Optimizing the Arrangement of Citations in Related Work Section

Masashi Oshika Ryohei Sasano
Graduate School of Informatics, Nagoya University
oshika.masashi.f6@s.mail.nagoya-u.ac.jp
sasano@i.nagoya-u.ac.jp

Abstract

In related work section of a scientific paper, authors collect relevant citations and structure them into coherent paragraphs that follow a logical order. Previous studies have addressed citation recommendation and related work section generation in settings where both the citations and their order are provided in advance. However, they have not adequately addressed the optimal ordering of these citations, which is a critical step for achieving fully automated related work section generation. In this study, we propose a new task, citation arrangement, which focuses on determining the optimal order of cited papers to enable fully automated related work section generation. Our approach decomposes citation arrangement into three tasks: citation clustering, paragraph ordering, and citation ordering within a paragraph. For each task, we propose a method that uses a large language model (LLM) in combination with a graph-based technique to comprehensively consider the context of each paper and the relationships among all cited papers. The experimental results show that our method is more effective than methods that generate outputs for each task using only an LLM.[1]

1 Introduction

Related work section of scientific papers plays a crucial role in demonstrating the positioning and novelty of the research. However, simply citing relevant papers is not sufficient for readers to fully understand the research; it is also necessary to consider the relationships among these papers and to arrange them in well-structured, logically ordered paragraphs. Existing work aimed at the automatic related work section generation includes research on citation recommendation and section generation. However, citation recommendation research has generally focused on selecting which papers to cite from a set of candidates, not on determining their order within the text (McNee et al., 2002; Ostendorff et al., 2022; Ghosh Roy and Han, 2024). In addition, many studies on related work section generation use a predefined citation order (Hoang and Kan, 2010; Chen et al., 2022; Li and Ouyang, 2025). Consequently, previous research has not addressed how to arrange or structure the cited papers, and thus has not yet achieved fully automated related work section generation (Li and Ouyang, 2024).

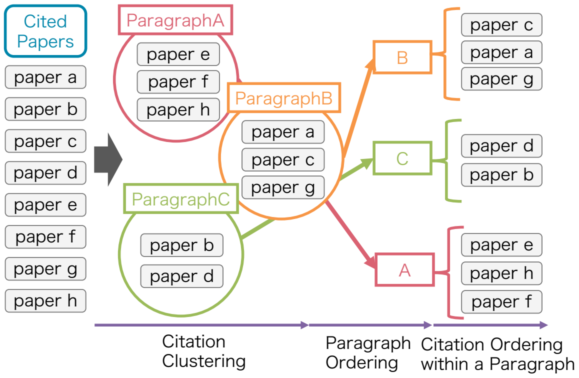

Figure 1: Overview of the citation arrangement.

Although recent LLMs can generate high-quality text, their attempts to produce an entire related work section in a single pass still lag behind human-written sections.[2] This suggests that, in order to achieve a level of quality comparable to human-written sections, it is necessary to decompose the related work section into smaller units and generate each one separately. By doing so, it becomes easier to construct a coherent structure and incorporate relevant citations more effectively.

To address these issues, we propose a new task, citation arrangement, which focuses on optimally arranging cited papers in the related work section. Figure 1 shows an overview of this task. Given a set of papers to be cited, this task involves cluster-

[1] We have released our code and dataset at https://github.com/masashi-o8/Citation-Arrangement

[2] Experimental results are shown in Table 7.

Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 1362–1373
December 20-24, 2025 ©2025 Association for Computational Linguistics

ing them into appropriate paragraphs, determining a coherent order of paragraphs, and ordering the citations within each paragraph. By accomplishing this, we address a previously underexplored aspect of related work generation and move one step closer to complete automation. We further propose a three-step approach to achieve this goal, as depicted in Figure 1. First, we cluster the cited papers into paragraphs; second, we determine the order in which these paragraphs should appear; and third, we decide the citation order within each paragraph.

When determining the citation order within the related work section, it is important not only to consider the content of each paper but also to account for the interrelations among them and the relationships between paragraphs. To achieve these requirements, we propose a method that integrates an LLM with a graph-based approach. By constructing a weighted graph based on LLM outputs, our method assigns context-aware scores and comprehensively captures the relationships among all citations, thereby enabling a more coherent arrangement of the cited papers.

2 Dataset

To train and evaluate our model, we used a dataset that includes lists of papers cited in the related work sections, along with the corresponding paragraph structures and the citation orders within those sections. Our dataset was constructed from two resources: ACL OCL (Rohatgi et al., 2023), which is a corpus of 80,013 papers published in the ACL Anthology up to August 2022; and S2ORC (Lo et al., 2020), which provides metadata for papers on Semantic Scholar.

The steps for creating the dataset are as follows:

1. Extract papers published in ACL, EMNLP, NAACL, or TACL between 2013 and 2022 from ACL OCL (12,564 papers).
2. Convert each paper to text using PDFNLT,[3] and then select those that contain a "Related Work" section (3,741 papers).
3. Filter the subset to include only papers whose "Related Work" section comprising multiple units,[4] where each unit cites at least two citations and the section contains at least 10 citations (2,869 papers).
4. Use the titles of the cited papers to retrieve author information and citation relationships from S2ORC.

# of source papers	2,869
# of paragraphs	8,709 (3.0)
# of cited papers	56,765 (19.8)

Table 1: Statistics of the dataset. The values in ( ) represent averages per paper.

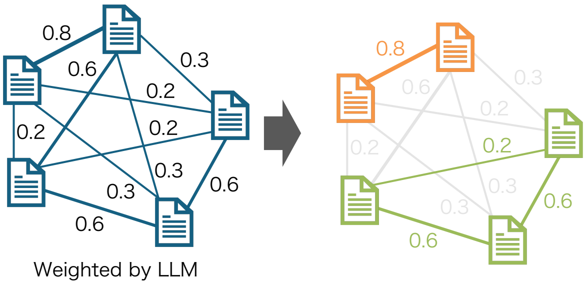

Figure 2: Overview of citation clustering.

Table 1 shows the statistics of our dataset.

3 Citation Clustering

We address citation clustering, which groups the set of cited papers into paragraphs. As illustrated in Figure 2, we construct a graph in which each node represents a cited paper. This graph structure allows us to capture the relationships among papers during clustering.

3.1 Graph Construction

To incorporate information from all cited papers, we construct a complete graph $G = (V, E)$, where each node corresponds to a cited paper and every pair of nodes is connected by an edge. The weight of each edge indicates how likely it is that the two papers would appear in the same paragraph, and we compute this weight using an LLM.

Specifically, we fine-tune an LLM on a binary classification task. Given a pair of cited papers, the model outputs 1 if they are cited in the same paragraph and 0 otherwise. After fine-tuning, we score every pair of cited papers with the model. We then apply a softmax function to the two-class logits and take the output score as the edge weight.

We compare two configurations of model inputs. The base model $\mathcal{M}_{\text{base}}$ uses only the title and abstract of each cited paper. The extended model

[3] https://github.com/KMCS-NII/PDFNLT-1.0

[4] Here, a "unit" denotes the highest-level division within the related work section. Accordingly, a single unit may correspond to a paragraph or, in some cases, to a subsection. However, for simplicity, we consistently refer to each unit as a "paragraph" throughout this paper.

$\mathcal{M}_{\text{extd}}$ additionally incorporates the title and abstract of the source paper, the publication year, author information of each cited paper, and a flag indicating whether a citation relationship exists between the two papers. Additionally, we construct four variants of $\mathcal{M}_{\text{extd}}$ as an ablation study, each omitting one of its additional input features. Specifically, $\mathcal{M}_{\backslash\text{year}}$ omits the publication year, $\mathcal{M}_{\backslash\text{auth}}$ excludes the author information, $\mathcal{M}_{\backslash\text{cite}}$ leaves out the citation relationship, and $\mathcal{M}_{\backslash\text{seed}}$ does not include the source paper information.[5]

3.2 Clustering Method

For graph clustering, we employ the normalized cut (Ncut) method (Shi and Malik, 2000), which partitions the graph so that the total weight of intra-cluster edges is maximized while that of inter-cluster edges is minimized. Let w_{ij} denote the weight between nodes i and j. Given two clusters A and $\bar{A}$ resulting from the partition, we define the cut between them as:

$$\text{Cut}(A, \bar{A}) = \sum_{i \in A, j \in \bar{A}} w_{ij}, \tag{1}$$

$$\text{Vol(A)} = \sum_{i \in A, j \in V} w_{ij}. \tag{2}$$

The Ncut is given by:

$$\text{Ncut}(A, \bar{A}) = \frac{\text{Cut}(A, \bar{A})}{\text{Vol}(A)} + \frac{\text{Cut}(A, \bar{A})}{\text{Vol}(\bar{A})}, \tag{3}$$

and the Ncut method seeks to minimize this value. By applying Ncut to the graph G, we obtain clusters that correspond to the paragraphs of the related work section.

3.3 Experiments

Settings We used the dataset described in Section 2, divided into three sets: training, development, and test, at a ratio of 8:1:1. Table 2 shows the statistics of the three sets in the constructed dataset. Since Ncut requires the number of clusters to be specified in advance, we set it to the actual number of paragraphs in the source paper. As evaluation metrics, we adopted the B-Cubed F1-score (Bagga and Baldwin, 1998) and CEAF$_\text{m}$ F-score (Luo, 2005). For the LLM, we used the LLaMA3.1-8B-Instruct,[6] Mistral-7B-Instruct-v0.2,[7] and Qwen2.5-7B-Instruct.[8] We used LoRA (Hu et al., 2022) for fine-tuning, applying it to all linear layers in the model, with hyperparameters set to r = 8 and α = 16. The learning rate for fine-tuning was decayed linearly from an initial rate of 1e-4. We set the batch size to 8 and used AdamW as the optimization method. Fine-tuning was performed for 1 epoch.[9] We used five different prompts with both the $\mathcal{M}_{\text{base}}$ and $\mathcal{M}_{\text{extd}}$, then chose the one that achieved the highest binary-classification F-score on the development set. We applied the best prompt found for $\mathcal{M}_{\text{base}}$ to all other models. Prompt tuning was performed only on LLaMA; Mistral and Qwen used the same prompt optimized for LLaMA.[10]

	Training	Development	Test
# of source papers	2,296	287	286
# of paragraphs	9,205	1,166	1,207
# of cited papers	45,447	5,691	5,627
# of pairs of paragraphs	16,388	2,072	2,426
# of pairs of cited papers	493,367	62,211	60,198

Table 2: Statistics of the train/dev/test splits.

Compared Methods We constructed two baseline methods to verify the effectiveness of our approach. The first baseline is a K-means clustering approach applied to the embeddings of the cited papers. For the K-means method, we obtained embeddings of the cited papers using SciNCL (Ostendorff et al., 2022), which was pre-trained to handle scientific papers. Specifically, we concatenated the title and abstract of each cited paper using <sep> and then encoded the text with SciNCL. The second method is an LLM-based method (LLM$_\text{base}$). The LLM$_\text{base}$ takes the source paper and all cited papers as input, and is fine-tuned to generate the cluster assignment for each cited paper. This baseline also employs LLaMA3.1-8B-Instruct, Mistral-7B-Instruct-v0.2, and Qwen2.5-7B-Instruct and is fine-tuned via LoRA. We set the batch size to 1, fine-tuned for 3 epochs, selecting the model with the smallest loss on the development set for evaluation. All other parameters were the same as those used in the proposed method, and the input features are identical to those in $\mathcal{M}_{\backslash\text{cite}}$. We fine-tuned models using five different prompts and selected

[5] Appendix A shows a summary of the inputs.

[6] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

[7] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2

[8] https://huggingface.co/Qwen/Qwen2.5-7B-Instruct

[9] More details are shown in Appendix B.

[10] The actual prompts are shown in Appendix C.

You are a researcher specializing Natural Language Processing. I am writing scientific paper with the following title and abstract.
Title:{Title of Source Paper}
Abstract:{Abstract of Source Paper}
Please cluster the following {# of references} references for {# of paragraphs} clusters. You can determine cluster membership according to the topic of the references.
references
Title:{Tit. of paper1}, Abstract:{Abs. of paper1}, Author:{Auth. of paper1}, Year of Publication:{Year of Pub. of paper1}
Title:{Tit. of paper2}, Abstract:{Abs. of paper2}, Author:{Auth. of paper2}, Year of Publication:{Year of Pub. of paper2}
Title:{Tit. of paper3}, Abstract:{Abs. of paper3}, . . .

Table 3: Prompt used in LLM$_{\text{base}}$ of citation clustering. All cited papers are input after **references**.

	B^3	CEAF$_\text{m}$				
K-means	.637	.625				

	LLaMA		Mistral		Qwen	
	B^3	CEAF$_\text{m}$	B^3	CEAF$_\text{m}$	B^3	CEAF$_\text{m}$
LLM$_\text{base}$	.697	.709	.689	.703	.688	.707
$\mathcal{M}_\text{base}$	.715	.727	.696	.706	.702	.717
$\mathcal{M}_\text{extd}$	.717	.730	.711	.721	.713	.725
$\mathcal{M}_{\backslash\text{year}}$	.715	.726	.713	.727	**.719**	**.734**
$\mathcal{M}_{\backslash\text{auth}}$	.717	.731	.717	.729	.714	.720
$\mathcal{M}_{\backslash\text{cite}}$	.708	.718	.715	.729	.602	.624
$\mathcal{M}_{\backslash\text{seed}}$	.716	.726	.704	.715	.717	.728

Table 4: Experimental results of citation clustering.

the one with the lowest loss on the development set for evaluation. Prompt tuning was also performed exclusively on LLaMA; Mistral and Qwen used the same prompt optimized for LLaMA. The actual prompt is shown in Table 3.

Results Table 4 shows the results of citation clustering. K-means showed a lower score than methods using LLM, demonstrating that the LLM can adequately account for the context of cited papers. Compared to the LLM$_\text{base}$ and proposed methods, almost all proposed methods achieve a higher F-score than the baseline, confirming the effectiveness of the graph-based method that takes the relationships among cited papers into account.

Comparing $\mathcal{M}_\text{base}$ and $\mathcal{M}_\text{extd}$, we observed score improvements across all models. This indicates that clustering performance is improved by increasing the amount of input features. Moreover, the results of our ablation study indicate that $\mathcal{M}_\text{base}$ tends to perform worse than other variants, further confirming the benefit of augmenting paper features. However, the features that prove effective for clustering differ across LLMs, so it is essential to employ the most suitable ones.

4 Paragraph Ordering

Next, we address paragraph ordering, which determines the order in which the clustered paragraphs appear in the related work section. As shown in Figure 3, we construct a directed graph in which each node represents a paragraph.

4.1 Graph Construction

We build a directed graph by treating each paragraph as a node and connecting every pair of nodes with edges in both directions. We then compute and assign a weight to each edge using an LLM, which indicates how likely one paragraph is to immediately precede another.

Concretely, we fine-tune an LLM on a binary classification task where the input consists of two paragraphs, A and B. Each paragraph is formed by concatenating the cited papers within it. The model outputs 1 if paragraph A directly precedes paragraph B in the source paper and 0 otherwise. After fine-tuning, we score a paragraph pair (A, B) with the model. We then apply a softmax function to the two-class logits, and use the resulting score as the directed edge weight w_{AB}.

We compare five configurations: $\mathcal{M}_\text{base}$, $\mathcal{M}_\text{extd}$, $\mathcal{M}_{\backslash\text{year}}$, $\mathcal{M}_{\backslash\text{auth}}$, $\mathcal{M}_{\backslash\text{seed}}$. In paragraph ordering, since each paragraph is composed of the entire concatenated text of cited papers, citation relationships are not separately considered and thus are not utilized.

4.2 Ordering Method

We determine the paragraph order by exhaustively searching all possible permutations to find the permutation π that maximizes the objective function defined in Equation (4) :

$$\arg\max_{\pi} \sum_{1\leq i<j\leq n} w_{\pi(i)\pi(j)}, \tag{4}$$

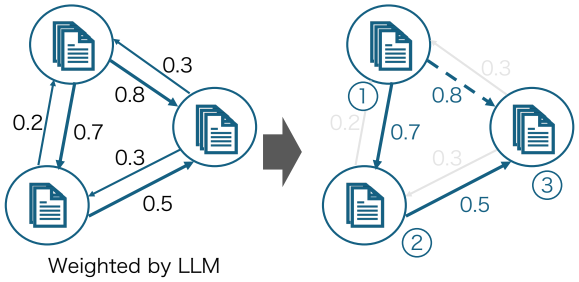

Figure 3: Overview of the paragraph ordering. In the right graph, the solid-line relationships are learned during fine-tuning, while the dashed-line weights are taken into account when determining the order.

where $\pi(i)$ is the paragraph at position i in the ordering, and $w_{\pi(i)\pi(j)}$ is the weight of the directed edge from paragraph $\pi(i)$ to paragraph $\pi(j)$. Here, the edge weight indicates how likely two paragraphs are to be adjacent, and by maximizing the sum of these weights as in Equation (4), our method effectively captures the desired adjacency relationships. Please note that when learning the weights, only adjacencies are considered, whereas in Equation (4) all backward nodes are taken into account. We also experimented with a setting that only considered adjacencies, but this resulted in an average score about 0.024 lower.

4.3 Experiments

Settings We employed LLaMA3.1-8B-Instruct, Mistral-7B-Instruct-v0.2, and Qwen2.5-7B-Instruct and adopted LoRA for fine-tuning. We set the batch size to 2 and fine-tuned the model for 3 epochs. The model with the smallest loss on the development set was used for evaluation. All other parameters and settings were the same as those used in citation clustering. For the evaluation metric, we used Spearman's rank correlation coefficient ρ and Kendall rank correlation coefficient τ. The prompt tuning followed the same setting as in citation clustering.

Compared Methods We constructed three baseline methods for comparison. The first method is *Mean-Publication-Year*. In this method, we computed the average publication year of the cited papers in each paragraph, then obtained the paragraph order by sorting these averages in ascending order. The second method is *Paragraph-Size*. In this method, we determined the paragraph order by sorting the paragraphs in descending order based on the number of cited papers in each paragraph. The LLM$_{\text{base}}$ took the source paper and all paragraphs as input and is fine-tuned to generate a paragraph order. The fine-tuning settings were identical to those used for the LLM$_{\text{base}}$ in citation clustering, and the input features were the same as those used for $\mathcal{M}_{\text{extd}}$ in paragraph ordering. The prompt tuning was done in the same setting as for citation clustering.

	ρ	τ				
Mean-Publication-Year	.253	.229				
Paragraph-Size	.304	.287				
	LLaMA		Mistral		Qwen	
	ρ	τ	ρ	τ	ρ	τ
LLM$_{\text{base}}$	**.573**	.537	.522	.490	.467	.440
$\mathcal{M}_{\text{base}}$	.470	.444	.494	.460	.486	.460
$\mathcal{M}_{\text{extd}}$	.554	.524	.572	.541	.567	.532
$\mathcal{M}_{\setminus\text{year}}$	.559	.521	.541	.511	.533	.501
$\mathcal{M}_{\setminus\text{auth}}$	**.573**	**.543**	.542	.504	.533	.498
$\mathcal{M}_{\setminus\text{seed}}$	.507	.475	.513	.480	.507	.477

Table 5: Experimental results of paragraph ordering.

Results Table 5 shows the experimental results. From the lower scores of the *Mean-Publication-Year* and *Paragraph-Size*, it appears difficult to determine paragraph order solely from such statistical features. Similar to the findings in citation clustering, the methods that employ an LLM achieve higher scores, indicating that LLM-based approaches can accurately determine paragraph order.

When comparing the LLM$_{\text{base}}$ with our proposed method, we find that our approach generally achieves higher scores; however, for LLaMA, the LLM$_{\text{base}}$ shows the highest score. This suggests that directly generating paragraph order with an LLM can be an effective strategy for this task. Within our proposed method, the $\mathcal{M}_{\text{base}}$ yields the lowest score, indicating that augmenting the number of paper features leads to more accurate paragraph ordering. In addition, in the ablation study, $\mathcal{M}_{\setminus\text{seed}}$ consistently shows low scores across all variants, demonstrating that incorporating information from the source paper is essential for determining paragraph order.

5 Citation Ordering within a Paragraph

Then, we address citation ordering within a paragraph, which arranges the cited papers within each paragraph in the order they should be mentioned. As shown in Figure 4, we construct a directed graph whose nodes correspond to cited papers, and then we determine an optimal ordering of these nodes.

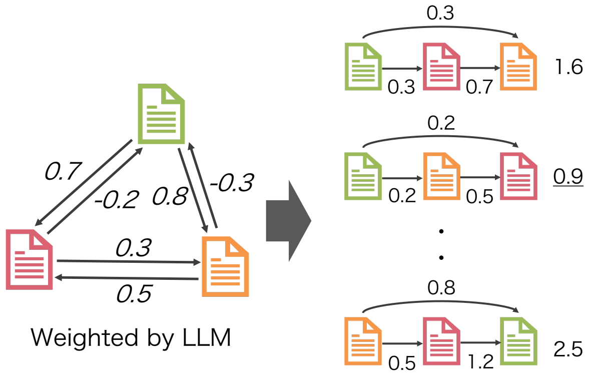

Figure 4: Overview of the citation ordering within a paragraph. The values in the left graph represent v_{ab}, while those in the right graph represent l_{ij}.

5.1 Graph Construction

We build a directed graph in which each node represents a cited paper and every pair of nodes is connected by edges in both directions. When determining the order of cited papers within a paragraph, their relative positions should be taken into account. Swapping papers that are far apart is generally unproblematic, as they often cover different topics. In contrast, swapping adjacent papers can disrupt the logical flow. Accordingly, we assign edge weights that decrease with distance, imposing higher penalties on adjacent swaps and lower penalties on distant ones.

Specifically, we fine-tune an LLM on a regression task where, given two papers a and b in that order, the model predicts $v_{ab} = \frac{1}{\pi^{-1}(b)-\pi^{-1}(a)}$. Here, $\pi^{-1}(x)$ denotes the mention position of paper x within the paragraph. v_{ab} is positive when two papers are input in the same order as given in the related work section, and negative when they are input in reverse order; its absolute value decreases as the distance between their citation positions increases. For example, if the papers with citation orders one and two are input in that order, v outputs 1, and if the papers with citation orders three and one are input in that order, v outputs -0.5.

As in citation clustering, we construct six models with different input configurations: $\mathcal{M}_{\text{base}}$, $\mathcal{M}_{\text{extd}}$, $\mathcal{M}_{\backslash\text{year}}$, $\mathcal{M}_{\backslash\text{auth}}$, $\mathcal{M}_{\backslash\text{cite}}$, and $\mathcal{M}_{\backslash\text{seed}}$.

5.2 Ordering Method

We determine the final citation order within each paragraph by exhaustively searching all possible permutations to find the permutation π that minimizes the objective defined in Equation (6):

$$l_{ij} = \left| v_{\pi(i)\pi(j)} - \frac{1}{j-i} \right|, \tag{5}$$

$$\arg\min_{\pi} \sum_{1 \le i < j \le n} l_{ij}, \tag{6}$$

where $\pi(i)$ is the paper at position i of permutation π, and l_{ij} is defined as the absolute difference between the $v_{\pi(i)\pi(j)}$ and the inverse of the distance of i, j in the permutation. This objective function measures the predicted distance error, and the permutation π that minimize this sum is taken as the most coherent order.

5.3 Experiments

Settings The fine-tuning settings were the same as those used for citation clustering. We evaluated five different prompts for both the $\mathcal{M}_{\text{base}}$ and $\mathcal{M}_{\text{extd}}$ using the development set and selected the one with the smallest mean squared error of regression task. As an evaluation metric, we used Spearman's rank correlation coefficient ρ and Kendall rank correlation coefficient τ. Exhaustively searching all permutations in Equation (6) becomes computationally infeasible for paragraphs with many cited papers. Therefore, to ensure that we could find an optimal solution, we limited our evaluation to paragraphs with fewer than 10 cited papers (777 out of 921 total).[11]

Compared Methods We constructed two baseline methods to verify the effectiveness of our approach. The first was Year only baseline (*Year-Only*). This model simply sorted the cited papers in ascending order based on their publication year. The second was LLM$_{\text{base}}$. The LLM$_{\text{base}}$ took the source paper and all cited papers in a paragraph as input, and was fine-tuned to generate the citation order. The fine-tuning settings for the LLM$_{\text{base}}$ are identical to those used in the other tasks, and the model inputs are the same as those for $\mathcal{M}_{\backslash\text{cite}}$. The prompt tuning was done in the same setting as before.

[11] We verified that the computation is feasible when fewer than 12 papers appear in a paragraph. In the test set, 840 instances (91% of the data) met this condition. In this study, "paragraph" refers to the highest-level division within the related work section, as described in Section 2. It may correspond to a relatively long unit such as a subsection. Paragraphs that cite more than 12 papers are often composed of several smaller paragraphs. Therefore, adjusting the number of clusters usually addresses these cases.

			ρ	τ		
Year-Only			.447	**.418**		
	LLaMA		Mistral		Qwen	
	ρ	τ	ρ	τ	ρ	τ
LLM$_{\text{base}}$	.414	.387	.376	.346	.390	.360
$\mathcal{M}_{\text{base}}$	.354	.313	.339	.301	.352	.309
$\mathcal{M}_{\text{extd}}$	.452	.414	.446	.409	**.456**	.417
$\mathcal{M}_{\setminus\text{year}}$	.403	.369	.423	.390	.420	.387
$\mathcal{M}_{\setminus\text{auth}}$	.441	.402	.451	.413	.452	.413
$\mathcal{M}_{\setminus\text{cite}}$	.449	.406	.438	.399	.439	.403
$\mathcal{M}_{\setminus\text{seed}}$	.443	.408	.442	.408	.441	.404

Table 6: Experimental results of citation ordering within a paragraph.

Results Table 6 shows the experimental results for citation ordering within a paragraph. The Year only baseline outperforms the LLM$_{\text{base}}$ of all LLMs, suggesting that publication year considerably influences this task. Moreover, $\mathcal{M}_{\text{extd}}$ in Qwen, the best-performing proposed method of all LLMs, exceeds the Year baseline by about 0.09 points in ρ, but the τ is slightly lower. This indicates that there is room for improvement in both the estimation of weights and the definition of the objective function.

Among the models that utilize an LLM, $\mathcal{M}_{\text{base}}$ performs worse than the LLM$_{\text{base}}$. However, incorporating additional paper features improves performance and can surpass LLM$_{\text{base}}$. Within our proposed methods, adding paper features consistently improves performance compared to $\mathcal{M}_{\text{base}}$, indicating that titles and abstracts are insufficient for this task. Furthermore, $\mathcal{M}_{\setminus\text{year}}$ shows the lowest score among the models except for $\mathcal{M}_{\text{base}}$, suggesting that publication year is a crucial feature. $\mathcal{M}_{\text{extd}}$ achieves higher score overall, indicating that all features may also be useful in citation ordering within a paragraph.

6 Integrated Evaluation

Up to this point, we evaluated citation clustering, paragraph ordering, and citation ordering within a paragraph as separate tasks. Finally, we conduct an integrated evaluation of the full pipeline.

While the weights obtained from the citation clustering can be reused, the weights for paragraph ordering and citation ordering within a paragraph cannot be directly reused because the paragraph structure changes with the clustering output. Thus, starting from the set of paragraphs generated by citation clustering, the paragraph order and citation order are determined by reconstructing the graph.

Following the same framework as before, we evaluate six LLM input configurations: $\mathcal{M}_{\text{base}}$, $\mathcal{M}_{\text{extd}}$, $\mathcal{M}_{\setminus\text{year}}$, $\mathcal{M}_{\setminus\text{auth}}$, $\mathcal{M}_{\setminus\text{cite}}$, and $\mathcal{M}_{\setminus\text{seed}}$. Because paragraph representations do not use citation relationships, $\mathcal{M}_{\setminus\text{cite}}$ is not applicable to paragraph ordering. Thus, in $\mathcal{M}_{\text{extd}}$ we use citation relationships only for citation clustering and citation ordering within-paragraph ordering; for paragraph ordering we use the same inputs as $\mathcal{M}_{\setminus\text{cite}}$.

6.1 Experiments

Settings Evaluation followed the same settings as in citation ordering within a paragraph. If citation clustering resulted in any paragraph being assigned more than ten papers, that instance was excluded from evaluation. Therefore the set of evaluable data differs because the number of papers in each paragraph varies with each model's clustering results. To ensure a fair evaluation, we considered only the common subset of data evaluable across all models. We evaluated 105 instances with LLaMA and Qwen, and 104 instances with Mistral (the test set comprises 286 instances in total).

Compared Methods We constructed three baseline methods to verify the effectiveness of our approach. The first was Year only baseline (*Year-Only*). This model simply sorted papers in ascending order of their publication year. The second baseline directly generated the related work section in a zero-shot setting. This model provides the LLM with a prompt to generate a related work section based on a set of cited papers and the source paper. The evaluation was performed by extracting the citation order from the generated related work section. In this study, we used OpenAI o1 (OpenAI, 2024) and OpenAI o3-mini [12] as LLMs. The third baseline was LLM$_{\text{base}}$. The LLM$_{\text{base}}$ took the source paper and all cited papers as input, and is fine-tuned to generate the citation order. The fine-tuning settings and the model inputs are the same as those used for the LLM$_{\text{base}}$ in the other tasks.

6.2 Results

Table 7 shows the experimental results. While the Year only baseline achieved high scores for citation ordering within a paragraph, it performs poorly when applied to the entire related work section.

[12] https://openai.com/index/o3-mini-system-card/

	LLaMA		Mistral		Qwen	
	ρ	τ	ρ	τ	ρ	τ
Year-Only	.271	.238	.264	.233	.258	.228
OpenAI o1	.111	.091	.103	.085	.090	.074
o3-mini	.128	.110	.113	.100	.116	.102
LLM_{base}	.300	.261	.400	.323	.322	.272
$\mathcal{M}_{\text{base}}$	.272	.239	.266	.234	.298	.257
$\mathcal{M}_{\text{extd}}$	.462	.402	**.515**	**.438**	.371	.322
$\mathcal{M}_{\setminus\text{year}}$	.435	.364	.488	.412	.467	.393
$\mathcal{M}_{\setminus\text{auth}}$	.468	.402	.480	.411	.364	.325
$\mathcal{M}_{\setminus\text{cite}}$	.428	.369	.406	.326	.064	.016
$\mathcal{M}_{\setminus\text{seed}}$	.337	.299	.348	.310	.387	.335

Table 7: Experimental results of integrated evaluation.

This finding indicates that while sorting based on publication year is effective within a small unit such as a single paragraph, it is not necessarily effective for the related work section as a whole. In addition, the OpenAI o1 and OpenAI o3-mini show the lowest scores among all models and methods. Although current LLMs are capable of generating high-quality text, we have shown that reproducing the citation order of a human-written related work section from a collection of cited papers is difficult. In contrast, the LLM_{base} shows stable performance, suggesting that fine-tuning the LLM can achieve high performance. Most of the proposed methods outperform the LLM_{base}, indicating the effectiveness of our approach over simply using an LLM, as shown in the individual tasks.

Within the proposed methods, $\mathcal{M}_{\text{base}}$ shows the lowest score, indicating that expanding the input features can lead to more appropriate ordering. Additionally, $\mathcal{M}_{\setminus\text{seed}}$ tends to yield a score closest to $\mathcal{M}_{\text{base}}$, suggesting that information of the source paper may be the most critical feature for citation arrangement. Furthermore, model comparisons reveal that Qwen generally attains lower scores than LLaMA and Mistral. In particular, the $\mathcal{M}_{\setminus\text{cite}}$ variant shows a markedly lower score, which we believe is due to the influence of the citation clustering performance, suggesting that prompt-tuning may be necessary for each model.

7 Related Work

7.1 Citation Recommendation

Citation recommendation is a task that suggests relevant research to cite during paper writing and can be broadly categorized into local and global citation recommendation.

Local citation recommendation, first proposed by He et al. (2010), uses contextual information from the source paper to recommend appropriate citations. Initially, methods utilizing TF-IDF and neural networks were introduced (He et al., 2010; Huang et al., 2014). Recently, methods based on deep neural networks and transformer-based approaches have shown high performance (Gu et al., 2022; Ghosh Roy and Han, 2024). However, because local citation recommendation relies on contextual information, it is not well-suited for assisting during the early stages of writing.

Global citation recommendation aims to recommend relevant research for the entire paper (McNee et al., 2002). The advancement of deep learning has enabled high performance through models such as SPECTER (Cohan et al., 2020) and SciNCL (Ostendorff et al., 2022), which are trained on scientific papers. However, existing studies mainly focus on whether a paper should be cited, and the determination of citation order remains underexplored.

7.2 Scientific Paper Generation

Related work section of scientific papers has been a particular focus in research on automatic text generation. Research on generating the related work section was first undertaken by Hoang and Kan (2010). With the development of deep learning techniques, many generative methods employing transformer-based models have been proposed (AbuRa'ed et al., 2020; Chen et al., 2022). In recent years, research has also focused on the automatic generation of related work section using LLMs such as GPT-4, which enables high-quality sentence generation (Martin-Boyle et al., 2024; Li and Ouyang, 2025). However, most existing methods predefine the citation order of cited papers, and a mechanism to automatically determine the citation order remains underexplored (Li and Ouyang, 2024).

Some studies have attempted to generate entire scientific papers (Taylor et al., 2022; Lu et al., 2024; Yamada et al., 2025). Taylor et al. (2022) developed a model called Galactica and released a demo that automatically generates survey papers. Additionally, Lu et al. (2024) introduced a model called The AI Scientist, in which an LLM handles everything from idea generation to full paper creation. However, the quality of the papers automatically generated by these models is insufficient, and that scientific paper generation requires a more fine-grained approach.

8 Conclusion

In this paper, we proposed a novel task, citation arrangement, which aims to enable fully automatic generation of related work sections by arranging cited papers in an appropriate order. To address this task, we decomposed citation arrangement into three tasks: citation clustering, paragraph ordering, and citation ordering within a paragraph, and proposed an approach that incorporates LLM and a graph-based method for each task. For our experiments, we constructed a dataset from PDFs of papers published at conferences on natural language processing. Our experimental results demonstrated that our method achieves performance comparable to or better than the baselines for each of the tasks, and in addition, for the integrated evaluation, it considerably outperforms some of the baselines. Furthermore, we demonstrated that task performance is influenced by the input features, and ablation studies identified that publication year and source paper information are often particularly beneficial.

Limitations

This study has three main limitations. The first limitation concerns estimating the number of paragraphs. In our current approach, we extract the number of paragraphs directly from the source paper during citation clustering. However, fully automating the process would require automatically estimating the appropriate number of paragraphs. Although we believe it may be possible to estimate the number of paragraphs based on the edge weights of the constructed graph, we leave this topic outside the scope of this paper.

The second limitation concerns the development of an effective algorithm for paper ordering when a paragraph contains a large number of cited papers. For our evaluation, we excluded cases with a high number of citations per paragraph. In practice, paragraphs containing a large number of cited papers are relatively rare, and clustering allows us to control the number of papers per paragraph; therefore we do not consider this to be a notable limitation.

The third limitation is that our evaluation does not assess the performance of generating the related work section. In this study, we focused on citation arrangement as a crucial step toward fully automatic related work section generation. However, the extent to which our approach improves the overall quality of automatically generated related work sections remains unclear. Future work will evaluate the integration of our method with complete related work section generation.

Acknowledgments

This work was supported by JST Moonshot R&D Grant Number JPMJMS2033 and JST BOOST Grant Number JPMJBS2422.

References

Ahmed AbuRa'ed, Horacio Saggion, Alexander Shvets, and Àlex Bravo. 2020. Automatic related work section generation: experiments in scientific document abstracting. *Scientometrics*, 125(3):3159–3185.

Amit Bagga and Breck Baldwin. 1998. Entity-based Cross-Document Coreferencing Using the Vector Space Model. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING)*, pages 79–85.

Xiuying Chen, Hind Alamro, Mingzhe Li, Shen Gao, Rui Yan, Xin Gao, and Xiangliang Zhang. 2022. Target-aware Abstractive Related Work Generation with Contrastive Learning. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 373–383.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel Weld. 2020. SPECTER: Document-level Representation Learning using Citation-informed Transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2270–2282.

Sayar Ghosh Roy and Jiawei Han. 2024. ILCiteR: Evidence-grounded Interpretable Local Citation Recommendation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING)*, pages 8627–8638.

Nianlong Gu, Yingqiang Gao, and Richard H. R. Hahnloser. 2022. Local Citation Recommendation with Hierarchical-Attention Text Encoder and SciBERT-Based Reranking. In *Advances in Information Retrieval*, pages 274–288.

Qi He, Jian Pei, Daniel Kifer, Prasenjit Mitra, and Lee Giles. 2010. Context-aware citation recommendation. In *Proceedings of the 19th International Conference on World Wide Web (WWW)*, pages 421–430.

Cong Duy Vu Hoang and Min-Yen Kan. 2010. Towards Automated Related Work Summarization. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, pages 427–435.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-Rank Adaptation of Large Language Models. In *The Tenth International Conference on Learning Representations (ICLR)*.

Wenyi Huang, Zhaohui Wu, Prasenjit Mitra, and C. Lee Giles. 2014. Refseer: a citation recommendation system. In *Proceedings of the 14th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 371–374.

Xiangci Li and Jessica Ouyang. 2024. Related Work and Text Generation: A Survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 13846–13864.

Xiangci Li and Jessica Ouyang. 2025. Explaining Relationships Among Research Papers. In *Proceedings of the 31st International Conference on Computational Linguistics (COLING)*, pages 1080–1105.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. 2020. S2ORC: The Semantic Scholar Open Research Corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4969–4983.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. 2024. The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery. arXiv preprint arXiv:2408.06292.

Xiaoqiang Luo. 2005. On Coreference Resolution Performance Metrics. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT-EMNLP)*, pages 25–32.

Anna Martin-Boyle, Aahan Tyagi, Marti A. Hearst, and Dongyeop Kang. 2024. Shallow Synthesis of Knowledge in GPT-Generated Texts: A Case Study in Automatic Related Work Composition. arXiv preprint arXiv:2402.12255.

Sean M. McNee, Istvan Albert, Dan Cosley, Prateep Gopalkrishnan, Shyong K. Lam, Al Mamunur Rashid, Joseph A. Konstan, and John Riedl. 2002. On the recommending of citations for research papers. In *Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work (CSCW)*, pages 116–125.

OpenAI. 2024. Openai o1 System Card. arXiv preprint arXiv:2412.16720.

Malte Ostendorff, Nils Rethmeier, Isabelle Augenstein, Bela Gipp, and Georg Rehm. 2022. Neighborhood Contrastive Learning for Scientific Document Representations with Citation Embeddings. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 11670–11688.

Shaurya Rohatgi, Yanxia Qin, Benjamin Aw, Niranjana Unnithan, and Min-Yen Kan. 2023. The ACL OCL Corpus: Advancing Open Science in Computational Linguistics. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 10348–10361.

Jianbo Shi and J. Malik. 2000. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A Large Language Model for Science. arXiv preprint arXiv:2211.09085.

Yutaro Yamada, Robert Tjarko Lange, Cong Lu, Shengran Hu, Chris Lu, Jakob Foerster, Jeff Clune, and David Ha. 2025. The AI Scientist-v2: Workshop-Level Automated Scientific Discovery via Agentic Tree Search. arXiv preprint arXiv:2504.08066.

A Model Inputs

Table 8 shows the input features of each proposed method when we fine-tuned the LLM. “Citation linkage” indicates whether a citation relationship exists between two input papers, and “Source paper” refers to the title and abstract of the source paper. For paragraph ordering, the entire set of cited paper is used as input, which prevents us from separately considering citation relationships. Consequently, citation linkage is not used in this task.

B Details of Experimental Settings

In the proposed methods of citation clustering and citation ordering within a paragraph, fine-tuning and graph construction were performed by creating cited paper pairs, and paragraph ordering uses paragraph pairs. On the other hand, fine-tuning in the LLM_{base} was performed on a source paper basis. Therefore, the number of data used for fine-tuning varies by model.

Fine-tuning for proposed methods and LLM_{base} were conducted on 4 NVIDIA RTX A6000 GPUs, each with 48 GB memory. Fine-tuning for LLM_{base} in all tasks took about 6 hours. Fine-tuning the proposed methods for citation clustering and citation ordering within a paragraph took about 12 hours, and fine-tuning the proposed methods for paragraph ordering took about 6 hours.

C Prompts

Tables 9 and 10 show prompts for the LLM_{base} for paragraph ordering and citation ordering within

	Method	Title	Abstract	Year of publication	Author	Citation linkage	Source paper
Citation clustering, Citation ordering within a paragraph	$\mathcal{M}_{\mathrm{base}}$	✓	✓				
	$\mathcal{M}_{\mathrm{extd}}$	✓	✓	✓	✓	✓	✓
	$\mathcal{M}_{\backslash\mathrm{year}}$	✓	✓		✓	✓	✓
	$\mathcal{M}_{\backslash\mathrm{auth}}$	✓	✓	✓		✓	✓
	$\mathcal{M}_{\backslash\mathrm{cite}}$	✓	✓	✓	✓		✓
	$\mathcal{M}_{\backslash\mathrm{seed}}$	✓	✓	✓	✓	✓	
	$\mathrm{LLM}_{\mathrm{base}}$	✓	✓	✓	✓		✓
Paragraph ordering	$\mathcal{M}_{\mathrm{base}}$	✓	✓				
	$\mathcal{M}_{\mathrm{extd}}$	✓	✓	✓	✓		✓
	$\mathcal{M}_{\backslash\mathrm{year}}$	✓	✓		✓		✓
	$\mathcal{M}_{\backslash\mathrm{auth}}$	✓	✓	✓			✓
	$\mathcal{M}_{\backslash\mathrm{cite}}$	✓	✓	✓	✓		✓
	$\mathcal{M}_{\backslash\mathrm{seed}}$	✓	✓	✓	✓		
	$\mathrm{LLM}_{\mathrm{base}}$	✓	✓	✓	✓		✓

Table 8: Input information used by each proposed method.

You are a researcher specializing Natural Language Processing.
I am writing scientific paper with the following title and abstract.
Title:{Title of Source Paper}
Abstract:{Abstract of Source Paper}
Please sort the following {# of paragraphs} clusters based on their topics to determine the order in which they should appear in the related work section.
references
Cluster1
Title:{Tit. of paper1 in paragraph1}, Abstract:{Abs. of paper1 in paragraph1}, Author:{Auth. of paper1 in paragraph1}, Year of Publication:{Year of Pub. of paper1 in paragraph1}
Title:{Tit. of paper2 in paragraph1}, Abstract:{Abs. of paper2 in paragraph1}, Author:{Auth. of paper2 in paragraph1}, Year of Publication:{Year of Pub. of paper2 in paragraph1}
. . .
Cluster2
Title:{Tit. of paper1 in paragraph2}, Abstract:{Abs. of paper1 in paragraph2}, Author:{Auth. of paper1 in paragraph2}, Year of Publication:{Year of Pub. of paper1 in paragraph2}
Title:{Tit. of paper2 in paragraph2}, Abstract:{Abs. of paper2 in paragraph2}, Author:{Auth. of paper2 in paragraph2}, Year of Publication:{Year of Pub. of paper2 in paragraph2}
. . .

Table 9: Prompt used in $\mathrm{LLM}_{\mathrm{base}}$ of paragraph ordering.

a paragraph. Table 11 shows the prompt for the $\mathcal{M}_{\mathrm{base}}$ of citation clustering, Table 12 shows the prompt for $\mathcal{M}_{\mathrm{extd}}$ of citation clustering and citation ordering within a paragraph, and Table 13 shows the prompt for paragraph ordering. Table 14 show the prompt used by OpenAI o1 and OpenAI o3-mini to generate the related work section.

You are a researcher specializing Natural Language Processing.

I am writing scientific paper with the following title and abstract.

Title:{Title of Source Paper}

Abstract:{Abstract of Source Paper}

Please order the following {# of cited papers} references in the order they should be mentioned in the paper. You can rely on the year of publication to order references.

references

Title:{Tit. of paper1}, Abstract:{Abs. of paper1}, Author:{Auth. of paper1}, Year of Publication:{Year of Pub. of paper1}

Title:{Tit. of paper2}, Abstract:{Abs. of paper2}, Author:{Auth. of paper2}, Year of Publication:{Year of Pub. of paper2}

...

Table 10: Prompt used in LLM$_{\text{base}}$ of citation ordering within a paragraph.

Please output 1 if **Cited Paper1** and **Cited Paper2** belong in the same paragraph in the related work section, and 0 otherwise.

Paper1

Title:{Tit. of paper1}, Abstract:{Abs. of paper1}

Paper2

Title:{Tit. of paper2}, Abstract:{Abs. of paper2}

Table 11: Prompt used in $\mathcal{M}_{\text{base}}$ of citation clustering.

Seed Paper

Title:{Title of Source Paper}

Abstract:{Abstract of Source Paper}

Paper1

Title:{Tit. of paper1}, Abstract:{Abs. of paper1}, Author:{Auth. of paper1}, Year of Publication:{Year of Pub. of paper1}

Paper2

Title:{Tit. of paper2}, Abstract:{Abs. of paper2}, Author:{Auth. of paper2}, Year of Publication:{Year of Pub. of paper2}

Citation linkage

{True or False}

Table 12: Prompt used in $\mathcal{M}_{\text{extd}}$ of citation clustering and all models of citation ordering within a paragraph.

Seed Paper

***Cluster1**

Title:{Tit. of paper1 in paragraph1}, Abstract:{Abs. of paper1 in paragraph1}, Author:{Auth. of paper1 in paragraph1}, Year of Publication: {Year of Pub. of paper1 in paragraph1}

Title:{Tit. of paper2 in paragraph1}, Abstract:{Abs. of paper2 in paragraph1}, Author:{Auth. of paper2 in paragraph1}, Year of Publication:{Year of Pub. of paper2 in paragraph1}

...

Cluster2

Title:{Tit. of paper1 in paragraph2}, Abstract:{Abs. of paper1 in paragraph2}, Author:{Auth. of paper1 in paragraph2}, Year of Publication:{Year of Pub. of paper1 in paragraph2}

Title:{Tit. of paper2 in paragraph2}, Abstract:{Abs. of paper2 in paragraph2}, Author:{Auth. of paper2 in paragraph2}, Year of Publication:{Year of Pub. of paper2 in paragraph2}

...

Table 13: Prompt used in proposed method of paragraph ordering.

You are a researcher specializing Natural Language Processing. I am writing scientific paper with the following title and abstract.

Title:{Title of Source Paper}

Abstract:{Abstract of Source Paper}

Please generate the related work section of this paper in exactly {# of paragraphs} paragraphs by citing all references only once and rearranging them in the optimal order. Also, please use markers such as [1] and [2] when citing it.

references

Title:{Tit. of paper1}, Abstract:{Abs. of paper1}, Author:{Auth. of paper1}, Year of Publication:{Year of Pub. of paper1}

Title:{Tit. of paper2}, Abstract:{Abs. of paper2}, Author:{Auth. of paper2}, Year of Publication:{Year of Pub. of paper2}

...

Table 14: Prompt input into OpenAI o1 and OpenAI o3-mini to generate related work section.