Improving Sentence-level Attribution in RAG through Linguistic Aligned Matching

Anonymous ACL submission

Abstract

Retrieval-Augmented Generation (RAG) has been widely applied to enhance large language models (LLMs)' integration of external knowledge. Attributing the RAG-generated content, which provides citations to support responses, has attracted a lot of research interest. However, most existing studies focus on coarse-level attribution by linking claims to passages or documents, which still require certain time costs for verification. On the other hand, existing fine-grained attribution methods rely on fine-tuned LLMs to generate citations along with the content, which is expensive and hard to control. In this work, we introduce a simple yet effective *Linguistic Aligned Matching (LAM)* approach for sentence-level attribution, which follows a two-step process: refinement and matching. The refinement step aligns the expression of claims with expressions of retrieved documents using LLMs. The matching step then combines the claims and refined expressions to identify supporting sentences via vector-based matching. Unlike traditional fine-grained attribution methods, LAM is training-free and can be seamlessly integrated into existing RAG systems. Experiments across diverse domains and tasks demonstrate significant improvements, achieving an average 7.87% ROUGE-F1 gain on both short- and long-context datasets [1].

1 Introduction

Retrieval-Augmented Generation has achieved remarkable success across knowledge-intensive NLP tasks like question answering (Gao et al., 2023c) and summarization (Edge et al., 2024). However, RAG systems are still suffering from generating hallucinated content due to imperfect retrieval or overconfident generation (Tonmoy et al., 2024). To address this, recent efforts have focused on augmenting RAG answers with attribution or citations, enabling users to trace claims back to source documents (Li et al., 2023; Nakano et al., 2021; Gao et al., 2023a). Unfortunately, existing approaches predominantly rely on paragraph-level (Nakano et al., 2021; Gao et al., 2023b) or document-level (Thoppilan et al., 2022) citations, which requires users to expend considerable additional time and effort in identifying supporting sentences, undermining efficiency and trustworthiness.

Recent advances in fine-grained attribution attempt to address this through fine-tuned LLMs to generate citations along with the content (Sun et al., 2023; Zuccon et al., 2023; Jain et al., 2023). Approaches such as LongCite (Zhang et al., 2024) and ReClaim (Xia et al., 2024) require extensive synthetic annotation data for fine-tuning, incurring substantial annotation and computational costs. Moreover, these approaches remain susceptible to producing hallucinated citations during generation, which limits their applicability in safety-critical scenarios demanding rigorous verification. In contrast to generation-based approaches, matching-based approaches attributing supporting sentences to claims through Natural Language Inference (NLI) models or vector-based models (Gao et al., 2023b; Huo et al., 2023; Chen et al., 2024). These methods ensure verifiability and traceability by directly extracting verbatim supporting sentences from source texts. However, matching-based approaches often exhibit limited capacity in capturing global document-level coherence and contextual dependencies, resulting in suboptimal performance in complex scenarios that demand coreference resolution, ellipsis interpretation, or implicit reasoning capabilities.

To address the limitations of matching approaches, we propose a post-hoc Linguistic Aligned Matching (LAM) approache that synergizes the global linguistic comprehension capabili-

[1] Our code and data can be found at https://anonymous.4open.science/r/LAM-Linguistic-Aligned-Matching-1C4C/

ties of LLMs with rigorous textual correspondence verification. The LAM follows a two-step procedure: (1) *The refinement step*: Leveraging LLMs to establish semantic alignment between original claims and document expressions through context-aware rephrasing, thereby encoding document-level contextual information; (2) *The matching step*: Employing vector-based models to attribute supporting sentences to contextually-aligned claims while preserving textual consistency. Our training-free approach uniquely synthesizes the contextual comprehension strengths of generation-based approaches with the textual fidelity inherent in matching-based approaches.

To evaluate the effectiveness of our method, we adapted multi-domain open-source datasets (including FEVER (Thorne et al., 2018), WebGLM-QA (Liu et al., 2023) and LongBench (Bai et al., 2023)) to fit the task of sentence-level attribution. Experiments on these datasets demonstrate that LAM achieves an average 7.87% improvement in Rouge-F1 over baseline methods, show the effectiveness and generalization on various tasks and document lengths.

2 Method

This paper studies the sentence-level attribution problem in RAG, aiming to provide supporting sentences for all claims in the answers. Formally, given a set of K claims $\{c_0, c_1, ..., c_K\}$ in the answers generated by RAG and a corpus of retrieved documents $\mathcal{D}$, the sentence-level attribution is to identify the supporting sentences $S = \{s_{i0}, s_{i1}, ...\}$ within $\mathcal{D}$ that substantiate the claim c_i.

In this paper, we integrate the strengths of both the generation method and the matching method to propose LAM, a training-free, reliable method that is capable of effectively handling complex scenarios. Specifically, LAM consists of two steps: *the refinement step* and *the matching step*, as shown in Figure 2. **The refinement step** utilizes generative models to refine the claims, aligning expressions and key information between the claims and the document, thus mitigating information incompleteness and expression discrepancies caused by coreference resolution, cross-sentence inference, and so on. Then,**the matching step** combines the claims and refined expressions, using vector-based models to attribute supporting sentences to the claim, ensuring consistency with the original document.

2.1 The Refinement Step

To achieve the refinement, we use LLM as the foundational model and design a zero-shot prompt to guide the model in generating sentences related to claims within the document. We have carefully designed a structured prompt to achieve constrained generation, which ensures textual consistency with the document. The staged instruction format *"Memorize...select..."* induces structured reasoning simulating human cognitive processing. In addition, *"return original sentences"* instruction implements strict textual consistency constraints, preventing paraphrasing or generative hallucination. The detailed design of the prompt is shown in Appendix C. The formal description is as follows:

$$\hat{c} = f_{\text{LM}}(c, \mathcal{D}) \tag{1}$$

where $\hat{c}$ denotes aligned claim and $f_{LM}(\cdot)$ denotes the model.

2.2 The Matching Step

After refinement, we use the claim and the refined expressions as input to achieve precise supporting sentence identification.

Specifically, to perform sentence-level attribution, we first use NLTK's sentence tokenizer to segment the given relevant document into atomic sentences, forming a candidate sentence set $S = \{s_1, s_2, ..., s_n\}$. Then we encode the claims c, refined expressions $\hat{c}$ and candidate sentences s_i with vector model, formalized as:

$$\mathbf{v}_c, \mathbf{v}_{\hat{c}}, \mathbf{v}_{s_i} = Encode(c, \hat{c}, s_i) \tag{2}$$

Where $\mathbf{v}_c, \mathbf{v}_{\hat{c}}, \mathbf{v}_{s_i}$ donate encoded texts.

As the fusion manner of the claim and refined expressions may affect the matching performance, we designed two strategies, *i.e.* concatenation and feature pooling, to provide greater flexibility in adapting to various models. The details are outlined below:

- **Concatenation**: Directly concatenate the two claims and then encode the concated claim $\mathbf{c}_{\text{con}}$ to the fused vector $\mathbf{v}_{\text{claim}}$:

$$\mathbf{c}_{\text{con}} = concat\,(\mathbf{c}, \hat{\mathbf{c}}) \tag{3}$$

$$\mathbf{v}_{\text{claim}} = Encode\,(\mathbf{c}_{\text{con}}) \in \mathbf{R}^d \tag{4}$$

- **Feature Pooling**: Compute element-wise mean pooling of the two vectors:

$$\mathbf{v}_{\text{claim}} = \frac{\mathbf{v}_c + \mathbf{v}_{\hat{c}}}{2} \in \mathbf{R}^d \tag{5}$$

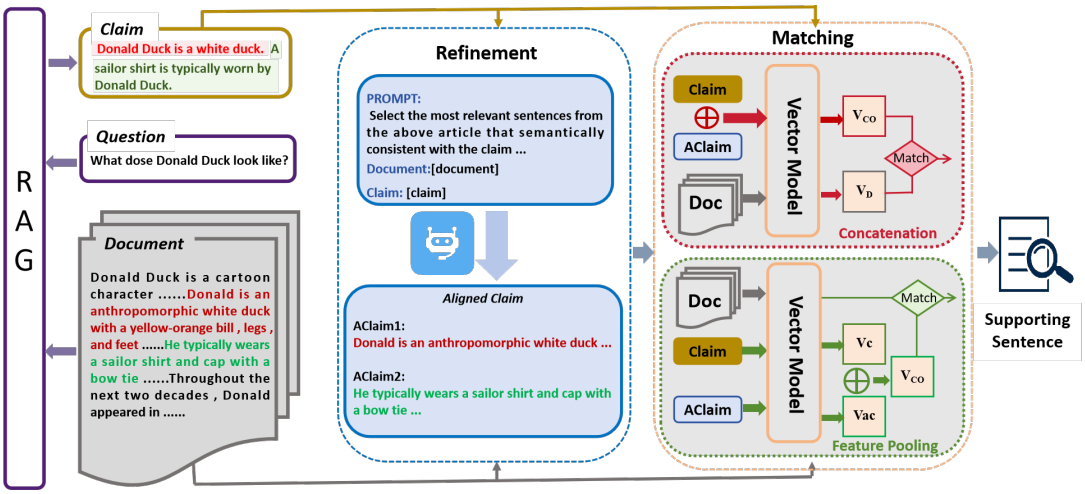

Figure 1: **Method overview of LAM**

Finally, we compute the cosine similarity between the embeddings of the fused claim v_{claim} and candidate sentences v_{s_i} to identify the supporting sentences. Then the sentence with the highest score is selected as the supporting sentence s_c for the current claim c.

$$\text{score}(\mathbf{v}_{\text{claim}}, \mathbf{v}_{s_i}) = \frac{\mathbf{v}_{\text{claim}} \cdot \mathbf{v}_{s_i}}{\|\mathbf{v}_{\text{claim}}\| \|\mathbf{v}_{s_i}\|} \quad (6)$$

$$s_c = \arg\max_{s_i \in S} \text{score}(s_i) \quad (7)$$

3 Experiments

In this section, we evaluate the performance of our method on several natural language processing (NLP) tasks across five datasets by comparing it with multiple baselines.

3.1 Experiments Settings

Datasets. We construct a multi-dimensional benchmark comprising five datasets with various task types and document lengths. Specifically, we choose short-text fact verification dataset FEVER, open-domain QA dataset WebGLM-QA, and three long-text datasets from LongBench, including MultiFieldQA, HotpotQA and GovReport. All datasets are converted into a unified *claim-document-supporting sentence* triplet format through specific transformation pipelines. The statistics of our evaluation datasets are presented in Appendix A.

Evaluation Metric. We use ROUGE-L as the metric to evaluate our LAM, comparing the consistency between the model output and the ground truth. Recognizing that low-precision citations risk introducing hallucinations, we focus on high-confidence outputs by setting a strict precision threshold:

$$\text{Valid } s_c = \{s_c | P_{rouge}(s_c, s_{\text{gt}}) \geq 0.9\} \quad (8)$$

where s_{gt} denote the ground truth supporting sentence. Specifically, for matching-based models, we focus on the top-1 retrieved sentence.

Methods and Baselines. We compare LAM with five representative baselines, including generation-based methods and matching-based methods. For generation-based methods we choose **GPT-4o-0806** (Achiam et al., 2023), as it is the best performing generative model. For matching-based methods, we choose two tower models including **DPR** (Karpukhin et al., 2020), bert-based model **Simcse** (Gao et al., 2021), and LLM-based vector model **NV-Embed-V2** [2] (Lee et al., 2024). In addition, we also include the recently introduced constrained generation method, named **CFIC** (Qian et al., 2024), as the baseline. We choose Mistral-7B (Jiang et al., 2023) as the base model of CFIC.

3.2 Experiment Results

Overall Performance. As shown in Table 1, the experimental results highlight three principal findings:

First, despite GPT-4o's impressive language generation capabilities, its next-token decoding paradigm introduces limitations in preserving strict textual consistency, which is evidenced by average F1 score of only 41.98%. Due to the limitations of

[2]NV-Embed-V2 is the best open-source embedding model on MTEB benchmark.

Method	WebGLM-QA			FEVER			MultiFieldQA			HotpotQA			GovReport		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
CFIC	24.08	17.90	19.72	49.46	46.53	47.42	21.24	16.14	17.70	10.95	7.28	8.26	18.39	10.56	12.52
GPT-4o	39.72	35.72	37.04	59.43	54.01	55.72	50.33	42.17	44.77	52.81	38.91	43.41	33.45	26.82	28.97
DPR	50.83	36.81	40.93	56.36	53.14	54.14	46.31	32.85	36.59	43.39	30.66	33.98	39.96	23.43	27.77
Simcse	63.14	46.95	51.98	66.82	63.01	64.19	69.74	53.82	58.10	69.42	52.40	57.00	61.60	39.24	45.24
NV-Embed-V2	66.25	50.33	54.98	62.45	61.39	61.28	79.88	61.07	65.86	73.84	56.39	61.09	70.38	47.71	53.71
LAM(Ours)	**70.96**	**53.95**	**59.04**	**83.14**	**82.00**	**81.76**	**81.90**	**64.08**	**68.74**	**78.32**	**59.85**	**64.82**	**80.24**	**54.90**	**61.92**

Table 1: Main results of our experiment. LAM here uses GPT-4o in contextual refinement Step and NV-Embed-V2 in precision matching step, as well as feature pooling method in claim vector fusion.

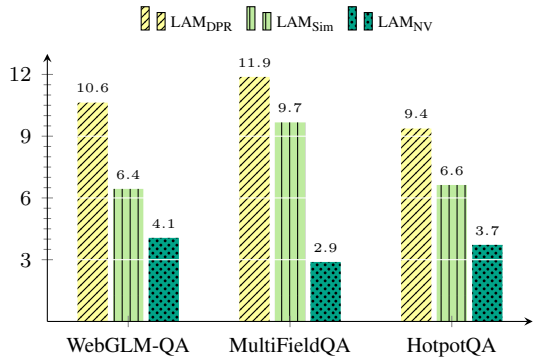

Figure 2: F1 increase of LAM results with different vectors. Results on FEVER and GovReport can be found in Appendix 3.

fusion method	WQA	GR	HQA
No fusion	54.98	53.71	61.09
Concatenation	57.57	61.78	63.04
Feature pooling	59.04	61.92	64.82

Table 2: Results of F1 score for different vector fusion methods on WebGLM-QA, GovReport and HotpotQA datasets. The vector model here is NV-Embed-V2. Other results in Appendix B.

the basic model, the performance of CFIC performs less well than GPT-4o.

Second, conventional vector-based models achieve higher accuracy (average F1=59.38%) through exact pattern matching but encounter inherent limitations. The local semantic matches fail to capture document-level coherence, resulting in mismatch errors according to our diagnostic analysis.

Lastly, our LAM method establishes new state-of-the-art performance by synergistically combining complementary strengths: the generative model's contextual awareness with the vector model's textual consistency. On short-text dataset FEVER, LAM surpasses GPT-4o by 27.47% in F1-score (83.19% vs. 55.72%) and exceeds standalone vector models by 19%. These advantages persist in long-context tasks, where LAM achieves consistent improvements of 3.73%–8.21% across various long-text datasets, demonstrating exceptional scalability.

Ablation Study. To evaluate the robustness of our LAM approach across various vector models, we performed a comparison of embeddings from DPR, Simcse and NV-Embed-V2. As shown in Figure 2, our LAM approach has demonstrated improvements across all of three vector models with different architectures on various dataset. Especially, models with relatively weaker capabilities, such as DPR, demonstrate particularly pronounced (more than 10 F1 score on average) improvement, which demonstrates that our method have strong robustness and excellent generalization.

To evaluate the influence of different vector fusion methods, we compared the performance between No fusion, Concatenation and Feature pooling. As shown in Table 2, feature pooling shows better performance in most cases. The reason is that concatenation may exceed the input length limit of model in some cases, which significantly affect the performance of the concatenation method.

4 Conclusion

This work proposes LAM, a simple but effective two-step approach designed for fine-grained, sentence-level attribution. Through contextual refinement and precision matching, our approach achieve new SOTA sentence-level attribution. Experiments on various task datasets show the effectiveness and scalability of our approach. In the future, we intend to further explore other directions for enhancing matching methods, such as reason-enhanced matching, in order to achieve more precise and comprehensive fine-grained attribution.

Limitations

While our framework demonstrates promising results, two principal constraints merit consideration. First, constrained by practical experimentation scales, our comparative analysis with generative baselines is currently limited to GPT-4o model. An empirical investigation encompassing open-source generative models (e.g., LLaMA-3, Mistral) would provide more comprehensive insights into cross-model generalizability.

Second, the inherent document segmentation process in matching-based paradigms introduces limitations when handling composite evidence requiring multi-sentence reasoning, which demands further experiments and optimization.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*.

Jifan Chen, Grace Kim, Aniruddh Sriram, Greg Durrett, and Eunsol Choi. 2024. Complex claim verification with evidence retrieved in the wild. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3569–3587.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, et al. 2023a. Rarr: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16477–16508.

T Gao, X Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *EMNLP 2021-2021 Conference on Empirical Methods in Natural Language Processing, Proceedings*.

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.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023c. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Siqing Huo, Negar Arabzadeh, and Charles Clarke. 2023. Retrieving supporting evidence for generative question answering. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, pages 11–20.

Palak Jain, Livio Soares, and Tom Kwiatkowski. 2023. 1-pager: One pass answer generation and evidence retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14529–14543.

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

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. Nv-embed: Improved techniques for training llms as generalist embedding models. *arXiv preprint arXiv:2405.17428*.

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

Xiao Liu, Hanyu Lai, Hao Yu, Yifan Xu, Aohan Zeng, Zhengxiao Du, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. Webglm: Towards an efficient web-enhanced question answering system with human preferences. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4549–4560.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

Hongjin Qian, Zheng Liu, Kelong Mao, Yujia Zhou, and Zhicheng Dou. 2024. Grounding language model with chunking-free in-context retrieval. *arXiv preprint arXiv:2402.09760*.

Zhiqing Sun, Xuezhi Wang, Yi Tay, Yiming Yang, and Denny Zhou. 2023. Recitation-augmented language models. In *The Eleventh International Conference on Learning Representations*.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819.

SM Tonmoy, SM Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. 2024. A comprehensive survey of hallucination mitigation techniques in large language models. *arXiv preprint arXiv:2401.01313*.

Sirui Xia, Xintao Wang, Jiaqing Liang, Yifei Zhang, Weikang Zhou, Jiaji Deng, Fei Yu, and Yanghua Xiao. 2024. Ground every sentence: Improving retrieval-augmented llms with interleaved reference-claim generation. *arXiv preprint arXiv:2407.01796*.

Jiajie Zhang, Yushi Bai, Xin Lv, Wanjun Gu, Danqing Liu, Minhao Zou, Shulin Cao, Lei Hou, Yuxiao Dong, Ling Feng, et al. 2024. Longcite: Enabling llms to generate fine-grained citations in long-context qa. *arXiv preprint arXiv:2409.02897*.

Guido Zuccon, Bevan Koopman, and Razia Shaik. 2023. 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*, pages 46–51.

A Dataset Statistics

The statistics of our evaluation datasets are presented in Table 3. WQA, MQA, HQA and GR denote WebGLM-QA, Multifield-QA, HotpotQA and GovReport. AvL donates average document length, as well as t donates token length.

Dataset	AvL(t)	Numc	Task Type
FEVER	1881(483)	3595	Fact Verification
WQA	8656(2022)	460	Open-domain QA
MQA	31459(8086)	287	Long-form QA
HQA	56191(14492)	274	Multi-hop QA
GR	54548(11868)	663	Summarization

Table 3: Dataset Statistics and Transformation Details

For FEVER, We select 3,595 SUPPORTS-labeled instances. For WebGLM-QA, we curating 200 high-quality QA pairs, search relevant documents through Google API, and manual annotated supporting sentences. For the other three datasets, we follow LongCite's methodology, employ GPT-4o to annotate citation spans, retaining only claims with perfect citation precision (no redundancy).

B Ablation study of Embedding and Fusion

The results of LAM with different embeddings and fusion methods on FEVER and GovReport datasets are shown in Figure 3 and Table **??**.

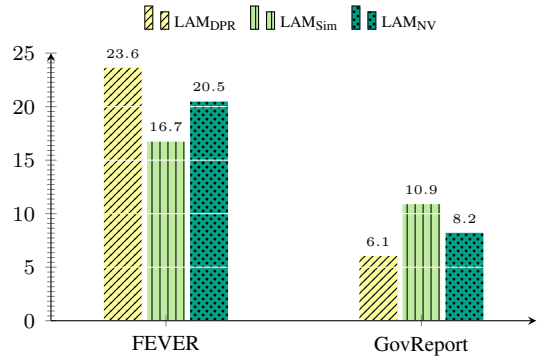

Figure 3: F1 increase of LAM results with different vector models on FEVER and GovReport datasets.

vector fusion method	FEVER	MQA
No fusion	61.28	65.86
Concatenation	79.97	69.79
Feature pooling	81.76	68.74

Table 4: Results of F1 score for different vector fusion methods on FEVER and Multifield-QA datasets.

C Prompt Template

The detailed design of our prompt in contextual refinement step is shown in Table 5

Input: Original claim c + relevant document D
Prompt Template:Below is an article. Memorize the article and select several sentences supporting my claim after the article.
The article begins:[Document]
Now the article ends.
Select the most relevant sentences from the above article that semantically consistent with the claim. Return the original sentences without any additional information.
Claim: [claim]
Output: [sentences]

Table 5: The prompt of Contextual Refinement.