises the motivation to explore, how may we improve language models factual accuracy capabilities?

One approach to mitigate hallucination and enhance reliability is through citation generation. By providing citations, users can verify the factual basis of generated content through cross-referencing original sources. Commercial applications like Perplexity¹ have implemented this approach by augmenting responses with attributable information for user verification.

¹<https://www.perplexity.ai>

A recent work by Varun et al. [2025] has shown that backward or reversed language models are able to improve citation accuracy. In this project, we attempt to further examine reversed language models capability to improve citation.

2 Related Work and Background

2.1 Citation

There are two fundamental approaches to generate citations: parametric and non-parametric. Parametric approaches require the model to learn sources through its training data, internalizing knowledge about facts and their sources within model parameters. Non-parametric approaches, by contrast, create hybrid systems that separates the retrieval from generation [Li et al., 2024].

A non-parametric citation generation system typically may consist of the following components:

1. Identifying when a claim requires citation - determining which statements need supporting evidence
2. Retrieving relevant source material (document retrieval) - finding documents that contain the information needed to support the claim
3. Attributing citation to the relevant source (citation attribution) - correctly linking specific claims to their source documents

In this work, we focus on citation attribution.

2.2 Citing in Reverse

Recent work by Varun et al. [2025] introduced Time-Reversed Language Models (TRLM). The authors designed three variants of TRLMs:

- TRLM-Fo: A forward-trained model prompted to operate in reverse through appropriate conditioning prompts (e.g., "Generate a question that would result in the following answer").
- TRLM-Ba: A model pre-trained from scratch in reverse token order, processing information from right to left. This model receives tokens in reversed order during both training and inference, allowing it to naturally predict preceding tokens given subsequent ones. From the authors experiments, TRLM-Ba consistently shows the strongest performance in citation attribution tasks.
- TRLM-FoBa: A hybrid model trained in both forward and reverse directions, capable of operating in either mode. This versatility allows it to leverage both perspectives, though with a modest computational overhead during training.

For citation attribution, TRLM variants demonstrate significant advantages over traditional forward-only models. When tested on the CNN-Daily Mail dataset, the authors reported significant improvement (34-44%) in citation attribution accuracy using TRLM-Ba compared to their baseline forward approach.

The effectiveness of TRLMs in citation tasks stems from how they approach the information direction problem. In citation attribution, the model must match a high-information document (source) with a lower-information summary or claim (response). Forward prompting ($S \rightarrow A$) struggle with this information asymmetry, while reverse prompting ($A \rightarrow S$) excel by evaluating how likely a document is to generate a particular claim [Varun et al., 2025].

The empirical result of TRLM may suggest that any language models trained or queried in the backward direction may have similar capability to improve citation. The authors have so far only used TRLM architecture which is based on PALM2-Otter. Therefore, we focus and frame our project question as follows: can backward models in general improve citation attribution?

3 Methodology

3.1 Overview

There are two objectives in our investigation of reverse language models for citation. First, we attempt to evaluate whether alternative reverse language model architectures can reproduce the citation attribution improvements demonstrated in the TRLM paper. Second, we will also attempt to explore how instruction tuned language models can improve citation attribution compared to their pre-trained variants. In doing these experiments, we attempt to replicate the model setup, scoring mechanism, and datasets as closely with what is described in Varun et al. [2025].

3.2 Models Selection

To examine the findings of Varun et al. [2025], we need to have two models that are identical except for the order of training tokens. We found two open source pre-trained models that fits this criteria:

1. Pythia 160 million, which is part of the Pythia Suite, designed to study model behavior across scales and training stages Biderman et al. [2023]. The model are trained normally (forward tokens).
2. Reverse Pythia, a 160 million parameter model that was trained in similar configuration as Pythia 160 million in Pythia suite, with the main difference of reverse the token for the training set. It is reported to have similar average loss with the forward model Pfau et al. [2023].

Based on these two models, we create 5 experiment setups:

1. Baseline (Base), which is the Pythia 160 million model and perform scoring or inference in the typical forward direction (query to response).
2. Vanilla Forward (VFo), is the same Pythia 160 million model. The difference between this model and Base is that during attribution scoring it will be performed in reversed direction (response to query). This corresponds to the (TRLM-Fo) model.
3. Vanilla Backward (VBa), which is the Reverse Pythia model. During attribution scoring it will also be performed in reverse. This corresponds to the (TRLM-Ba) model.
4. Fine-tuned Forward (FTFo), which is VFo fine tuned with FLAN dataset. This corresponds to TRLM-Fo that is instruction-tuned.
5. Fine-tuned Backward (FTBa), which is VBa fine tuned with FLAN dataset. This corresponds to TRLM-Ba that is instruction-tuned.

3.3 Dataset

There are two key datasets that we will use in this set of experiment.

1. CNN-Daily Mail Dataset [See et al., 2017, Hermann et al., 2015], a dataset comprising of over 300.000 news articles from CNN and Daily Mail. Each articles have several highlight or summarized sentences related to it. The preprocessing steps are generally described in section 3.4, where it is basically combining the texts with conditioning prompt before tokenizing.
2. FLAN [Longpre et al., 2023], a collection of dataset for instruction fine-tuning. Each data has an instruction and expected answer. The preprocessing steps are generally described in section 3.6.

3.4 Attribution Scoring

To find the optimal attribution between text passages, we follow the scoring approach outlined by Varun et al. [2025]. The log probability $\log P$ is calculated as the sum of token-wise log probabilities:

$$\log P(Q|A) = \sum_{i=1}^{|Q|} \log P(q_i|A, q_1, \dots, q_{i-1}) \quad (1)$$

Where q_i represents the i -th token in the query sequence and $|Q|$ is the length of the query in tokens. This decomposition follows from the autoregressive nature of language models, which predict each token conditioned on all previous tokens.

The core of the attribution method relies on computing a score that quantifies how likely a summary sentence could be derived from a specific article passage, based on the model’s conditional probability distribution. We implement Algorithms 2 and 3 from Varun et al. [2025], which provide methods for computing this score using both forward and backward models. This is expressed as:

$$\text{Score}_{\text{Fo}}(Q|A) = \log P_{\text{Fo}}(Q|A + \text{CP}) \quad (2)$$

$$\text{Score}_{\text{Ba}}(Q|A) = \log P_{\text{Ba}}(\text{Reverse}(Q)|\text{Reverse}(A + \text{CP})) \quad (3)$$

Where Q represents the query text (e.g., a summary sentence), A represents the answer text (e.g., an article passage), and CP is a conditioning prompt such as "is a summary of" or "has an answer to" depending on the task.

Note that the equations above illustrate a single-sample case. In our actual implementation, we first prepare (context, target) pairs by inserting conditioning prompts such as “is a summary of” or “is summarized by” between the highlight and each candidate sentence during the search phase. Then, in the scoring step (calculate_scores_batch), we batch-tokenize and pad all of these pairs at once, perform a single forward pass to obtain logits for the entire batch, and finally sum and normalize each sample’s log-probabilities to compute its perplexity. (See Alg. 4 in the Appendix)

3.5 Attribution Search Algorithms

To identify the optimal attribution between summary sentences and article passages, we implement three search algorithms as described by Varun et al. [2025]: Linear Search, Binary Search, and Exclusion Search. Each algorithm leverages the attribution scoring mechanism described in 3.4 but differs in its approach to searching through the article text.

3.5.1 Linear Search

Linear Search is a straightforward approach, which evaluates every possible sentence in the article against the highlight or summary sentence. The algorithm computes the score for each article sentence paired with the highlight sentence using the scoring function and selects the article sentence with the highest score as the attribution. (See Alg. 1 in the Appendix)

3.5.2 Binary Search

Binary Search takes a more efficient approach by recursively dividing the article into halves and evaluating which half is more likely to contain the appropriate attribution. The algorithm starts by comparing the scores of the first half against the second half of the article when paired with the highlight sentence. Based on which half scores higher, it recursively narrows down the search space until it reaches a contiguous set of article sentences with sufficient granularity. (See Alg. 2 in the Appendix)

3.5.3 Exclusion Search

Exclusion Search takes an inverse approach compared to Linear Search. Instead of scoring each article sentence directly, it removes one sentence at a time from the article and scores the remaining text against the highlight sentence. The algorithm selects the article sentence whose removal leads to the lowest score (indicating that its presence is most critical for attribution). (See Alg. 3 in the Appendix)

3.6 Fine Tuning Setup

To create instruction-tuned versions of our models, we fine-tuned using the FLAN dataset [Longpre et al., 2023], following an approach conceptually aligned with the TRLM pretrain functions described in Varun et al. [2025].

3.6.1 Dataset Selection and Processing

Given computational constraints, we opted to use a subset of the FLAN dataset. We applied several critical preprocessing steps:

1. We filtered out all CNN/Daily Mail data from the training set to avoid data contamination, as this dataset is used in our evaluation.
2. We implemented task diversity by using a probability parameter `TASK_DIVERSITY_P` (set to 0.1 or 0.2) that controls the inclusion of non-summarization tasks, balancing between summarization-focused training and general instruction-following capabilities.
3. For the forward model (FTFo), the data was processed in the standard token order following the conventional next-token prediction paradigm. Meanwhile for the reverse model (FTBa), we processed the data by reversing the token order as described in Varun et al. [2025], Pfau et al. [2023].

3.6.2 Finetuning Training Approach

Our fine-tuning approach used a standard causal language modeling objective with the following key characteristics:

1. **Loss Function:** We employed the standard cross-entropy loss function for causal language modeling, where the model predicts the next token (for forward models) or previous token (for reverse models) given the context.
2. **Data Preprocessing:**
 - For FTFo, we maintained the standard format where the model predicts token x_t given tokens $x_1, x_2, \dots, x_{t-1}$.
 - For FTBa, we reversed the token order so that the model predicts token x_{T-t} given tokens $x_T, x_{T-1}, \dots, x_{T-t+1}$, where T is the sequence length.
 - Instruction-response pairs were formatted as continuous text with the instruction (merged with the query) preceding the response for forward models
 - For reverse models, the same content was used but with the token order reversed to maintain consistency with the pretraining approach
 - Input sequences were tokenized and padded/truncated to a maximum length of 1024 tokens

3.7 Metrics and Evaluation

To evaluate the quality of generated citations, we employ multiple complementary metrics that capture different aspects of alignment between source texts and their attributed citations.

- **Embedding Cosine Similarity** calculates the cosine of the angle between two vectors, measuring semantic similarity regardless of magnitude. Since Gecko embeddings (used in the original TRLM paper) are not publicly available, we will instead use SentenceTransformers [Reimers and Gurevych, 2019] to generate the embeddings to calculate cosine similarity.
- **TF-IDF** calculates relevance by assessing word importance by measuring the term frequency within a document against its rarity across a corpus.
- **ROUGE** [Lin, 2004] evaluates the longest common subsequence between texts. The paper did not describe which exactly which ROUGE metric to use, but we will use ROUGE-L.

4 Baseline

Our baseline approach uses Base (pre-trained Pythia model) as it aligns most with common practice for language models (normal ordered tokens for training and *query* $\rightarrow$ *response* direction) as described in Section 3.2. For the citation search algorithm, we will use Base with all three search algorithms (linear, binary, and exclusion) and compare with the rest of the experiments.

5 Results and Analysis

5.1 Finetuning Results

In our last fine tuning ablation study, we decided to build upon prior ablations and set up 6 configurations. The models that yielded the lowest loss on validation set were used later to perform the citation attribution tasks.

Table 1 shows that the forward direction consistently yields lower validation and training loss compared to the backward direction across all configurations. However, the starting loss for the FTBa model is also significantly higher than that of FTFo model.

Configuration 4 achieves the lowest forward validation error (2.52), indicating better generalization performance in that direction. Conversely, the backward direction has higher errors overall, with Configuration 5 showing the lowest validation error (4.27) among backward runs.

Compared to the pretraining loss of Pythia Vanilla, FTFo shows overall improvement when trained on the FLAN dataset subset, whereas the FTBa model performs worse. This discrepancy may be due to the parameter configurations used for the FTBa models, as the project aimed to maintain consistent ablation settings across both directions.

Table 1: Validation and Training Results by Direction

Configuration	Forward		Backward	
	Validation	Training	Validation	Training
Pythia Vanilla CE loss	2.69	–	2.62	–
Config 1	2.58	2.56	4.67	4.68
Config 2	3.70	3.66	4.58	4.58
Config 3	2.59	2.58	4.69	4.70
Config 4	2.52	2.52	4.64	4.68
Config 5	2.57	2.55	4.27	4.26
Config 6	2.64	2.62	4.33	4.35

In addition to the configurations above the project has implemented different ablations: You can find the code in our Github repository and our project ablations on WandB.

Figure 1 presents the evaluation loss during FTFo fine-tuning for five different configurations. `full-tuning-config4-restart` (orange) exhibits the best performance, achieving the lowest evaluation loss of around 2.52 after 120k steps. `Full-tuning-config5` (dark green) follows closely, converging slightly above 2.55. In contrast, `full-tuning-config1-restart` and `config3-restart` converge to higher loss values near 2.6, with slower learning curves. The second configuration fine tuning plot was excluded for clear presentation as it diverged early on during the training. Refer to the WandB project to see the full plot.

Figure 2 shows the evaluation loss across training steps for six FTBa fine-tuning configurations. Among them, `full-tuning-config5` (blue) achieves the best performance, reaching the lowest evaluation loss of approximately 4.3 by the 60k step mark. `full-tuning-config6` (green) follows closely, converging around 4.27. Config2 shows moderate performance, stabilizing slightly below 4.6. The results suggest that config5 and config6 are better suited for reverse token fine-tuning, likely due to favorable hyperparameters or architectural settings that enhance generalization in this task.

The future ablation work on different configurations may result in better convergence on loss for the FTBa model compared to the Pythia Vanilla. Sample generation of all the fine tuned models are included in Appendix A.6.

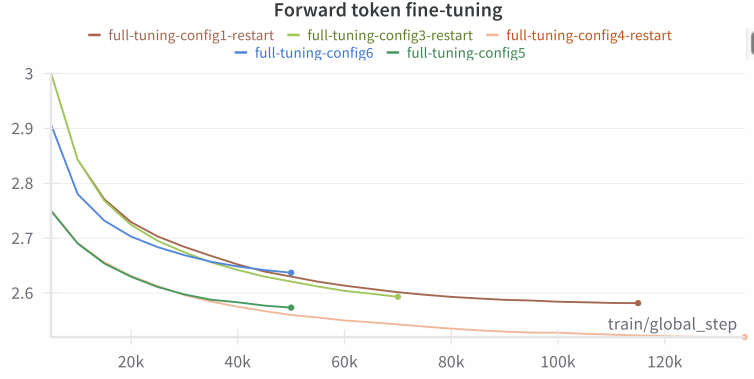


Figure 1: Evaluation loss curves for six ablation experiments on forward token fine-tuning using the Pythia-160M model.

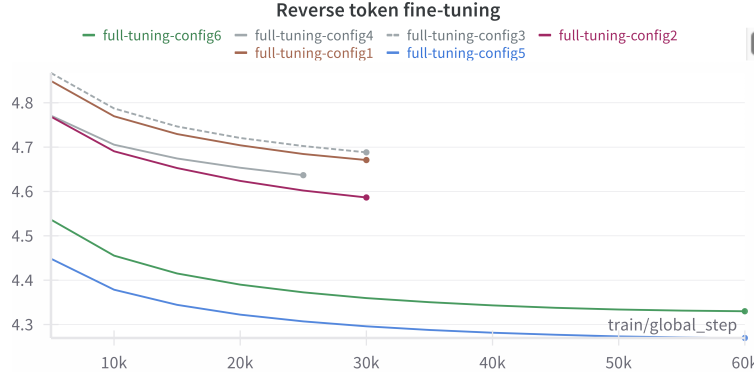


Figure 2: Evaluation loss curves for six ablation experiments on reverse token fine-tuning using the Reverse Pythia-160M model.

5.2 Attribution Results

Table 2 presents the performance comparison between our Base, VFo, VBa, FTFo, and FTBa on the citation attribution task for a sample of the CNN Daily Mail dataset. While the dataset has 287, 113 rows of data, we used $n = 95704$ (30%) for linear and binary search, while $n = 15950$ (5%) for exclusion due to the limitations we further describe in 5.5.

Table 2: Performance comparison across different search algorithms and models

Model	Direction	Linear Search			Binary Search			Exclusion Search		
		Similarity	TF-IDF	ROUGE	Similarity	TF-IDF	ROUGE	Similarity	TF-IDF	ROUGE
Base	S→A	0.479	0.199	0.221	0.441	0.165	0.183	0.456	0.164	0.184
VFo	A→S	0.652	0.296	0.336	0.641	0.292	0.332	0.615	0.290	0.332
VBa	A→S	0.649	0.303	0.342	0.641	0.299	0.339	0.647	0.307	0.354
FTFo	A→S	0.653	0.295	0.334	0.619	0.279	0.318	0.598	0.282	0.324
FTBa	A→S	0.658	0.296	0.342	0.646	0.293	0.339	0.642	0.305	0.351

5.3 Discussion

Our experiments with backward language models for citation attribution reveal several interesting findings that both support and extend the results presented by Varun et al. [2025].

5.3.1 Comparison with TRLM

As also observed in Varun et al. [2025], there is a significant improvement in citation attribution performance when using prompts in reverse direction ($A \rightarrow S$) compared to the conventional forward direction ($S \rightarrow A$). This can be seen from comparing the results across VFo and VBa with Base, for example the embedding similarity increased from 44-48% to 61-65% across the different search algorithms.

However, in contrary to TRLM, there isn't a noticeable or significant improvement of using reverse trained models. While VBa showed slight improvements over VFo in some metrics, these differences are substantially smaller than the gains achieved by simply changing inference direction. This may suggest that we might achieve significant citation improvements simply by prompting existing models in reverse without requiring specialized backward training.

Moreover, while in TRLM the search indicates clear better performance in Linear Search and worst performance in Exclusion Search, we observe no similar behavior in our experiment. The three search algorithms produced similar results.

5.3.2 Alternative Reverse Models and Scaling

Our experiments utilized relatively small models (160M parameters) compared to the larger models in the original TRLM paper. As seen from the results, we still observe significant improvements in reverse direction prompting. This may suggest that the phenomenon is consistent across different model scales and is applicable toward different language models.

However, the effect of reverse language model is not as apparent as that of TRLM's. There could be several reason to this, one being the observed emergent capabilities of language models beyond a certain size Wei et al. [2022]. Since the models we use are significantly smaller than the threshold where emergent capabilities have been observed, this may explain the different magnitude of effect compared to TRLM.

5.3.3 Effects of Instruction Fine-tuning

Our experiments showed mixed results regarding the effect of instruction fine-tuning. Across the board, FTFo results appear to have decreased compared to its pre-trained counterpart VFo, particularly in binary and exclusion search methods. Meanwhile, FTBa showed mixed results with marginal improvements in some metrics but decreases in others compared to VBa. This finding suggests that instruction fine-tuning may not meaningfully improve citation attribution performance and could potentially be detrimental in some cases.

However, it remains relevant considering that most models used in production environments are instruction-tuned variants. In a particular scenario where non-parametric citation system is using language models scoring to attribute a citation, our results suggest that further study regarding the effect of fine tuning toward citation attribution might be necessary, where it could lead to practices whether we need to use pre-trained or instruction-tuned models for the task.

5.4 Unsupervised Fine-Tuning with Reverse-Score Feedback

Motivation. Varun et al. [2025] show that a frozen backward model can already deliver large gains when used purely as an *inference-time* scorer. We wondered whether feeding the same score back into training—without any labelled citations—would push performance further.

Setup. The instruction-tuned backward model (**FTBa**) is frozen as a critic, while its forward counterpart is updated. For each highlight q we sample $N=3$ candidates $r_i \sim P_{\text{Fo}}(\cdot | q)$, compute rewards $s_i = \log P_{\text{FTBa}}(q | r_i)$, and perform a REINFORCE step

$$\mathcal{L} = -(s_{i^*} - \bar{s}) \log P_{\text{Fo}}(r_{i^*} | q), \quad i^* = \arg \max_i s_i.$$

Training for one epoch on a 50 k FLAN subset (batch 1, $\eta=5 \times 10^{-5}$) yields the checkpoint we call **U-Fo**.

Evaluation. We performed evaluation on CNN Daily Mail and the results are seen in Table 5.4.

Model	Similarity	TF-IDF	ROUGE
U-Fo (ours)	0.643	0.292	0.339

Take-away. Reverse-score fine-tuning is fully label-free; on this small pilot it matches (but does *not* exceed) the best numbers we achieved elsewhere, suggesting that most of the benefit is already captured by inference-time reverse scoring. Scaling the method to larger models or tasks with a wider forward/backward gap is left to future work.

5.5 Limitations

While we attempted to reproduce the results, several key limitations on our study exist:

1. Our current implementation of the search algorithms takes several hours to run even with the reduced CNN/Daily Mail subset. Exclusion search notably takes several times longer. It also batches nearly entire articles (minus one sentence each time), allocating enormous intermediate tensors that often trigger GPU out-of-memory errors. Because of both the excessive runtime and these OOM failures, we further reduced the dataset size.
2. Due to time constraints and compute availability, we felt it would be hard to perform fine tuning ablations on the whole FLAN dataset and so we opted to use a subset as described in earlier section.
3. However, this sampling may have introduced biases that affect the direct comparability and generalization of our findings. Moreover, prior work suggests that training on a diverse set of instructions improves downstream and generalization performance [Sanh et al., 2022, Chung et al., 2022, Bukharin et al., 2024]. Our sampling may have affected the expected model performance.
4. As mentioned in 5.3.2, the model we used are relatively small in size and may yet to exhibit the emergent capabilities seen in larger models (like TRLM and the PALM2-Otter that it is based upon, unfortunately the architecture and size are withheld from publication so we do not know the exact size Anil et al. [2023]). Therefore it may potentially be limited in to reach similar or comparable level of ability with TRLM and larger reverse-trained language models may behave differently.

6 Future Directions

Based on what we’ve found during our experiment and the limitations it has, we propose the following future directions of work:

1. To test performing fine tuning experiment and evaluation using the whole dataset (i.e. full FLAN and full CNN Daily Mail).
2. Review search attribution code to allow faster evaluation.
3. As our current result do not support that reverse trained model perform better but is also limited in size, explore using other larger reverse trained language models to see if reverse trained language models do in fact perform better in citation attribution (similar with TRLM).
4. Explore and compare the performance of this attribution mechanism for larger language models and smaller models.
5. Comparing publicly available instruction-tuned and pre-trained language models in citation attribution performance.

7 Conclusion

Our study confirms that reversing the inference direction significantly improves citation attribution performance across all search algorithms. However, unlike the TRLM paper, our experiments with Pythia show minimal advantage from reverse token training compared to conventional forward-trained models. The effect of each search algorithms are also minimal. Instruction fine tuned model in our

study and similarly our additional experiment using backward model for reinforcement learning do provide better performance compared to their pre-trained variants. Therefore, a key finding is that practitioners may achieve substantial citation attribution improvement by simply prompting existing models in reverse without requiring specialized backward training.

Despite the results, we acknowledge several limitations in our current work. We strongly encourage future research to address these constraints and build upon our notes in Section 6.

8 Administrative

8.1 Authors Contribution

The first three authors below contributed equally in producing the initial idea, project proposal and midterm report. Below are the detailed contributions up until the final report.

Ivan Wiryadi

- Set up initial citation attribution code (models, scoring, and linear search)
- Continued the work on forward model fine tuning and ran fine tuning ablations
- Performed citation attribution for fine tuned models
- Wrote initial report drafts

Jihu Hwang

- Mainly responsible on citation attribution search
- Continued the code work on initial citation attribution (adding binary and exclusion search)
- Further refined citation attribution code for performance
- Performed citation attribution for the pretrained models
- Reviewed and revised the search algorithms and attribution scoring section, and reviewed and added algorithms

Janbol Jangabyl

- Mainly responsible on fine tuning
- Wrote the initial code for fine tuning the forward and backward models
- Continued the work on backward model fine tuning and ran fine tuning ablations
- Wrote the fine tuning results section draft

Xuanyi Shen

- Ran fine tuning ablations
- Wrote code and report section 5.4 unsupervised fine tuning

8.2 Github and WandB Repository

Our implementation code can be found in this Github repository. Our ablations can be found in this WandB project

References

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun,

- Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. Palm 2 technical report, 2023. URL <https://arxiv.org/abs/2305.10403>.
- Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O’Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR, 2023.
- Alexander Bukharin, Shiyang Li, Zhengyang Wang, Jingfeng Yang, Bing Yin, Xian Li, Chao Zhang, Tuo Zhao, and Haoming Jiang. Data diversity matters for robust instruction tuning, 2024. URL <https://arxiv.org/abs/2311.14736>.
- Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models, 2022. URL <https://arxiv.org/abs/2210.11416>.
- Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *NIPS*, pages 1693–1701, 2015. URL <http://papers.nips.cc/paper/5945-teaching-machines-to-read-and-comprehend>.
- Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *CoRR*, abs/2202.03629, 2022. URL <https://arxiv.org/abs/2202.03629>.
- Weitao Li, Junkai Li, Weizhi Ma, and Yang Liu. Citation-enhanced generation for llm-based chatbots. *arXiv preprint arXiv:2402.16063*, 2024.
- Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.
- Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, and Adam Roberts. The flan collection: Designing data and methods for effective instruction tuning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 22631–22648. PMLR, 23–29 Jul 2023. URL <https://proceedings.mlr.press/v202/longpre23a.html>.
- Jacob Pfau, Alex Infanger, Abhay Sheshadri, Ayush Panda, Julian Michael, and Curtis Huebner. Eliciting language model behaviors using reverse language models. In *Socially Responsible Language Modelling Research*, 2023.
- Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL <https://arxiv.org/abs/1908.10084>.
- Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen

- Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Tali Bers, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. Multitask prompted training enables zero-shot task generalization, 2022. URL <https://arxiv.org/abs/2110.08207>.
- Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1099. URL <https://www.aclweb.org/anthology/P17-1099>.
- Yerram Varun, Rahul Madhavan, Sravanti Addepalli, Arun Suggala, Karthikeyan Shanmugam, and Prateek Jain. Time-reversal provides unsupervised feedback to llms. *Advances in Neural Information Processing Systems*, 37:29777–29806, 2025.
- Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.
- Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models, 2025. URL <https://arxiv.org/abs/2303.18223>.

A Appendix

A.1 Linear Attribution Search

Algorithm 1 Linear Attribution Search

- 1: **Input:** h - highlight sentence, $A = \{a_1, \dots, a_N\}$ - Article, model, tokenizer, backward flag, query_direction.
 - 2: Calculate scores for each sentence in A using `calculate_scores_batch( $h$ ,  $a_i$ , model, tokenizer, backward, query_direction)`.
 - 3: Return a_j corresponding to the highest score.
-

A.2 Binary Attribution Search

Algorithm 2 Binary Attribution Search

- 1: **Input:** h - highlight, $A = \{A_s, A_{s+1}, \dots, A_t\}$ - Article, model, tokenizer, backward flag, query_direction.
 - 2: Create context and target text according to query_direction and backward flags.
 - 3: $s_1 \leftarrow \text{calculate_scores_batch}(\text{left_context}, \text{left_target}, \text{model}, \text{tokenizer}, \text{backward})$
 - 4: $s_2 \leftarrow \text{calculate_scores_batch}(\text{right_context}, \text{right_target}, \text{model}, \text{tokenizer}, \text{backward})$
 - 5: **if** $s_1 > s_2$ **then**
 - 6: $t \leftarrow s + \lfloor \frac{t-s}{2} \rfloor$
 - 7: **else**
 - 8: $s \leftarrow s + \lfloor \frac{t-s}{2} \rfloor + 1$
 - 9: **end if**
 - 10: **if** $|t - s|$ has sufficient granularity **then**
 - 11: return $A_{s:t}$
 - 12: **else**
 - 13: Binary_Search_Citation(h , $A_{s:t}$, model, tokenizer, backward, query_direction)
 - 14: **end if**
-

A.3 Exclusion Attribution Search

Algorithm 3 Exclusion Attribution Search

- 1: **Input:** h - highlight sentence, $A = \{a_1, \dots, a_N\}$ - article sentences, model, tokenizer, backward flag, query_direction.
 - 2: Iterate over each sentence a_j in A
 - 3: Create $A_j = A \setminus a_j$ (article without sentence a_j)
 - 4: Create context and target according to query_direction and backward flags
 - 5: Calculate score using `calculate_scores_batch(context, target, model, tokenizer, backward)`
 - 6: Find a_j whose exclusion resulted in the lowest score
 - 7: Return a_j as the citation
-

A.4 Scoring

Algorithm 4 Batch Attribution Scoring (calculate_scores_batch)

```

1: Input: Contexts  $C = \{c_1, \dots, c_B\}$ , Targets  $T = \{t_1, \dots, t_B\}$ , Model  $M$ , Tokenizer Tok, Flag backward
2: Batch-tokenize all  $(c_i, t_i)$  pairs and move tensors to GPU
3: Compute non-pad lengths  $\text{ctx\_len}[i]$  and  $\text{tgt\_len}[i]$ 
4: Forward pass (FP16) to get logits and log-softmax probabilities
5: for  $i = 1, \dots, B$  do
6:   if  $\text{tgt\_len}[i] = 0$  then
7:      $\ell_i \leftarrow -\infty$ ,  $p_i \leftarrow +\infty$ 
8:   else
9:     Sum log-probs of the true tokens
10:    Normalize:  $\ell_i \leftarrow (\text{sum log-prob}) / \text{tgt\_len}[i]$ 
11:    Perplexity:  $p_i \leftarrow \exp(-\ell_i)$ 
12:   end if
13: end for
14: return Normalized log-probs  $\ell \in \mathbb{R}^B$ , Perplexities  $p \in \mathbb{R}^B$ 

```

A.5 Visualization for Reverse Model Input and Output

Input text: A quick brown fox

Final Output:

that I'm not familiar with?

"It depends on the shape, color and size of the animal," he said. "A quick brown fox"

Label	Content
Input Text	A quick brown fox
Tokenized	34, 3158, 8516, 30013
Flipped	30013, 8516, 3158, 34
Model Input	30013, 8516, 3158, 34
Model Input (decoded)	fox, brown, quick, A
Model Output	30013, 8516, 3158, 34, 346, 15, 753, 344, 937, 5893, 253, 273, 1979, 285, 3295, 13, 5281, 253, 327, 7024, 1147, 3, 187, 32, 342, 7615, 417, 1353, 309, 326
Model Output (decoded)	fox, brown, quick, A, ", ., said, he, ., animal, the, of, size, and, color, ., shape, the, on, depends, It, ", \n, ?, with, familiar, not, 'm, I, that
Flipped Output	326, 309, 1353, 417, 7615, 342, 32, 187, 3, 1147, 7024, 327, 253, 5281, 13, 3295, 285, 1979, 273, 253, 5893, 937, 344, 753, 15, 346, 34, 3158, 8516, 30013
Decoded Output	that, I, 'm, not, familiar, with, ?, \n, ", It, depends, on, the, shape, ., color, and, size, of, the, animal, ., he, said, ., ", A, quick, brown, fox

Table 3: Detailed steps visualization from input to final decoded output for VBa, FTBa model

A.6 Reverse Model Inference Outputs

Prompt: "A quick brown fox"

First Fine-tuned Reverse Model Inference

typically contain one or more words that have similar meaning. For example, researchers may use terms like "superfamily", "members-of-the same group" or "the same genus (species)". The tags are also useful for identifying specific individual(s). Each individual is given a common name by which they are identified, along with information such as animal nicknames, place names, and other types of information that can be used to identify the species.

Output:

A quick brown fox

Second Fine-tuned Reverse Model Inference

news for him.

'He added: 'It was not good.... We'll just wait and see what happens when they break up or something like that.'

"There's no way we're dealing with this," Coughlin said. "We've had a lot of problems." He asked himself the following question: What if the problem is still there?

"I know, but I don't want to talk about it," he said.

A quick brown fox

Third Fine-tuned Reverse Model Inference

) What happened 20 years ago. What's really happening in this country right now. I thought those things would be very bad. They're not good for us. And I think there's a lot of people who say "This will never happen again." They don't talk about it, but...

BOWMAN: Why is that?

ROBERTS: We know what they are -- and we should pay attention to them, as well.

A quick brown fox

Fourth Fine-tuned Reverse Model Inference

happens when we take a closer look at a fairly large oak tree (see picture below).

It looks like a very small plant. It stands about 6 feet tall. It has dark green leaves wrapped around its trunk. Did you notice this? The first branch must be here, but all other branches are, too! And then there's a branch that goes down and ends up on top of the tree, which is right next to it.

Output:

A quick brown fox

Fifth Fine-tuned Reverse Model Inference

them. They talk about what they would do after their father's death. He refuses to stay at a hospital because he thinks there is something which could kill him. The doctor says he may suffer for the rest of his life, but she must not give up or she will fall in love with him. He tells her that she must make sure he doesn't run away. She set out to see if she can find it, and go back to the...

Answer:

A quick brown fox

Sixth Fine-tuned Reverse Model Inference

Do you really want to go? She replies "I can come back one day." "You won't call me?" He smiles and says "Then I'll see you." He then turns around and fixes his eyes on this rather shy but very beautiful young woman. She does not think of himself as a man; she hopes that he will fall in love with her. He returns to the two women who are waiting for him, and...

The answer is:

A quick brown fox