Iterative Introspection based Refinement: Boosting Multi-Document Scientific Summarization with Large Language Models

Anonymous ACL submission

Abstract

The current task setting and constructed datasets for Multi-Document Scientific Summarization (MDSS) have led to a significant gap between existing research and practical applications. However, the emergence of Large Language Models (LLMs) provides us with an opportunity to address MDSS from a more practical perspective. To this end, we redefine MDSS task based on the scenario that automatically generates the entire related work section, and then construct a corresponding new dataset, ComRW. We first conduct a comprehensive evaluation of the performance of LLMs on the newly defined task, and identify three major deficiencies in their ability to address MDSS task: low coverage of reference papers, disorganized structure, and high redundancy. To alleviate these three deficiencies, we propose an **I**terative **I**ntrospection based **R**efinement (IIR) method that utilizes LLMs to generate higher-quality summaries. The IIR method uses prompts equipped with Chain-of-Thought and fine-grained operators to treat LLMs as an evaluator and a generator to evaluate and refine the three deficiencies, respectively. We conduct thorough automatic and human evaluation to validate the effectiveness of our method. The results demonstrate that the proposed IIR method can effectively mitigate the three deficiencies and improve the quality of summaries generated by LLMs. Moreover, our exploration provides insights for better addressing MDSS task with LLMs.

1 Introduction

Multi-**D**ocument **S**cientific **S**ummarization (MDSS) aims to generate a concise and condensed summary for a group of topic-relevant scientific articles. In order to meet the training demand of data-driven abstractive summarization models, the existing MDSS studies (Chen et al., 2021, 2022; Wang et al., 2023a) mainly focus on the scenario of automatically generating related work of academic papers. When constructing the corresponding datasets, such as Multi-Xscience (Lu et al., 2020), TAD (Chen et al., 2022) and TAS2 (Chen et al., 2022), individual paragraphs of a related work section are used as gold standard summaries, and the abstract section of the target paper and the reference papers are used as input documents. Such task setting and constructed datasets have greatly advanced research on MDSS.

> Recent studies usually present the task of relation classification in a supervised perspective, and traditional supervised approaches can be divided into feature based methods and kernel methods.
>
> Feature based methods focus on extracting and selecting relevant feature for relation classification. Kambhatla (2004) leverages lexical, syntactic and semantic features, and feeds them to a maximum entropy model. Hendrickx et al. (2010) show that the winner of SemEval-2010 Task 8 used the most types of features and resources, among all participants. Nevertheless, it is difficult to find an optimal feature set, since traversing all combinations of features is time-consuming for feature based methods.
>
> To remedy the problem of feature selection mentioned above, kernel methods represent the input data by computing the structural commonness between sentences, based on carefully designed kernels. Mooney and Bunescu (2005) split sentences into subsequences and compute the similarities using the proposed subsequence kernel. Bunescu and Mooney (2005) propose a dependency tree kernel and extract information from the Shortest Dependency Path (SDP) between marked entities. Since kernel methods require similarity computation between input samples, they are relatively computationally expensive when facing large-scale datasets.

Figure 1: Example of related work section

However, we argue that the above task setting and constructed datasets induce three drawbacks: (1) The gold standard summary is merely a paragraph of a related work section in the current task setting. However, the content and structural styles of paragraphs in different positions of the related work section vary significantly, as shown in Figure 1. Therefore, datasets built based on this task setting are prone to problems like missing context and incomplete structure. (2) The input documents of the datasets are only the abstract section of the papers. However, the information required to generate the summary may come from other sections of the papers. Therefore, incomplete input information may make it difficult to infer parts of the gold summary from the input, known as the intrinsic hallucination issue (Maynez et al., 2020; Ji et al., 2023). (3) In existing datasets, all citation markers (such as "*Kambhatla (2004)*" in Figure 1) are normalized to a particular symbol "*@cite*", making it difficult to locate different reference papers in the generated summaries. The above three drawbacks have led to a significant gap between existing

research on MDSS and practical applications, resulting in the neglect of content consistency and structural rationality which should be emphasized in MDSS.

Recently, Large Language Models (LLMs), such as GPT-3.5 (Ouyang et al., 2022) and GPT-4 (Achiam et al., 2023), have demonstrated remarkable capabilities in tackling numerous reasoning and text generation tasks. These capabilities offer exciting new solutions for MDSS task, that is, leveraging the powerful text generation and in-context learning (Brown et al., 2020) ability of LLMs to solve MDSS task more flexibly from a perspective closer to practical applications.

In this regard, although previous researchers (Haman and Školník, 2023; Huang and Tan, 2023; Agarwal et al., 2024; Martin-Boyle et al., 2024) have attempted to utilize LLMs to address MDSS from the perspective of practical applications, their work has only stayed at the level of qualitative analysis of LLMs. For instance, Martin-Boyle et al. (2024) use citation graphs to analyze the difference in structural complexity between human-written summaries and GPT-4 generated summaries. Huang and Tan (2023) discuss the role and advantages of LLMs in assisting the literature review process. However, we argue that these studies fail to provide a systematic and comprehensive evaluation of the performance of LLMs on MDSS task by constructing reasonable datasets, rendering the shortcomings of LLMs in addressing MDSS remaining unknown.

To solve the above issue, we start from the perspective of practical applications of MDSS and redefine MDSS task as *given the full text of a target paper and all the reference papers cited by it as input documents, the goal is to generate the entire related work section of the target paper.* Based on the definition, we construct a new dataset called **ComRW**, which contains 30 instances, each including a target paper, several reference papers, and a gold summary.

Based on ComRW dataset, we conduct a comprehensive evaluation of the performance of LLMs on MDSS task. Specifically, the evaluation is conducted on GPT-3.5 and GPT-4, and compared with fully-trained models BART (Lewis et al., 2020) and EDITSum (Wang et al., 2023a). The results reveal that although LLMs are not yet comparable to EDITSum in terms of ROUGE (Lin, 2004) metric, both BERTScore (Zhang et al., 2019) metric and human evaluation results indicate that the quality of summaries generated by LLMs is higher, showcasing their strong capability in addressing MDSS task. According to the results, we also identify three major deficiencies of LLMs in generating summaries: (1) **Low Coverage of Reference Papers**: LLMs tend to omit some input reference papers in the generated summaries; (2) **Disorganized Structure**: the structure of summaries generated by LLMs is unclear, with disorganized sub-topics; (3) **High Redundancy**: the summaries generated by LLMs contain much redundant or repetitive content.

Regarding the above three deficiencies, we further propose an **I**terative **I**ntrospection based **R**efinement (IIR) method that utilizes LLMs to generate higher-quality summaries. Specifically, IIR divides the summary generation process into draft generation and iterative refinement stages. While the concept of iterative refinement has been widely employed in text editing (Iso et al., 2020; Awasthi et al., 2019; Schick et al., 2022), the novelty of our work lies in leveraging the powerful natural language evaluation capability (Liu et al., 2023a; Fu et al., 2023; Chiang and Lee, 2023) and instruction-following ability of LLMs by designing reasonable prompts. Concretely, we design prompts equipped with Chain-of-Thought (Wei et al., 2022) and fine-grained operators to treat LLMs as an evaluator and a generator to evaluate and refine the three deficiencies, respectively and iteratively.

We conduct both automatic and human evaluation to validate the effectiveness of our IIR method. The results indicate that IIR method can effectively alleviate the three deficiencies of LLMs, thereby enhancing the quality of generated summaries.

Our contributions are: (1) We redefine MDSS task from the perspective of practical applications and conduct a comprehensive evaluation of the performance of LLMs on MDSS. (2) We propose IIR method to mitigate the three deficiencies of LLMs in addressing MDSS task. (3) Both automatic and human evaluations validate the effectiveness of our IIR method.

2 Task Redefinition

The existing task setting of MDSS and constructed datasets lead to a significant gap between existing research on MDSS and practical applications. Hence, in this paper, we redefine MDSS task from a more practical perspective. The new definition is: *Given the full text of a target paper that needs to generate a related work section, along with the full text of all reference papers in the related work*

section of the target paper as input, the goal is to generate the entire related work section of the target paper.

Our new definition differs from the previous one in the following three aspects: (1) In our setting, the gold summary is the full text of the related work section, avoiding the problems of missing context and incomplete structure caused by using only paragraphs as gold summary. (2) In our setting, the input documents consist of the full texts of the target paper and reference papers, thus avoiding the intrinsic hallucination issue caused by incomplete input information. (3) We retain all citation markers within the gold summary, which facilitates precise location of different reference papers and enables us to assess content consistency of the generated summary.

3 Basic Performance Analysis of LLMs

According to the above task definition, we first construct a new dataset ComRW. The construction process and dataset analysis of ComRW are introduced in Appendix A. Please refer to Appendix A for more details.

In this section, we conduct a comprehensive evaluation of LLMs' performance on MDSS task based on ComRW dataset.

3.1 Evaluation Setup

Model Selection We choose GPT-3.5[1] (Ouyang et al., 2022) and GPT-4[2] (Achiam et al., 2023) as representatives of LLMs. We use zero-shot prompting (0-shot) and one-shot prompting (1-shot) to interact with LLMs. The prompt design strategies for LLMs are introduced in Appendix B. To effectively demonstrate the performance of LLMs, we compare them with previous fully-trained MDSS models. For this purpose, we choose the state-of-the-art MDSS model EDITSum (Wang et al., 2023a) and the widely-used pretrained text generation model BART (Lewis et al., 2020) for comparison. The detailed settings for EDITSum and BART are introduced in Appendix C.

Evaluation Metrics We use ROUGE-1/2/L (R-1/R-2/R-L) (Lin, 2004) and BERTScore (BS) (Zhang et al., 2019) as the automatic metrics. We also employ a LLM-based metric G-Eval (Liu et al., 2023a), which utilizes LLM with Chain-of-Thought and a form-filling paradigm to assess summary quality, with scores ranging from 1 to 5.

[1]We use the `gpt-3.5-turbo-0125` variant.

[2]We use the `gpt-4-0125-preview` variant.

Table 1: Automatic evaluation of LLMs and other models on ComRW dataset.

Model	R-1(%)	R-2(%)	R-L(%)	BS(%)	G-Eval
BART	45.3	12.11	43.35	84.33	3.69
EDITSum	**48.68**	**12.54**	**47**	84.84	3.73
GPT-3.5 (0-shot)	44.67	11.2	42.31	85.87	4.02
GPT-3.5 (1-shot)	46.3	12.14	43.65	85.82	4.01
GPT-4 (0-shot)	47.25	11.34	44.46	**86.19**	4.08
GPT-4 (1-shot)	47.38	11.61	44.75	86.08	**4.10**

Furthermore, we also conduct human evaluation to ensure a more reliable and comprehensive assessment.

3.2 Evaluation Results

3.2.1 Automatic Evaluation

The result of automatic evaluation is shown in Table 1. We conclude two observations from it.

Firstly, apart from GPT-3.5 (0-shot), other LLM variants are able to outperform BART on most metrics such as ROUGE-1/L, BERTScore, and G-Eval. However, when compared with EDITSum, we can find that all LLMs variants lag behind EDITSum on ROUGE metric. The best-performing LLM variant is GPT-4 (1-shot), achieving ROUGE-1/2/L scores of 47.38/11.61/44.75, which show a noticeable gap compared with EDITSum's performance of 48.68/12.54/47. However, on BERTScore and G-Eval, all LLM variants surpass EDITSum. The best-performing model on BERTScore, GPT-4 (0-shot), achieves a score of 86.19, which exceeds EDITSum by 1.35%. Similarly, the leading model on G-Eval, GPT-4 (1-shot), achieves a score of 4.10, exceeding EDITSum by 0.37. The above result demonstrates that LLMs have strong zero-shot learning ability and can achieve satisfactory results in MDSS task.

Secondly, for GPT-3.5, adding a demonstration in the prompt leads to a noticeable improvement in ROUGE scores. However, for GPT-4, the performance difference between zero-shot prompting and one-shot prompting is not apparent. The reason may be that GPT-4 has better zero-shot learning ability.

3.2.2 Human Evaluation

We also conduct human evaluation to assess the quality of summaries comprehensively. We consider the following six aspects for human evaluation: *Critical Analysis (CA)*, *Structural Rationality (SR)*, *Grammatical Fluency (GF)*, *Content Succinctness (CS)*, *Reference Coverage (RC)* and *Content Consistency (CC)*. The detailed definitions of the six aspects and the settings for human evaluation are introduced in Appendix D.

Table 2: Human evaluation of LLMs and other models.

Model	CA	SR	GF	CS	RC	CC
BART	2.233	2.567	2.6	4.133	-	-
EDITSum	2.8	2.7	2.667	4	-	-
GPT-3.5 (0-shot)	3.1	3.467	4.633	**4.333**	41.8%	3.433
GPT-3.5 (1-shot)	4.167	4.3	5.067	3.633	54.9%	3.567
GPT-4 (0-shot)	5.333	5.167	5.733	4.133	**59.3%**	**3.833**
GPT-4 (1-shot)	**5.6**	**5.633**	**5.767**	3.933	58.2%	3.567

The result of human evaluation is shown in Table 2. We conclude the following four observations: (1) LLMs outperform BART and EDITSum in aspects such as Critical Analysis, Structural Rationality, and Grammatical Fluency. This indicates that although LLMs perform worse than EDITSum in automatic evaluation, they are capable of generating better summaries in terms of human evaluation. (2) Among LLMs, GPT-4 performs better than GPT-3.5 in all aspects except for Critical Analysis, which is consistent with the result of automatic evaluation. Since GPT-3.5 (0-shot) tends to generate shorter summaries, they obtain higher scores in human evaluation. (3) For both GPT-3.5 and GPT-4, the performance of adding a demonstration (1-shot) is superior to that without a demonstration (0-shot), in several aspects. This demonstrates the importance of high-quality demonstrations for LLMs to understand task instructions and generate high-quality summaries. (4) For Reference Coverage, it can be observed that both GPT-3.5 and GPT-4 struggle to include all the provided reference papers in the summaries. The highest Reference Coverage is only 59.3%, indicating that 40.7% of the reference papers are still omitted. The result underscores the urgency to address this issue when utilizing LLMs to address MDSS task.

3.3 Deficiencies of Summaries Generated by LLMs

The above results show that LLMs tend to overlook some reference papers, resulting in low coverage of references in the generated summaries. Additionally, we also identify two other deficiencies of LLMs: **disorganized structure** and **high redundancy**. Disorganized structure refers to the structure of summaries generated by LLMs is unclear, with disorganized sub-topics, while high redundancy refers to the summaries generated by LLMs contain much redundant or repetitive content. We provide detailed analyses of the two deficiencies in Appendix E.

4 Method

In this section, we propose **I**terative **I**ntrospection based **R**efinement (**IIR**) method, which utilizes LLMs with prompt engineering to mitigate the above three deficiencies in LLM-generated summaries. IIR consists of four modules: *Key Aspects Extraction*, *Reference Paper Supplement*, *Structural Rationality Enhancement*, and *Content Succinctness Enhancement*. The framework of IIR is illustrated in Figure 2.

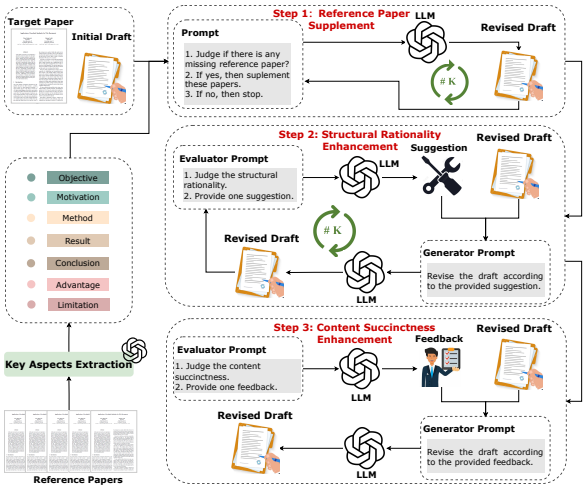

Figure 2: The framework of our IIR method.

4.1 Key Aspects Extraction

We use the LLM-generated summary from Section 3 as the draft for further refinement. Due to the context window limitation of LLMs, only the Abstract, Introduction, and Conclusion sections are used as input in Section 3, which may cause some key information missing when summarizing. To ensure the integrity of input information during refinement, we extract the key aspects of each paper as additional input, given the limited context window of LLMs.

To this end, we refer to the scientific concept classification scheme proposed by Teufel (2010) to classify aspects of scientific articles relevant to summarization tasks into the following seven categories: *Objective*, *Motivation*, *Method*, *Results*, *Conclusion*, *Advantage*, and *Limitation*. Then, we employ LLMs as a Key Aspects Extractor to extract or generate statements for each aspect from every input paper. The prompt used for the Key Aspects Extractor is shown in Appendix H.2.

4.2 Reference Paper Supplement

After Key Aspects Extraction, we utilize LLMs to add the missing reference papers to the summary. In the prompt setting for interacting with LLMs, the target paper, the reference papers, and the draft are provided in the form of key-value pairs in JSON format. We adopt a Chain-of-Thought (Wei et al., 2022) based prompting method, requiring LLMs to first count the number of the input reference

papers, then count those included in the draft, and compare the two to judge if they are equal. If not, the draft must be revised to include the missing reference papers. This process iterates until LLMs determine that no further modifications are needed. The prompt used for Reference Paper Supplement (Ref_Supple) is shown in Appendix H.3.

4.3 Structural Rationality Enhancement

After Ref_Supple, we take the draft obtained from it, along with key aspects of the target paper and reference papers, as input for Structural Rationality Enhancement (Struc_Enhance). We employ LLMs as an evaluator and a generator, respectively. The evaluator gives feedbacks and refinement suggestions on structural rationality of the draft, while the generator refines the draft based on the feedbacks and refinement suggestions.

In the preliminary experiment, we empirically observe that, when providing general and vague revising feedback, the generator tends to make extensive revisions to the draft, which causes two problems: First, it is difficult to track the modification trajectory of LLMs and difficult to evaluate the effectiveness of the modifications; Second, LLMs are prone to omitting some reference papers again when revising the draft, rendering the Ref_Supple step ineffective.

To address the above two problems, we design a fine-grained and controllable prompt method equipped with Chain-of-Thought and fine-grained operators for the evaluator and generator. Specifically, we refer to the operations commonly used in text editing systems (Reid and Neubig, 2022; Liu et al., 2023b), and predefine five types of possible refinement operations: *Modify*, *Delete*, *Insert*, *Move* and *Merge*. Details about these operations are listed in Table 6 of Appendix F. The five types of operations are applied at the sentence level and each draft sentence is labeled with a unique identifier "<*SENTENCE_?*>". This setting guarantees the generated feedbacks and suggestions are specific and easily traceable.

When prompting LLMs as the evaluator, we require LLMs to identify all sentences of the draft into different sub-topics, and then determine whether the division of these sub-topics is appropriate or whether they can be merged. This process helps identify structural irrationalities in the current draft and provides corresponding suggestions. The suggestions should be from the predefined operations of Table 6. The prompt for the evaluator is shown in Appendix H.4.

When prompting LLMs as the generator, we require LLMs to revise the draft strictly in accordance with the suggestions from the evaluator. The prompt for the generator is also shown in Appendix H.4. Finally, to prevent conflicts of sentence identifiers after different operations, the evaluator is required to give only one suggestion at a time, ensuring that there are no conflicts between suggestions. The evaluation-generation process then proceeds iteratively to continuously improve the structural rationality of the draft. The complete process is shown in algorithm 1 of Appendix.

4.4 Content Succinctness Enhancement

After Struc_Enhance, we further take the draft from it as input for Content Succinctness Enhancement (Cont_Enhance). We employ LLMs as a content succinctness evaluator and a content succinctness generator. The evaluator needs to inspect and provide feedbacks on the corresponding three aspects of high redundancy illustrated in Appendix E.2. We also predefine three types of text editing operations: *Modify*, *Delete*, and *Merge*. Details of these operations are listed in Table 7 of Appendix F. Since the operations in this step are simpler than those required for Cont_Enhance, no iteration is required for this step. The revision of the draft is completed in only one evaluation-generation process. The prompts for the content succinctness evaluator and generator are shown in Appendix H.5.

5 Experiments

In this section, we conduct experiments to validate the effectiveness of our proposed IIR method.

5.1 Experimental Setup

Metrics We employ the same automatic and human evaluation as in Section 3.1.

Compared Prompting Method To show the superiority of our IIR method, we compare it with other LLM prompting methods. Specifically, we introduce a new direct prompting method called **Single-Turn Prompt (SinTurn)**. SinTurn also utilizes LLMs as both the evaluator and the generator. However, it differs in that, the evaluator of SinTurn directly evaluates the six aspects of related work: *Critical Analysis, Structural Rationality, Grammatical Fluency, Content Succinctness, Reference Coverage*, and *Content Consistency*, and then it directly provide feedbacks and suggestions without predefined operations. Subsequently, the generator revises the draft based on the feedbacks and suggestions from the evaluator.

Table 3: Automatic evaluation results of our IIR method and the compared method. Structural Rationality Enhancement (Struc_Enhance) includes three iterations (#1, #2, #3).

Summary Type	R-1 (%)	R-2 (%)	R-L (%)	BS (%)	G-Eval
Initial Draft	47.38	11.61	44.75	86.08	4.10
SinTurn	45.05	10.59	42.46	85.72	4.01
IIR					
After Ref_Supple	47.78 (↑ 0.40)	12.73 (↑ 1.12)	44.7 (↓ 0.05)	**86.24** (↑ 0.16)	4.13 (↑ 0.03)
After Struc_Enhance (#1)	48.2 (↑ 0.42)	**12.83** (↑ 0.10)	44.87 (↑ 0.17)	86.21 (↓ 0.03)	4.16 (↑ 0.03)
After Struc_Enhance (#2)	48.39 (↑ 0.19)	12.8 (↓ 0.03)	45 (↑ 0.13)	86.15 (↓ 0.06)	4.15 (↓ 0.01)
After Struc_Enhance (#3)	48.43 (↑ 0.04)	**12.83** (↑ 0.03)	45.12 (↑ 0.12)	86.14 (↓ 0.01)	**4.18** (↑ 0.03)
After Cont_Enhance	**48.67** (↑ 0.24)	12.76 (↓ 0.07)	**45.29** (↑ 0.17)	86.17 (↑ 0.03)	4.16 (↓ 0.02)

More experimental details are introduced in Appendix G.

5.2 Experimental Results

5.2.1 Automatic Evaluation

In automatic evaluation, we report the progressive performance of each step of IIR: Reference Paper Supplement (**Ref_Supple**), Structural Rationality Enhancement (**Struc_Enhance**), and Content Succinctness Enhancement (**Cont_Enhance**). The result is shown in Table 3. We have the following two observations.

(1) The compared SinTurn method fails to improve the performance of the drafts, with notable decreases across various metrics. This indicates that it is challenging for LLMs to simultaneously enhance multiple aspects that affect summary quality. Additionally, without predefined operations, the evaluator can only provide general and vague suggestions, which leads to extensive revisions and causes the quality of the revised draft drop significantly. Conversely, our IIR method addresses the three main deficiencies of LLMs through iterative introspection based refinement with predefined operations, therefore bringing substantial improvements on summary performance.

(2) Each module of IIR can enhance the performance of summary on most metrics. After Ref_Supple, the summary achieves obvious improvements in ROUGE-1/2. This is because this module supplements the missing reference papers in the summary, thus increasing the informativeness of the summary. After Struc_Enhance, the summary shows improvements over the Ref_Supple module in all metrics except for BERTScore metric. When looking at each step of this module (#1, #2, #3), since the generator performs only one operation each time, the performance change before and after each iteration is minimal. After Cont_Enhance, a large increase in ROUGE-1/L can be observed compared with Struc_Enhance. Finally, comparing the final refined summary to the initial draft, we can find the summary performance increases by 1.29%, 1.15%, 0.54%, 0.09%, and 0.06% on the five metrics, respectively. The result demonstrates the effectiveness of our IIR method in improving the quality of summaries generated by LLMs.

Table 4: Human evaluation results of different prompt methods on ComRW dataset.

Summary Type	CA	SR	GF	CS	RC	CC
Initial Draft	2.067	1.967	2.533	1.633	58.2%	**2.667**
SinTurn	**2.5**	2.467	**2.6**	2.133	57.42%	**2.667**
IIR	2.267	**2.7**	**2.6**	**2.733**	**89.14%**	2.6

5.2.2 Human Evaluation

We further conduct human evaluation to analyze the impact of IIR on summary quality in a more specific and comprehensive way.

Overall Performance The first human evaluation compares our IIR method against SinTurn and the initial draft. The evaluation settings are generally the same as those of Appendix D, but differ in that the ranking score is from 3 (best) to 1 (worst).

The result is shown in Table 4. We draw three conclusions from it: (1) Comparing the initial draft with IIR, we find that IIR brings obvious improvements on Reference Coverage (RC), Structural Rationality (SR), and Content Succinctness (CS), which demonstrates the effectiveness of our method in addressing the deficiencies in summaries generated by LLMs. (2) Comparing IIR with SinTurn, it is evident that IIR can help achieve higher human scores in multiple aspects, indicating that our iterative introspection based refinement method is more conducive to improving summary performance than the single-turn prompting method. (3) It is worth noting that although SinTurn requires LLMs to improve Reference Coverage (RC) of the draft, the RC result is only 57.42%, which is even worse than the initial draft's 58.2%. This indicates that LLMs still struggle to understand complex instructions on multi-dimensional summary evaluation. Therefore, decomposing complex instructions into simple and specific instructions is an effective strategy to harness the power of LLMs.

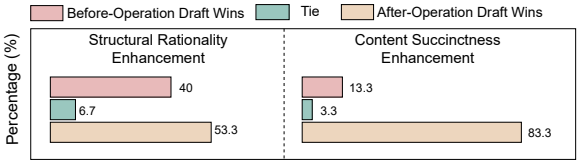

Figure 3: Human evaluation of Structural Rationality Enhancement and Content Succinctness Enhancement.

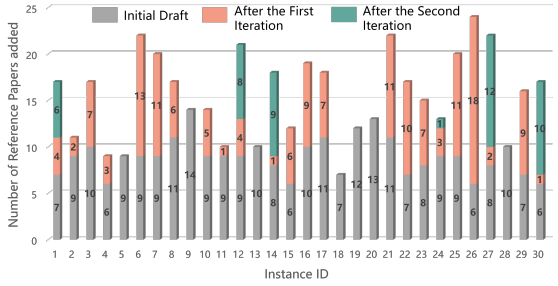

Figure 4: Statistical results of the number of modification iterations and reference papers added in each iteration for each instance.

Module Performance We conduct another human evaluation to analyze the effectiveness of the modules of our IIR method. We set up two sets of pairwise comparisons on Structural Rationality Enhancement (**Struc_Enhance**) and Content Succinctness Enhancement (**Cont_Enhance**). We randomly select 10 summaries and invite three assessors with expertise in natural language processing. Take Struc_Enhance as an example, the assessors are asked to compare the two drafts, before and after operation, to determine which one is better, or choose a tie. Since the effectiveness of Reference Paper Supplement module has already been demonstrated before, it will not be repeated here.

The result is shown in Figure 3. We observe that the assessors have clear preferences for after-operation draft on both Struc_Enhance and Cont_Enhance. Specifically, regarding Struc_Enhance, after-operation draft obtains an average of 53.5% preference, whereas the average preference of before-operation draft is 40%. Similarly, for Cont_Enhance, the average preference of after-operation draft is 83.3%, notably higher than the 13.3% preference for before-operation draft. The above results indicate the effectiveness of our IIR method in handling deficiencies in structural rationality and content succinctness.

5.3 More Analyses on IIR

5.3.1 Analysis of Reference Paper Supplement

We first count the number of modification iterations and the number of reference papers added in each iteration for each instance, as shown in Figure 4.

We can find that, the average number of iterations for Ref_Supple is 1.27. Most instances require only one iteration of revision, with the first iteration introducing an average of 6.57 reference papers. Only six instances require a second iteration of revision, which generally occurs when the first iteration is unsatisfactory, and the second iteration introduces an average of 7.67 reference papers.

When analyzing the draft after Ref_Supple, we find LLMs merely list the supplementary reference papers at the end of the draft, which makes the structural rationality problem of the summary more prominent. Therefore, it becomes more crucial to take steps to enhance structural rationality.

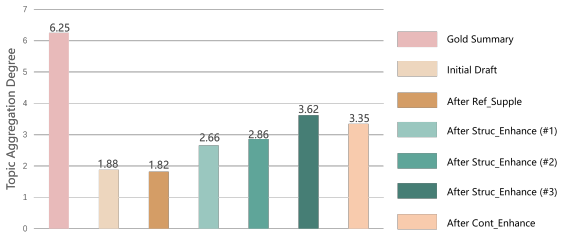

Figure 5: TA results of summaries generated at different steps of IIR.

5.3.2 Analysis of Structural Rationality Enhancement

Statistical Result of TA We first define the concept of Topic Aggregation Degree (TA) to quantitatively analyze structural rationality of summaries. TA is introduced detailedly in Appendix E.1. We count TA of summaries generated at different steps of IIR and the result is shown in Figure 5.

We can find that after Struc_Enhance, TA increases from 1.88 of the initial draft to 3.62. Each iteration of Struc_Enhance contributes to this improvement, with scores rising from 2.66 to 2.86, and finally to 3.62. These results indicate that our Struc_Enhance module can effectively enhance the structural rationality of summaries.

Predefined Operation Analysis We predefine five types of operations: *Modify*, *Delete*, *Insert*, *Move* and *Merge*, in Struc_Enhance. We now count the proportions of the five operations to clarify the modification strategy used by LLMs.

The result is shown in Figure 6 (a). We can find that *Merge* operation accounts for the highest proportion at 59.62%, indicating that the primary operation taken by LLMs to improve structural rationality is merging dispersed sub-topics. The next most common operation is *Insert*, accounting for 32.69%, which is also a necessary action to make the contextual transition of the summary more coherent. The remaining three operations,

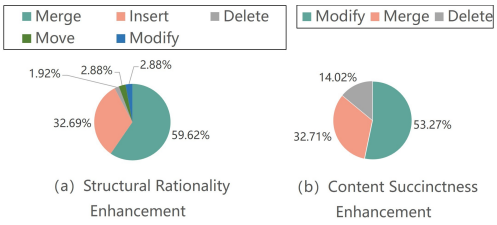

Figure 6: The proportion of predefined operations used by LLMs.

Delete, *Move*, and *Modify*, have lower proportions, suggesting that LLMs prioritize topic-level operations over sentence-level operations when enhancing structural rationality.

5.3.3 Analysis of Content Succinctness Enhancement

We also predefine three types of operations: *Modify, Merge*, and *Delete* in Cont_Enhance. We analyze the proportions of the three operations to clarify the modification strategy used by LLMs. The result is shown in Figure 6 (b). We can find that *Modify* operation accounts for the highest proportion at 53.27%, primarily involving modifications to make sentences more concise. Besides, *Merge* operation accounts for 32.71%, which is used to merge different sentences to remove redundant information. Finally, *Delete* operation is also widely used, accounting for 14.02%, which deletes the whole redundant sentence.

6 Related Work

6.1 Multi-Document Scientific Summarization

Multi-Document Scientific Summarization (MDSS) involves consolidating scattered information from multiple papers. Previous studies can be categorized into extractive, abstractive and LLM-based methods. Extractive methods are commonly used in the early stages, which select off-the-shelf sentences to form the summary (Hoang and Kan, 2010; Hu and Wan, 2014; Wang et al., 2018). With the advancement of deep neural networks, abstractive methods have rapidly become the dominant approach to MDSS (Chen et al., 2021, 2022; Wang et al., 2022; Moro et al., 2022; Wang et al., 2023a), which generate summaries from scratch, bringing better coherence and readability. Despite their advantages, current task setting and constructed datasets (Lu et al., 2020; Chen et al., 2022) lead to a significant gap between existing research on MDSS and practical applications. Recently, LLMs have brought new solutions to MDSS by leveraging the powerful zero-shot learning and in-context learning (Brown et al., 2020) ability. These LLM-based methods (Haman and Školník, 2023; Huang and Tan, 2023; Agarwal et al., 2024; Martin-Boyle et al., 2024) can tackle MDSS task via flexible instructions without the need for large amounts of data. However, these methods fail to provide a systematic and comprehensive evaluation of the performance of LLMs on MDSS, resulting in the shortcomings of LLMs in addressing MDSS remaining unknown, which is the objective of this paper.

6.2 Prompting Methods based Text Generation

LLMs exhibit a new ability of learning merely from a few demonstrations in the context, called In-Context Learning (ICL) (Brown et al., 2020; Dong et al., 2022), which brings a novel task-solving paradigm for text generation from the perspective of prompting methods. Recently, a plenty of prompting methods have been proposed to unleash more capabilities of LLMs via Chain-of-Thought (Radhakrishnan et al., 2023; Zhang et al., 2023a; Wang et al., 2023b), content plan (Narayan et al., 2021; Creo et al., 2023; You et al., 2023), iterative refinement (Zeng et al., 2023; Zhang et al., 2023b; Madaan et al., 2024), and problem decomposition (Sun et al., 2023; Khot et al., 2022). Our work differs from these prompting methods by designing prompts with Chain-of-Thought and fine-grained sentence-level operators, which ensures the modifications made by LLMs are specific, controllable and traceable, thereby contributing to a better solution for MDSS task.

7 Conclusion

In this paper, we redefine MDSS task from the perspective of practical applications, and construct a new dataset ComRW. Then, we conduct a comprehensive evaluation of the performance of LLMs on this newly defined task, and find that the summaries generated by LLMs suffer from three major deficiencies: low coverage of reference papers, disorganized structure, and high redundancy. To mitigate these deficiencies, we propose an Iterative Introspection based Refinement (IIR) method, which uses prompts equipped with Chain-of-Thought and fine-grained operators to treat LLMs as evaluators and generators to improve summary quality, respectively. Both automatic and human evaluations demonstrate that the proposed IIR method effectively alleviates these issues, resulting in higher-quality summaries. Our IIR method also provides inspiration for utilizing LLMs to tackle MDSS task effectively with prompting methods.

Limitations

The limitations of this paper are twofold: (1) The constructed dataset ComRW has only 30 instances, which cannot support more explorations of LLMs based MDSS from the perspective of practical applications, such as instruction tuning based methods or parameter-efficient fine-tuning. (2) Our proposed IIR method is somewhat complex and inflexible, involving separated evaluation and regeneration steps to handle different deficiencies of summaries generated by LLMs, which requires great effort in task decomposition and prompt designing. Therefore, more flexible and efficient prompting methods deserve exploration in the future.

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*.

Shubham Agarwal, Issam H Laradji, Laurent Charlin, and Christopher Pal. 2024. Litllm: A toolkit for scientific literature review. *arXiv preprint arXiv:2402.01788*.

Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. Parallel iterative edit models for local sequence transduction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4260–4270.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

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.

Xiuying Chen, Hind Alamro, Mingzhe Li, Shen Gao, Xiangliang Zhang, Dongyan Zhao, and Rui Yan. 2021. Capturing relations between scientific papers: An abstractive model for related work section generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6068–6077.

Cheng-Han Chiang and Hung-Yi Lee. 2023. Can large language models be an alternative to human evaluations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631.

Aldan Creo, Manuel Lama, and Juan C Vidal. 2023. Prompting llms with content plans to enhance the summarization of scientific articles. *arXiv preprint arXiv:2312.08282*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166*.

Michael Haman and Milan Školník. 2023. Using chatgpt to conduct a literature review. *Accountability in research*, pages 1–3.

Cong Duy Vu Hoang and Min-Yen Kan. 2010. Towards automated related work summarization. In *Coling 2010: Posters*, pages 427–435.

Yue Hu and Xiaojun Wan. 2014. Automatic generation of related work sections in scientific papers: an optimization approach. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1624–1633.

Jingshan Huang and Ming Tan. 2023. The role of chatgpt in scientific communication: writing better scientific review articles. *American journal of cancer research*, 13(4):1148.

Hayate Iso, Chao Qiao, and Hang Li. 2020. Fact-based text editing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 171–182.

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

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2022. Decomposed prompting: A modular approach for solving complex tasks. In *The Eleventh International Conference on Learning Representations*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Yutong Li, Lu Chen, Aiwei Liu, Kai Yu, and Lijie Wen. 2024. Chatcite: Llm agent with human workflow guidance for comparative literature summary. *arXiv preprint arXiv:2403.02574*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023a. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522.

Yixin Liu, Budhaditya Deb, Milagro Teruel, Aaron Halfaker, Dragomir Radev, and Ahmed Hassan. 2023b. On improving summarization factual consistency from natural language feedback. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15144–15161.

Yao Lu, Yue Dong, and Laurent Charlin. 2020. Multi-xscience: A large-scale dataset for extreme multi-document summarization of scientific articles. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8068–8074.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2024. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36.

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*.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919.

Gianluca Moro, Luca Ragazzi, Lorenzo Valgimigli, and Davide Freddi. 2022. Discriminative marginalized probabilistic neural method for multi-document summarization of medical literature. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 180–189.

Shashi Narayan, Yao Zhao, Joshua Maynez, Gonçalo Simões, Vitaly Nikolaev, and Ryan McDonald. 2021. Planning with learned entity prompts for abstractive summarization. *Transactions of the Association for Computational Linguistics*, 9:1475–1492.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Ansh Radhakrishnan, Karina Nguyen, Anna Chen, Carol Chen, Carson Denison, Danny Hernandez, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, et al. 2023. Question decomposition improves the faithfulness of model-generated reasoning. *arXiv preprint arXiv:2307.11768*.

Machel Reid and Graham Neubig. 2022. Learning to model editing processes. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3822–3832.

Timo Schick, A Yu Jane, Zhengbao Jiang, Fabio Petroni, Patrick Lewis, Gautier Izacard, Qingfei You, Christoforos Nalmpantis, Edouard Grave, and Sebastian Riedel. 2022. Peer: A collaborative language model. In *The Eleventh International Conference on Learning Representations*.

Simeng Sun, Yang Liu, Shuohang Wang, Chenguang Zhu, and Mohit Iyyer. 2023. Pearl: Prompting large language models to plan and execute actions over long documents. *arXiv preprint arXiv:2305.14564*.

Simone Teufel. 2010. The structure of scientific articles: Applications to summarisation and citation indexing.

Pancheng Wang, Shasha Li, Shenling Liu, Jintao Tang, and Ting Wang. 2023a. Plan and generate: Explicit and implicit variational augmentation for multi-document summarization of scientific articles. *Information Processing & Management*, 60(4):103409.

Pancheng Wang, Shasha Li, Kunyuan Pang, Liangliang He, Dong Li, Jintao Tang, and Ting Wang. 2022. Multi-document scientific summarization from a knowledge graph-centric view. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6222–6233.

Yiming Wang, Zhuosheng Zhang, and Rui Wang. 2023b. Element-aware summarization with large language models: Expert-aligned evaluation and chain-of-thought method. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8640–8665.

Yongzhen Wang, Xiaozhong Liu, and Zheng Gao. 2018. Neural related work summarization with a joint context-driven attention mechanism. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1776–1786.

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

Wang You, Wenshan Wu, Yaobo Liang, Shaoguang Mao, Chenfei Wu, Maosong Cao, Yuzhe Cai, Yiduo Guo, Yan Xia, Furu Wei, et al. 2023. Eipe-text: Evaluation-guided iterative plan extraction for long-form narrative text generation. *arXiv preprint arXiv:2310.08185*.

Qi Zeng, Mankeerat Sidhu, Hou Pong Chan, Lu Wang, and Heng Ji. 2023. Meta-review generation with checklist-guided iterative introspection. *arXiv preprint arXiv:2305.14647*.

Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2023a. Extractive summarization via chatgpt for faithful summary generation. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2023b. Summit: Iterative text summarization via chatgpt. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10644–10657.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Li Zhao, Minlie Huang, Haiqiang Chen, Junjun Cheng, and Xiaoyan Zhu. 2014. Clustering aspect-related phrases by leveraging sentiment distribution consistency. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1614–1623.

A Dataset Construction and Analysis

According to the new task definition in Section 2, we first construct a new dataset ComRW. The construction process of ComRW is introduced below.

A.1 Dataset Construction

Target Papers Selection We first select target papers from top conferences of natural language processing, such as ACL, EMNLP, and NAACL. Papers from these top conferences adhere to academic writing conventions and provide thorough reviews of references, thus having high-quality related work sections. We manually select 30 papers as target papers. Their related work sections exhibit clear structure, moderate length, and appropriate number of references, rendering them suitable as gold summaries for our task.

Reference Papers Collection Then we identify all the references from the related work section and automatically download them using Google Scholar[3]. For references that cannot be downloaded automatically, we manually retrieve them using school library resources. This ensures that no reference paper is missed.

Content Extraction After gathering all the target papers and reference papers, we utilize PDFMINER[4] to convert all downloaded papers from PDF to TXT format. We also develop a section extraction tool to automatically extract contents of different sections and save them in JSON files.

A.2 Dataset Analysis

Statistical Analysis The constructed dataset ComRW contains 30 instances and the statistical information of ComRW is shown in Table 5. On average, each instance includes 15.8 reference papers. The input document contains an average of 67,934.16 words, while the gold summary has an average of 491.1 words. Although ComRW has only 30 instances, the strong few-shot learning and in-context learning capabilities of LLMs enable the dataset to support a reasonable assessment of LLMs' performance on MDSS task.

Compared with previous MDSS datasets like Multi-Xscience (Lu et al., 2020), TAD (Chen et al., 2022) and TAS2 (Chen et al., 2022), ComRW significantly surpasses them in terms of the average number of reference papers, input words, and summary words. Furthermore, an analysis of the proportion of novel n-grams in the gold summary that do not appear in the input documents indicates that ComRW, by using the full text of papers as input, can greatly reduce the proportion of new unigrams and bigrams in the summary, thereby avoiding the problem of intrinsic hallucination. Thus, our dataset enables a more objective assessment of model performance.

More Analyses on ComRW Figure 7 illustrates the distribution of the number of reference papers and sub-topics in each instance for ComRW dataset. It can be observed that the number of reference papers is roughly distributed evenly between 9 and 21. Moreover, each instance in ComRW dataset contains 1 to 5 sub-topics, with an average of 2.53 sub-topics. Particularly, instances containing 2 sub-topics are the most common, with 13 intances, followed by 10 instances containing 3 sub-topics. How to effectively identify and organize reference papers according to different sub-topics will be a significant challenge to MDSS models.

[3] https://scholar.google.com/

[4] https://pypi.org/project/pdfminer/

Table 5: Statistical information of ComRW and other MDSS datasets.

Dataset	# Test Set	# Input Words	# Summary Words	# Reference Papers	Novel Unigrams	Novel Bigrams
Multi-Xscience	5,093	778.08	116.44	4.42	42.33%	81.75%
TAD	5,000	845	191	5.17	43.58%	83.29%
TAS2	5,000	788	126	4.8	42.62%	82.03%
ComRW	30	67,934.16	491.1	15.8	5.95%	37.44%

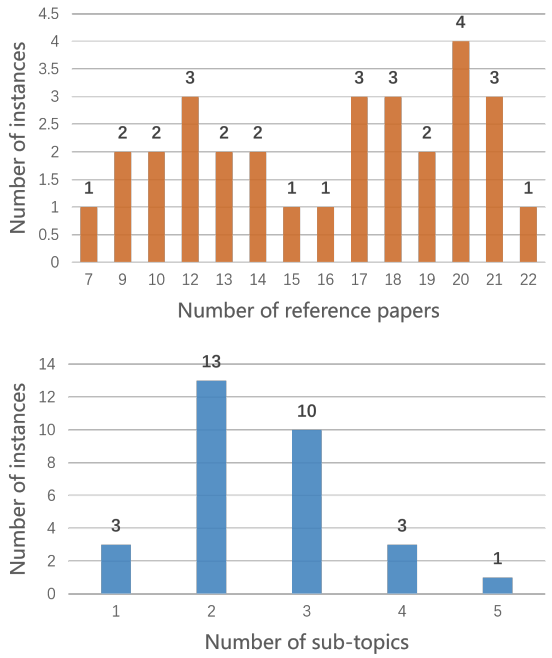

Figure 7: Distribution of the number of reference papers and the number of sub-topics for ComRW.

B Prompt Design for LLMs

We use zero-shot prompting (0-shot) and one-shot prompting (1-shot) to interact with LLMs. Given the limited context window of `gpt-3.5-turbo-0125` with only 16,385 tokens, we take the Abstract, Introduction, and Conclusion section of each paper as input. Despite the context window of `gpt-4-0125-preview` extends up to 128,000 tokens, we maintain input consistency with `gpt-3.5-turbo-0125` by using the same input. The prompt template for zero-shot prompting is shown in Figure 10, and the prompt template for one-shot prompting adds a specially selected demonstration based on zero-shot prompting.

C Compared Models Setting

Since our ComRW dataset contains only 30 instances, it lacks sufficient data for training BART and EDITSum from scratch. To address this, we consider an alternative method to generate summaries by BART and EDITSum. Considering that current large-scale MDSS datasets, such as Multi-Xscience, are constructed at the paragraph level, we first segment the ComRW dataset into individual paragraphs and identify reference papers of each paragraph. The modified dataset is denoted as ComRW-Para. Then, we train BART and EDITSum on Multi-Xscience training dataset and choose the best-performing models according to their performance on Multi-Xscience validation dataset. Subsequently, we apply the trained models to generate summaries on ComRW-Para. The generated summaries are then organized in order to serve as section-level predictions for BART and EDITSum on ComRW.

D Human Evaluation Settings

We refer to the human evaluation settings from Li et al. (2024), and take into account the definitions, content and structure requirements of a well-written related work, and then set the following six aspects for human evaluation:

- **Critical Analysis (CA)**: Whether the generated summary include proper analysis of the strengths and weaknesses of reference papers.
- **Structural Rationality (SR)**: Whether the summary is organized by sub-topics in a coherent and structured manner, rather than simply listing different reference papers.
- **Grammatical Fluency (GF)**: Whether the summary is fluent, with no obvious grammatical errors.
- **Content Succinctness (CS)**: Whether the summary is concise, does not contain repetition or lengthy information, or information that is irrelevant to the topics discussed in the target paper.
- **Reference Coverage (RC)**: Does the summary include all the provided reference papers without any omissions.
- **Content Consistency (CC)**: Whether the content of the summary is consistent with the input target paper and reference papers.

For *Reference Coverage*, the result can be cal-

culated automatically, thus requiring no human involvement. For *Reference Coverage* and *Content Consistency*, we only conduct evaluation on these two aspects for summaries generated by LLMs, because BART and EDITSum are trained on Multi-Xscience, and during training, all citation markers are normalized, rendering human evaluation infeasible for these two aspects. Regarding *Content Consistency*, we ask the evaluators to rank GPT-3.5 (0-shot), GPT-3.5 (1-shot), GPT-4 (0-shot), and GPT-4 (1-shot) from 1 (best) to 4 (worst). Models ranked 1, 2, 3, and 4 receive scores of 4, 3, 2, and 1 respectively. If the evaluators consider that different summaries have the same quality, they can assign them the same rank. For instance, if the rankings are 1, 2, 2, and 4, then scores are 4, 3, 3, and 1 respectively. For aspects other than *Reference Coverage* and *Content Consistency*, we ask the evaluators to rank all the six models from 1 (best) to 6 (worst) with scores ranging from 6 to 1 accordingly.

We ramdomly sample 10 instances from ComRW dataset for human evaluation and invite three graduate students majoring in natural language processing to conduct human evaluation. The final score is the average score of the three evaluators.

E Deficiencies of Summaries Generated by LLMs

In this section, we provide detailed analysis on the disorganized structure and high redundancy deficiencies of LLMs.

E.1 Disorganized Structure

To qualitatively and quantitatively analyze the Structural Rationality of summaries, we first define the concept of **Topic Aggregation Degree (TA)** $\mathcal{T}$ as follows:

$$\mathcal{T} = \frac{1}{|S|} \sum_{i=1}^{|S|} n_i / t_i \quad (1)$$

where S means the summary set, n_i and t_i denote the number of reference papers and sub-topics in the i-th generated summary, respectively.

Intuitively, TA measures the average number of reference papers contained within each sub-topic in the summary. This reflects the ability of a summarization model to organize reference papers into different sub-topics, where the higher the value, the stronger the ability. To count the number of sub-topics, we use one-shot prompting to employ GPT-4 as the sub-topic extractor to automatically identify different sub-topics in the summary. Prompt of the sub-topic extractor is shown in Appendix H.1. Through preliminary experiments, we find that GPT-4 can effectively identify different sub-topic groups in the summary, making it a reliable sub-topic extractor.

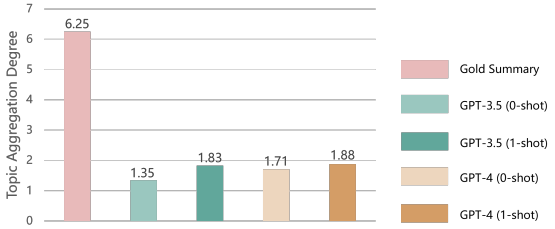

Figure 8: Statistical result of topic aggregation degree of different LLMs.

Then we use the sub-topic extractor to count TA of different LLMs and the gold summary, and show the result in Figure 8. Notably, the average TA of the gold summary is 6.25, indicating that the reference papers are effectively organized and summarized into different sub-topics, which is a necessary attribute for a well-written related work. In contrast, the average TA of the summaries generated by GPT-3.5 and GPT-4 is only 1.35∼1.88. This suggests that most sub-topics are supported by only one or two reference papers, or in some cases, no sub-topics at all, resulting in a simple enumeration of reference papers.

To illustrate this, we present an example of the summary generated by GPT-4 (0-shot) in Figure 9. For the convenience of showing the text fragments belonging to different reference papers, the summary in Figure 9 is divided into sentences and displayed in JSON format, where "<*SENTENCE_?*>" represents the sentence identifier, and citation markers are highlighted in green shading. From the figure, we can see that the summary generated by GPT-4 simply introduces the reference papers in the order of input, without summarizing a clear topic structure. In fact, the three reference papers "@*cite_10*", "@*cite_11*", and "@*cite_12*" in sentence "<*SENTENCE_19*>" belong to the category of "*sentiment analysis*", which can be described together with the reference paper "@*cite_1*" in sentence "<*SENTENCE_4*>". This indicates that existing LLMs, even the most powerful ones like GPT-4, have obvious shortcomings in organizing sub-topics in MDSS task.

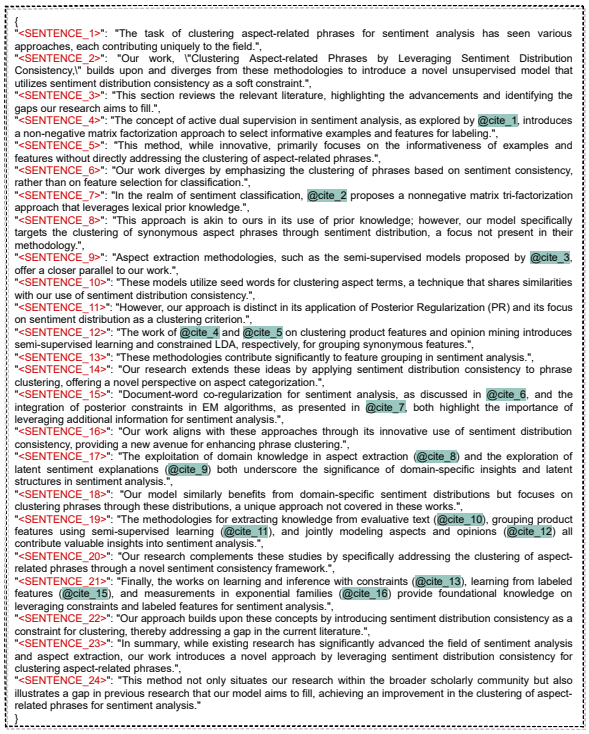

Figure 9: An example of the summary generated by LLMs (The target paper is from Zhao et al. (2014)).

E.2 High Redundancy

The summary generated by LLMs also exhibits high redundancy, manifested in the following three aspects: (1) **Repetition of introducing own work**. Taking the summary in Figure 9 as an example, in "<*SENTENCE_2*>", "<*SENTENCE_22*>", and "<*SENTENCE_23*>", the contribution of the target paper is redundantly expressed as "*leveraging sentiment distribution consistency for clustering aspect-related phrases*". (2) **Generation of unnecessary title information**, as shown in "<*SENTENCE_2*>" in Figure 9. (3) **Including too many aspects of a single reference paper**. For instance, in "<*SENTENCE_4*>" and "<*SENTENCE_5*>", the description of"@*cite_1*" includes too much information about its objective, method, advantages, and limitations, without selecting the most relevant aspects related to the target paper.

F Predefined Text Editing Operations

The five types of predefined text editing operations used in Structureal Rationality Enhancement is shown in Table 6. And the five types of predefined text editing operations used in Content Succinctness Enhancement is shown in Table 7.

Algorithm 1 Structural Rationality Enhancement based on Iterative Introspection of LLMs

Input: Target Paper $\mathcal{T}$, Reference Papers $\mathcal{D}$, Draft from last step $\mathcal{S}_0$, Evaluator $E(\cdot)$, Generator $G(\cdot)$, Predefined Operations $\mathcal{C} = \{Modify, Delete, Insert, Move, Merge\}$

Output: Draft after n steps of Structural Rationality Enhancement $\mathcal{S}_n$

1: **for** $i = 1$ to n **do**
2: Obtain feedbacks and suggestions $g \leftarrow E(\mathcal{T}, \mathcal{D}, \mathcal{S}_{i-1})$, where $g \in \mathcal{C}$
3: Refined draft $\mathcal{S}_i \leftarrow G(\mathcal{T}, \mathcal{D}, \mathcal{S}_{i-1}, g)$
4: **end for**

G Experimental Details of IIR

The experiments of IIR are also conducted on the ComRW dataset. We use the summaries generated by GPT-4 (1-shot) of Section 3 as the initial draft, because of its superior performance among all LLM settings, indicating that GPT-4 itself possesses stronger summarization capability. However, due to its shortcomings such as low coverage of reference papers, disorganized structure, and high redundancy, its application on MDSS still has limitations. Therefore, the improvements made on GPT-4 will be more meaningful for MDSS task. Additionally, in order to save costs and considering the less stringent requirements on the capability of LLMs in the key aspect extraction phase of Section 4.1, we employ GPT-3.5 (gpt-3.5-turbo-0125) as the extractor in this phase. For the evaluators and generators in Section 4.2 to Section 4.4, we use GPT-4 (gpt-4-0125-preview) for these three phases. Additionally, we set the number of iteration steps n for Structural Rationality Enhancement to 3 based on preliminary experiment.

H Prompt Templates

In this section, we list the prompt templates used throughout this paper.

H.1 Prompt for Sub-topic Extractor

The prompt for our sub-topic extractor is shown in Figure 11 and Figure 12.

H.2 Prompt for Key Aspects Extractor

The prompt for our Key Aspects Extractor is shown in Figure 13.

Table 6: Predefined text editing operations for Structural Rationality Enhancement

Operation Type	Instruction Template
Modify	"Modify the sentence <SENTENCE_?> to include information ___"
Delete	"Delete the sentence <SENTENCE_?>"
Insert	"Insert a new sentence about ___ between the position of sentence <SENTENCE_n> and <SENTENCE_m>"
Move	"Move sentence <SENTENCE_?> before sentence <SENTENCE_n>, then slightly Modify sentence <SENTENCE_?> and <SENTENCE_n> to make them contextual coherent"
Merge	"Merge different sub-themes ___, ___, ... ___ into a unified theme ___ by putting their sentences together, then slightly revise the sentences of the theme ___ to make them contexutal coherent and reduce fragmentation"

Table 7: Predefined text editing operations for Content Succinctness Enhancement

Operation Type	Instruction Template
Modify	"Modify the sentence <SENTENCE_?> to exclude information about ___"
Delete	"Delete the sentence <SENTENCE_?>"
Merge	"Merge different sentences <SENTENCE_?>,...,<SENTENCE_?> into a single sentence <SENTENCE_?> to make them more concise."

H.3 Prompt for Reference Paper Supplement

The prompt for Reference Paper Supplement is shown in Figure 14.

H.4 Prompt for Structural Rationality Enhancement

The prompt for structural rationality evaluator is shown in Figure 15 and Figure 16. The prompt for structural rationality generator is shown in Figure 17.

H.5 Prompt for Content Succinctness Enhancement

The prompt for content succinctness evaluator is shown in Figure 18. And the prompt for content succinctness generator is shown in Figure 19 and Figure 20.

I Case Study

We provide a case study to clearly demonstrate the effects of the three steps of IIR in improving the summary quality. Figure 21, Figure 22, Figure 23, and Figure 24 correspond to the initial draft, the summary after Reference Paper Supplement, the summary after Structural Rationality Enhancement, and the summary after Content Succinctness Enhancement, respectively. We also summarize the modifications made by the three steps of IIR in Table 8.

Comparing Figure 21 and Figure 22, we can see that Reference Paper Supplement step can effectively identify the missing reference papers in the initial draft and add them into the summary. Comparing Figure 22 and Figure 23, we can see that the draft after Reference Paper Supplement merely lists the reference papers in the summary with an incoherent context and dispersed sub-topics. For this reason, our Structural Rationality Enhancement step inserts transitional sentences between different sub-topics to make the transition smoother and merges different sub-topics effectively to enhance the inherent cohesion and organizational coherence of the summary. Comparing Figure 23 and Figure 24, it can be found that our Content Succinctness Enhancement step can effectively eliminate redundant information and irrelevant content from the summary, thereby enhancing the conciseness of the generated summary.

Table 8: Modifications of different steps of IIR.

Step	Modification
Reference Paper Supplement	❶ Insert a new sentence <SENTENCE_17>, describing reference paper @cite_5 ❷ Insert a new sentence <SENTENCE_18>, describing reference paper @cite_8 ❸ Insert a new sentence <SENTENCE_19>, describing reference paper @cite_9
Structural Rationality Enhancement	❶ Insert a new sentence about transition from traditional methods to neural network based methods before sentence <SENTENCE_9> ❷ Modify sentence <SENTENCE_9> to make contextual conherence ❸ Merge different sub-topics of <SENTENCE_10>...<SENTENCE_19> into a unified sub-topic "neural network based method"
Content Succinctness Enhancement	❶ Delete the title information of sentence <SENTENCE_2> ❷ Delete sentence <SENTENCE_6> ❸ Merge different sentences: <SENTENCE_7> and <SENTENCE_8>, and simplify the description of @cite_1 ❹ Delete sentences <SENTENCE_20> and <SENTENCE_21>

Imagine you are a scientific researcher and you are writing an academic paper. You have already completed the Abstract section of the target paper and have already collected the reference papers that should be included in the related work section. Now your task is to write the related work section of the target paper. Please read the target paper and the reference papers carefully, and generate the related work section according to the following steps:

#Step 1: Read the target paper and understand the main content of this paper precisely.

#Step 2: Read the reference papers one by one and identify the relationship of each reference paper and the target paper. Figure out the reason why the reference papers should be cited in the related work section. And summarize the reference papers in academic and concise manner.

#Step 3: Make sure the generated related work section fulfill the following objectives: (1) situates your work within the broader scholarly community - connects your work to the broader field and shows that your work has grown organically from current trends; (2)illustrates a "gap" in previous researches; (3) if needed, shows how you achieve the improvement compared with previous researches.

The input will be given in the following JSON format:

```
{
 "Target Paper":
 {
  "Title": xxxx,
  "Abstract":xxxx,
  "Introduction":xxxx,
  "Conclusion":xxxx
  },
 "Reference Papers":
 {
  "@cite_1":
   {
   "Title": xxxx,
   "Abstract":xxxx,
   "Introduction":xxxx,
   "Conclusion":xxxx
   },
   ...
   "@cite_n":
   {
   "Title": xxxx,
   "Abstract":xxxx,
   "Introduction":xxxx,
   "Conclusion":xxxx
   }
 }
}
```

"Target Paper" includes four key-value pairs: "title", "abstract", "introduction", and "conclusion".
"Reference Papers" contains multiple key-value pairs, where each key is a unique citation identifier (e.g., "@cite_1", ..., "@cite_n"), and each value is an object representing a reference paper. For each reference paper object, the meta information of the paper is provided, including "title", "abstract", "introduction", and "conclusion".

In the above input format, "@cite_1" ... "@cite_n" should be the citation markers of the corresponding references, which means when you cite one reference paper, you should use "@cite_?" to represent the corresponding reference paper.

Please also remember not to leave out any given reference.

Now I will give the input as follows:

Figure 10: Prompt template for Zero-shot Prompting.

You are an expert paper reviewer. You need to list the thematic groups of the related work section.

The related work will be given in the following JSON format:

```
{
 "<SENTENCE_1>": xxxx,
 "<SENTENCE_2>": xxxx,
 "<SENTENCE_3>": xxxx,
 ...
}
```

The output should be in the following JSON format:

```
{
 "thematic groups":
 {
  "theme_identifier": ["<SENTENCE_?>",...,"<SENTENCE_?>"],
  "theme_identifier": ["<SENTENCE_?>",...,"<SENTENCE_?>"],
  ...
  "theme_identifier": ["<SENTENCE_?>",...,"<SENTENCE_?>"],
 }
}
```

"thematic groups" should be a JSON object, with several key-value pairs, where the key is themetic identifier and the value is the list of the corresponding draft sentences identifier "<SENTENCE_?>".

I will first show you an example input and output:

Input:

```
{
 "<SENTENCE_1>": "The development of effective word representations is a cornerstone of progress in natural language processing (NLP), enabling systems to better understand and process human language by capturing semantic and syntactic nuances.",
 "<SENTENCE_2>": "Early approaches to word representation often treated words as atomic units, ignoring the rich morphological structure that many languages exhibit.",
 "<SENTENCE_3>": "This limitation has spurred research into more sophisticated models that can account for the internal structure of words, leading to significant improvements in various NLP tasks.",
 "<SENTENCE_4>": "One line of research has focused on leveraging morphological information to enhance word representations.",
 "<SENTENCE_5>": "For instance, the work by @cite_1 introduces a novel model that constructs representations for morphologically complex words from their constituent morphemes, combining recursive neural networks (RNNs) with neural language models to account for contextual information.",
 "<SENTENCE_6>": "This approach has shown to outperform existing word representations on word similarity tasks, highlighting the importance of morphological awareness in word representation.",
 "<SENTENCE_7>": "Similarly, @cite_4 presents a scalable method for integrating compositional morphological representations into vector-based probabilistic language models, demonstrating substantial reductions in perplexity and improvements in translation tasks for morphologically rich languages.",
 "<SENTENCE_8>": "Another significant advancement in the field has been the adoption of character-level models, which offer a way to mitigate the out-of-vocabulary (OOV) problem by composing word representations from smaller units.",
 "<SENTENCE_9>": "The work by @cite_2 describes a neural language model that relies solely on character-level inputs, employing a convolutional neural network (CNN) and a highway network over characters to produce word-level predictions.",
 "<SENTENCE_10>": "This model achieves state-of-the-art performance on several languages, underscoring the sufficiency of character inputs for language modeling.",
 "<SENTENCE_11>": "@cite_5 further explores this direction by introducing a model that constructs vector representations of words by composing characters using bidirectional LSTMs, achieving impressive results in language modeling and part-of-speech tagging, especially in morphologically rich languages.",
 "<SENTENCE_12>": "The exploration of character n-grams as a means to represent words and sentences has also yielded promising results.",
 "<SENTENCE_13>": "@cite_3 introduces CHARAGRAM embeddings, which represent textual sequences through character n-gram count vectors followed by a nonlinear transformation.",
 "<SENTENCE_14>": "This simple yet effective approach surpasses more complex architectures based on character-level RNNs and CNNs, setting new benchmarks on several similarity tasks.",
 "<SENTENCE_15>": "In addition to these developments, the field has seen efforts to enrich word embeddings with morpho-syntactic information.",
 "<SENTENCE_16>": "@cite_7 presents a graph-based semi-supervised learning method for generating morpho-syntactic lexicons, which, when used as features, improve performance in downstream tasks like morphological tagging and dependency parsing.",
 "<SENTENCE_17>": "@cite_8 proposes incorporating morphological information into word embeddings through a unified probabilistic framework, where morphological priors help improve embeddings for rare or unseen words.",
 "<SENTENCE_18>": "The integration of character-level information for part-of-speech tagging has been further explored by @cite_6, which proposes a deep neural network that combines word-level and character-level representations for enhanced accuracy in English and Portuguese.",
```

Figure 11: Prompt for sub-topic extractor

"<SENTENCE_19>": "The method of refining vector space representations using relational information from semantic lexicons, as proposed by @cite_10, shows substantial improvements in lexical semantic evaluation tasks, highlighting the importance of semantic lexicons in word vector refinement.",
"<SENTENCE_20>": "The challenges of morphological tagging in highly inflective languages are addressed by @cite_12, which uses an exponential probabilistic model to improve disambiguation of morphological categories.",
"<SENTENCE_21>": "Lastly, @cite_13 proposes an improved taxonomy for capturing grammatical relations across languages, enhancing the cross-linguistic applicability of the Stanford Dependencies representation.",
"<SENTENCE_22>": "Our work, \\\\\\\"Mimicking Word Embeddings using Subword RNNs,\\\\\\\" builds upon these foundations by presenting MIMICK, an approach that generates OOV word embeddings compositionally from spellings to distributional embeddings without requiring re-training on the original corpus.",
"<SENTENCE_23>": "This method not only addresses the limitations of previous models in handling OOV words but also demonstrates the potential of type-level learning for improving performance across a wide range of languages and NLP tasks.",
"<SENTENCE_24>": "By situating our work within this broader context, we aim to contribute to the ongoing dialogue in the field and address some of the gaps identified in previous research"
}

Output:
{
"thematic groups":
{
"general introduction on topic 'effective word representations": [["<SENTENCE_1>",""]],
"limitations of early approaches to word representation": [["<SENTENCE_2>",""], ["<SENTENCE_3>",""]],
"subtopic1: leveraging morphological information to enhance word representations.": [["<SENTENCE_4>",""], ["<SENTENCE_5>","@cite_1"], ["<SENTENCE_6>",""], ["<SENTENCE_7>","@cite_4"]],
"subtopic2: the adoption of character-level models": [["<SENTENCE_8>",""], ["<SENTENCE_9>","@cite_2"], ["<SENTENCE_10>",""], ["<SENTENCE_11>","@cite_5"]],
"subtopic3: The exploration of character n-grams as a means to represent words": [["<SENTENCE_12>",""], ["<SENTENCE_13>","@cite_3"], ["<SENTENCE_14>",""]],
"subtopic4: enrich word embeddings with morpho-syntactic information.": [["<SENTENCE_15>",""], ["<SENTENCE_16>","@cite_7"], ["<SENTENCE_17>","@cite_8"]],
"The integration of character-level information for part-of-speech tagging": [["<SENTENCE_18>","@cite_6"]],
"refining vector space representations": [["<SENTENCE_19>","@cite_10"]],
"morphological tagging": [["<SENTENCE_20>","@cite_12"]],
"capturing grammatical relations across languages": [["<SENTENCE_21>","@cite_13"]],
"introduction of the target paper": [["<SENTENCE_22>",""], ["<SENTENCE_23>",""], ["<SENTENCE_24>",""]]
}
}

Figure 12: Prompt for sub-topic extractor (Continued)

I will give you the full text of an academic paper. You need to extract as much information as possible about the objective, motivation, method, experimental result, conclusion, advantage, and limitation of the paper.

The input paper will be given in the following JSON format, with five keys "title", "abstract", "introduction", "conclusion", and "other sections", which refer to the title, the Abstract section, the Introduction section, the Conclusion section and other sections, respectively. The values are the corresponding contents:

{
"title": xxxx,
"abstract": xxxx,
"introduction": xxxx,
"conclusion": xxxx,
"other sections": xxxx
}

The output should also be in JSON format as follows:
{
"objective": (string) representing the objective of the paper,
"motivation": (string) representing the motivation behind the paper,
"method": (string) representing the method or approach used in the paper,
"experimental result": (string) representing the results obtained in the paper,
"conclusion": (string) representing the conclusion of the paper,
"advantages": (string) describing the advantages or strengths of the paper,
"limitations": (string) describing the limitations or weaknesses of the paper
}

Now I will give you the input:

Figure 13: Prompt for Key Aspects Extractor

You are a human evaluator and paper reviser. You will be given a target paper and some reference papers cited by the target paper, along with a draft related work section. Now you need to first judge whether the draft includes all the reference papers I have provided to you. If there are some reference papers not included in the draft, you need to regenerate the related work to include these missing references.

I will provide you with the draft related work, the target paper, and the reference papers in the following JSON format:

```
{
 "Target Paper":
 {
  "title": xxxx,
  "abstract": xxxx,
  "introduction": xxxx,
  "conclusion": xxxx
 },
 "Reference Papers":
 {
 "Total citation identifiers": [@cite_1, ... , @cite_n],
 "@cite_1":
 {
  "objective":  xxxx,
  "motivation":  xxxx,
  "method":  xxxx,
  "experimental result":  xxxx,
  "conclusion": xxxx,
  "advantages": xxxx,
  "limitations": xxxx
 },
 ...
 "@cite_n":
 {
  "objective":  xxxx,
  "motivation":  xxxx,
  "method":  xxxx,
  "experimental result":  xxxx,
  "conclusion": xxxx,
  "advantages": xxxx,
  "limitations": xxxx
 }
 },
 "Draft Related Work": xxxx,
}
```

"Target Paper" includes four key-value pairs: "title", "abstract", "introduction", and "conclusion".
"Reference Papers" are also structured as a JSON object, including "Total citation identifiers", which is a list that contains all the citation identifiers for all referenced papers (@cite_1, ..., @cite_n). And Each identifier (@cite_1, ..., @cite_n) is also a JSON object that represents an individual reference paper. For each reference paper object "@cite_n", the meta information of the paper is provided, including "objective", "motivation", "method", "experimental result", "conclusion", "advantages", and "limitations".

You need to solve this task step by step according to the following steps:

(1) Count the number of input reference papers N by counting the items of "Total citation identifiers";

(2) Count the number of cited reference papers M in the draft related work;

(3) if N > M, it means the draft related work fails to cite all the input reference papers; Then you should regenerate the related work to add all the missing reference papers.

(4) Remember that you should not simply concatenate the missing reference papers after the draft, but rather identify the relationship between the missing reference papers and the target paper, and put the missing reference papers to suitable position to make the related work contextual coherent. If the relationship is stated in the draft, then you should put the missing reference paper to the corresponding reasonable position. Otherwise, you should start a new paragraph to introduce the missing reference papers.

(5) if N = M, it means all the reference papers have been cited; Then you need to do nothing.

You should only output the refined related work as well as your modification operations towards the draft. The output should also be in JSON format as follows:

```
{
 "Refined Related Work": xxxx,
 "Modification Operations": xxxx,
}
```

I will first show you an example:

Figure 14: Prompt for Reference Paper Supplement

You are an expert paper reviewer. You need to evaluate the structure clarity of the related work draft written for a target paper and provide your operable instructions for improvements. Besides the related work, you will also be provided with information about the target paper as well as information about the reference papers it cites.

The target paper and the reference papers as well as the related work draft are given in the following JSON format:

```
{
 "Target Paper":
 {
 "title": xxxx,
 "abstract": xxxx,
 "introduction": xxxx,
 "conclusion": xxxx
 },
 "Reference Papers":
 {
 "Total citation identifiers": [@cite_1, ... , @cite_n],
 "@cite_1":
 {
 "objective":  xxxx,
 "motivation":  xxxx,
 "method":  xxxx,
 "experimental result":  xxxx,
 "conclusion": xxxx,
 "advantages": xxxx,
 "limitations": xxxx
 },
 ...
 "@cite_n":
 {
 "objective":  xxxx,
 "motivation":  xxxx,
 "method":  xxxx,
 "experimental result":  xxxx,
 "conclusion": xxxx,
 "advantages": xxxx,
 "limitations": xxxx
 }
 },
 "Related Work Draft":
 {
 "<SENTENCE_1>": xxxx,
 "<SENTENCE_2>": xxxx,
 "<SENTENCE_3>": xxxx,
 ...
 }
}
```

"Target Paper" includes four key-value pairs: "title", "abstract", "introduction", and "conclusion".
"Reference Papers" are also structured as a JSON object, including "Total citation identifiers", which is a list that contains all the citation identifiers for all referenced papers (@cite_1, ..., @cite_n). And Each identifier (@cite_1, ..., @cite_n) is also a JSON object that represents an individual reference paper. For each reference paper object "@cite_n", the meta information of the paper is provided, including "objective", "motivation", "method", "experimental result", "conclusion", "advantages", and "limitations".
"Related Work Draft" is the related work draft, in which the keys ("SENTENCE_1", ... "SENTENCE_n") represent the sentences of the draft in order.

You need to evaluate the structure clarity of the related work draft and give your instruction step by step according to the following steps:

1. Read the given target paper and all the reference papers, and make note of their contents.

2. Read the related work draft of the target paper.

3. List the thematic flows of the related work draft, and then check if the draft is well-written and well-organized.

4. Identify whether the organization of the reference papers is fragemented and loose.

5. Identify whether the draft contains abrupt transitions between sentences or themes.

6. Check whether there is unreasonable discourse organization in the draft. For example, the introduction of the target paper generally comes after the discussion of reference papers, rather than before introducing the reference papers.

Figure 15: Prompt for structural rationality evaluator

You should first generate the high-level thematic flows of the draft, and then point out the unreasonable text organization using sentence keys "<SENTENCE_?>", then you should give your instructions on how to improve structure clarity.

Remember when you give your instructions, you should use the following five pre-defined operations (Remove, Delete, Insert, Move_and_Modify, and Merge_and_Modify):

(1) Modify the sentence <SENTENCE_?> to include information ___.

(2) Delete the sentence <SENTENCE_?>.

(3) Insert a new sentence about ___ between the position of sentence <SENTENCE_n> and <SENTENCE_m>.

(4) Move sentence <SENTENCE_?> before sentence <SENTENCE_n>, then slightly Modify sentence <SENTENCE_?> and <SENTENCE_n> to make them contextual coherent.

(5) Merge different sub-themes ___, ___, ... ___ into a unified theme ___ by putting their sentences together, then slightly revise the sentences of the theme ___ to make them contexutal coherent and reduce fragmentation.

Remember that you should only give one instruction that deals with the most prominent problem. And do not suggest delete operation on any sentence including citation identifier "@cite_n".

The output should be in the following JSON format:

```
{
 "thematic flows":
 {
  "thematic name": ["<SENTENCE_?>",...,"<SENTENCE_?>"],
  "thematic name": ["<SENTENCE_?>",...,"<SENTENCE_?>"],
  ...
  "thematic name": ["<SENTENCE_?>",...,"<SENTENCE_?>"]
 },
 "most prominent problem in text organization": xxxx,
 "instructions": xxxx
}
```

"thematic flows" should be a JSON object, with several key-value pairs, where the key is themetic name and the value is the list of the corresponding draft sentences keys "<SENTENCE_?>".
"most prominent problem in text organization": refers to the most prominent problem in text organization. There should be only one problem.
"instructions": refers to the operation from the pre-defined operations, which is used to deal with the problem.

I will first show you an example input and output:
{example}

Figure 16: Prompt for structural rationality evaluator (Continued)

You are a scientist. Now you are writing the related work section of a target paper. You have already completed the related work draft and sent it to an expert reviewer for review. The reviewer reviewed your draft carefully and gave his feedback on the structure clarity of your draft and gave the instructions on how to improve the structure clarity and coherence. You need to revise your related work draft based on the target paper, the reference papers it cites, the draft, as well as the instructions from the reviewer. Please make sure you read and understand the instructions carefully. Please refer to the provided information while revising.

The input includes four parts:
(1) the target paper, including its title, abstract section, introduction section and conclusion section.
(2) the reference papers cited by the target paper, including the objective, motivation, method, experimental result, conclusion, advantage, and limitation of each reference paper summarized by experts.
(3) the related work draft
(4) the feedback from the reviewer.

The input information will be given in the following JSON format.

```
Input:
{
 "Target Paper":
 {
 "title": xxxx,
 "abstract": xxxx,
 "introduction": xxxx,
 "conclusion": xxxx
 },
 "Reference Papers":
 {
 "Total citation identifiers": [@cite_1, ... , @cite_n],
 "@cite_1":
 {
 "objective":  xxxx,
 "motivation":  xxxx,
 "method":  xxxx,
 "experimental result":  xxxx,
 "conclusion": xxxx,
 "advantages": xxxx,
 "limitations": xxxx
 },
 ...
 "@cite_n":
 {
 "objective":  xxxx,
 "motivation":  xxxx,
 "method":  xxxx,
 "experimental result":  xxxx,
 "conclusion": xxxx,
 "advantages": xxxx,
 "limitations": xxxx
 }
 },
 "Related Work Draft":
 {
 "<SENTENCE_1>": xxxx,
 "<SENTENCE_2>": xxxx,
 "<SENTENCE_3>": xxxx,
 ...
 },
 "Feedback From the Reviewer":
 {
  "thematic flows":
 {
  "thematic name": ["<SENTENCE_?>",...,"<SENTENCE_?>"],
  "thematic name": ["<SENTENCE_?>",...,"<SENTENCE_?>"],
  ...
  "thematic name": ["<SENTENCE_?>",...,"<SENTENCE_?>"]
 },
 "most prominent problem in text organization": xxxx,
 "instructions": xxxx
 }
}
```

"Target Paper" includes four key-value pairs: "title", "abstract", "introduction", and "conclusion".

"Reference Papers" is also structured as a JSON object, including "Total citation identifiers", which is a list that contains all the citation identifiers for all referenced papers (@cite_1, ..., @cite_n). And Each identifier (@cite_1, ..., @cite_n) is also a JSON object that represents an individual reference paper. For each reference paper object "@cite_n", the meta information of the paper is provided, including "objective", "motivation", "method", "experimental result", "conclusion", "advantages", and "limitations".

Figure 17: Prompt for structural rationality generator

You are an expert paper reviewer. You need to evaluate the succinctness of the related work draft written for a target paper. Besides the related work, you will also be provided with information about the target paper as well as information about the reference papers it cites.

The target paper and the reference papers as well as the related work draft are given in the following JSON format:

```
{
 "Target Paper":
 {
  "title": xxxx,
  "abstract": xxxx,
  "introduction": xxxx,
  "conclusion": xxxx
 },
 "Reference Papers":
 {
 "Total citation identifiers": [@cite_1, ... , @cite_n],
 "@cite_1":
 {
  "objective":  xxxx,
  "motivation":  xxxx,
  "method":  xxxx,
  "experimental result":  xxxx,
  "conclusion": xxxx,
  "advantages": xxxx,
  "limitations": xxxx
  },
  ...
  "@cite_n":
  {
  "objective":  xxxx,
  "motivation":  xxxx,
  "method":  xxxx,
  "experimental result":  xxxx,
  "conclusion": xxxx,
  "advantages": xxxx,
  "limitations": xxxx
  }
  },
  "Related Work Draft":
  {
  "<SENTENCE_1>": xxxx,
  "<SENTENCE_2>": xxxx,
  "<SENTENCE_3>": xxxx,
  ...
  }
}
```

"Target Paper" includes four key-value pairs: "title", "abstract", "introduction", and "conclusion".
"Reference Papers" are also structured as a JSON object, including "Total citation identifiers", which is a list that contains all the citation identifiers for all referenced papers (@cite_1, ..., @cite_n). And Each identifier (@cite_1, ..., @cite_n) is also a JSON object that represents an individual reference paper. For each reference paper object "@cite_n", the meta information of the paper is provided, including "objective", "motivation", "method", "experimental result", "conclusion", "advantages", and "limitations".
"Related Work Draft" is the related work draft, in which the keys ("SENTENCE_1", ... "SENTENCE_n") represent the sentences of the draft in order.

Please evaluate the succinctness of the related work draft step by step according to the following steps:
1. Read the given target paper and all the reference papers, and make note of their contents.
2. Read the related work draft of the target paper.
3. Check succinctness of citation: you should check whether the introduction of individual reference paper includes too much details. In general, the citing of a reference paper usually focuses on a particular aspect of "objective", "motivation", "method", "experimental result", and "conclusion", rather than multiple aspects. The particular aspect should be the most relevant aspect to the target paper. So If you find the introduce to a reference includes more than one aspect, then you should point out this problem.
4. Check succinctness of target paper: you should check the statements about introduction of own work in the draft to identify whether these statements contain too much redundant information. In general, in the related work, the authors should situate their own work in the context of reference papers and claim their contribution concisely. Other redundant information or irrelevant information should be removed.
5. Check sentence by sentence to identify whether it includes paper title. If so, then you should point out this problem.

Your output should be in the following JSON format:

```
{
  "Succinctness Problem": xxxx,
}
```

Where the value of "Succinctness Problem" is the problems about the succinctness of the draft.

I will first show you an example input and output:
{example}

Figure 18: Prompt for content succinctness evaluator

You are a scientist. Now you are writing the related work section of a target paper. You have already completed the related work draft and sent it to an expert reviewer for review. The reviewer reviewed your draft carefully and gave his feedback on the succinctness aspect of your draft. You need to revise your related work draft based on the target paper, the reference papers it cites, the draft, as well as the feedback from the reviewer. Please make sure you read and understand the feedback carefully. Please refer to the provided information while revising.

The input includes four parts:
(1) the target paper, including its title, abstract section, introduction section and conclusion section.
(2) the reference papers cited by the target paper, including the objective, motivation, method, experimental result, conclusion, advantage, and limitation of each reference paper summarized by experts.
(3) the related work draft
(4) the feedback from the reviewer.

The target paper and the reference papers as well as the related work draft are given in the following JSON format:

```
{
 "Target Paper":
 {
 "title": xxxx,
 "abstract": xxxx,
 "introduction": xxxx,
 "conclusion": xxxx
 },
 "Reference Papers":
 {
 "Total citation identifiers": [@cite_1, ... , @cite_n],
 "@cite_1":
 {
 "objective":  xxxx,
 "motivation":  xxxx,
 "method":  xxxx,
 "experimental result":  xxxx,
 "conclusion": xxxx,
 "advantages": xxxx,
 "limitations": xxxx
 },
 ...
 "@cite_n":
 {
 "objective":  xxxx,
 "motivation":  xxxx,
 "method":  xxxx,
 "experimental result":  xxxx,
 "conclusion": xxxx,
 "advantages": xxxx,
 "limitations": xxxx
 }
 },
 "Related Work Draft":
 {
 "<SENTENCE_1>": xxxx,
 "<SENTENCE_2>": xxxx,
 "<SENTENCE_3>": xxxx,
 ...
 },
 "Feedback From the Reviewer": xxxx,

}
```

"Target Paper" includes four key-value pairs: "title", "abstract", "introduction", and "conclusion".
"Reference Papers" are also structured as a JSON object, including "Total citation identifiers", which is a list that contains all the citation identifiers for all referenced papers (@cite_1, ..., @cite_n). And Each identifier (@cite_1, ..., @cite_n) is also a JSON object that represents an individual reference paper. For each reference paper object "@cite_n", the meta information of the paper is provided, including "objective", "motivation", "method", "experimental result", "conclusion", "advantages", and "limitations".
"Related Work Draft" is the related work draft, in which the keys ("SENTENCE_1", ... "SENTENCE_n") represent the sentences of the draft in order.
"Feedback From the Reviewer" includes the feedback from the reviewer on succinctness aspect of the draft.

You should improve the succinctness of the related work draft while ensuring all critical information are accurately maintained and ensure the contextual coherence. Use the information provided in "Target Paper" and "Reference Papers" to achieve a concise yet comprehensive revision.

Figure 19: Prompt for content succinctness generator

You can use the following three types of operations to revise the draft (Modify, Delete, and Merge):

(1) Modify the sentence <SENTENCE_?> to exclude information about ___ aspect.

(2) Delete the sentence <SENTENCE_?>.

(3) Merge different sentences <SENTENCE_?>,...,<SENTENCE_?> into a single sentence <SENTENCE_?> to make them more concise.

Rememeber when you revise the related work, the following principles should be followed:

(1) Do not delete a sentence easily, unless you think it's absolutely necessary.

(2) Do not exert delete operation on any sentence including citation identifier "@cite_n".

(3) Do not remove any citation identifier "@cite_n" when you modify a sentence or merge some sentences.

(4) Merge operation should be only exerted on different sentences that introduce the same reference paper or the target paper.

(5) when you delete one sentence, the contextual coherence cannot be damaged.

Your output should include (1) your actions on how to improve succinctness, (2) the revised related work. The output should be organized in the following JSON format:

```
{
 "Actions":
 {
  "1": xxxx,
  "2": xxxx,
  ...
 },
 "Revised Related Work":
 {
 "<SENTENCE_1>":{"content": xxxx, "trajectory": xxxx},
 "<SENTENCE_2>":{"content": xxxx, "trajectory": xxxx},
 ...
 "<SENTENCE_n>":{"content": xxxx, "trajectory": xxxx}
 }
}
```

Where the output JSON file should include two key-value pairs: "Actions" and "Revised Related Work":
The value of "Actions" is a JSON object, the key indicates the instruction index, the value refers to the instruction.
The value of "Revised Related Work" is also a JSON object, including multiple key-value pairs, where each key represents a sentence from the original related work section, and each corresponding value is an object containing two keys: "content": This key contains the revised content of the sentence, addressing the succinctness problem described in the "Succinctness Problem" key. "trajectory": This key contains information about the revision, which should be from the above pre-defined operations.

I will first show you an example input and output:
{example}

Figure 20: Prompt for content succinctness generator (Continued)

```
{
 "<SENTENCE_1>": "The burgeoning field of relation classification in natural language processing (NLP) has witnessed significant advancements through the integration of neural network architectures and the strategic utilization of syntactic information.",
 "<SENTENCE_2>": "Our work, \"A Position Encoding Convolutional Neural Network Based on Dependency Tree for Relation Classification,\" introduces a novel approach that leverages the syntactic structure of dependency parse trees to enhance relation classification.",
 "<SENTENCE_3>": "This section reviews relevant literature that has contributed to the development of relation classification methods, particularly those that incorporate dependency structures and neural network models.",
 "<SENTENCE_4>": "Early attempts to improve relation classification have focused on exploiting the syntactic structure provided by dependency trees.",
 "<SENTENCE_5>": "@cite_2 presents a pioneering approach that utilizes the shortest path in the dependency graph between two entities as a feature for relation extraction.",
 "<SENTENCE_6>": "This method underscores the importance of dependency paths in capturing the essential syntactic relations necessary for identifying semantic relationships between entities.",
 "<SENTENCE_7>": "Building on this concept, @cite_1 introduces an augmented dependency path (ADP) that combines the shortest dependency path with subtrees attached to this path.",
 "<SENTENCE_8>": "The study demonstrates how this enriched syntactic representation, processed through a dependency-based neural network, can achieve state-of-the-art results in relation classification.",
 "<SENTENCE_9>": "The exploration of neural network models for relation classification has led to the development of various architectures designed to capture the complex features necessary for this task.",
 "<SENTENCE_10>": "@cite_3 and @cite_6 both employ convolutional neural networks (CNNs) but differ in their approach to feature extraction and representation.",
 "<SENTENCE_11>": "While @cite_3 proposes a ranking-based classification model that leverages pairwise ranking loss to enhance performance, @cite_6 focuses on extracting both lexical and sentence-level features directly from word tokens using a deep CNN architecture.",
 "<SENTENCE_12>": "These studies highlight the versatility of CNNs in processing textual information for relation classification.",
 "<SENTENCE_13>": "Furthermore, the significance of capturing sequential and structural information along dependency paths is emphasized in @cite_4, which introduces a model based on long short-term memory networks (LSTMs) that operate along the shortest dependency paths (SDP-LSTM).",
 "<SENTENCE_14>": "This model effectively integrates heterogeneous information from different sources over the dependency paths, showcasing the potential of recurrent neural networks in understanding the syntactic and semantic nuances of language.",
 "<SENTENCE_15>": "In addition to neural network models, the role of semantic compositionality in relation classification is explored in @cite_7.",
 "<SENTENCE_16>": "This study presents a recursive neural network model that learns compositional vector representations for phrases and sentences, illustrating the importance of understanding the compositional meaning of language constructs for accurate relation classification.",
 "<SENTENCE_17>": "Our work builds upon these foundational studies by proposing a position encoding convolutional neural network (PECNN) that utilizes tree-based position features derived from dependency parse trees.",
 "<SENTENCE_18>": "Unlike previous approaches, our method redefines the context for relation classification by designing tree-based convolution kernels that capture both semantic and structural information provided by dependency trees.",
 "<SENTENCE_19>": "This innovative approach addresses the gap in existing research by enhancing word representations through the encoding of relative positions in dependency trees and by offering a more nuanced understanding of context in relation classification.",
 "<SENTENCE_20>": "In summary, our work is situated within a vibrant scholarly community that seeks to advance relation classification through the innovative use of syntactic information and neural network models.",
 "<SENTENCE_21>": "By introducing a novel method that emphasizes the structural and semantic richness of dependency parse trees, we contribute to the ongoing dialogue in the field and propose a new direction for future research."
}
```


Figure 21: Case study: Initial draft

{
"<SENTENCE_1>": "The burgeoning field of relation classification in natural language processing (NLP) has witnessed significant advancements through the integration of neural network architectures and the strategic utilization of syntactic information.",
"<SENTENCE_2>": "Our work, \"A Position Encoding Convolutional Neural Network Based on Dependency Tree for Relation Classification,\" introduces a novel approach that leverages the syntactic structure of dependency parse trees to enhance relation classification.",
"<SENTENCE_3>": "This section reviews relevant literature that has contributed to the development of relation classification methods, particularly those that incorporate dependency structures and neural network models.",
"<SENTENCE_4>": "Early attempts to improve relation classification have focused on exploiting the syntactic structure provided by dependency trees.",
"<SENTENCE_5>": "@cite_2 presents a pioneering approach that utilizes the shortest path in the dependency graph between two entities as a feature for relation extraction.",
"<SENTENCE_6>": "This method underscores the importance of dependency paths in capturing the essential syntactic relations necessary for identifying semantic relationships between entities.",
"<SENTENCE_7>": "Building on this concept, @cite_1 introduces an augmented dependency path (ADP) that combines the shortest dependency path with subtrees attached to this path.",
"<SENTENCE_8>": "The study demonstrates how this enriched syntactic representation, processed through a dependency-based neural network, can achieve state-of-the-art results in relation classification.",
"<SENTENCE_9>": "The exploration of neural network models for relation classification has led to the development of various architectures designed to capture the complex features necessary for this task.",
"<SENTENCE_10>": "@cite_3 and @cite_6 both employ convolutional neural networks (CNNs) but differ in their approach to feature extraction and representation.",
"<SENTENCE_11>": "While @cite_3 proposes a ranking-based classification model that leverages pairwise ranking loss to enhance performance, @cite_6 focuses on extracting both lexical and sentence-level features directly from word tokens using a deep CNN architecture.",
"<SENTENCE_12>": "These studies highlight the versatility of CNNs in processing textual information for relation classification.",
"<SENTENCE_13>": "Furthermore, the significance of capturing sequential and structural information along dependency paths is emphasized in @cite_4, which introduces a model based on long short-term memory networks (LSTMs) that operate along the shortest dependency paths (SDP-LSTM).",
"<SENTENCE_14>": "This model effectively integrates heterogeneous information from different sources over the dependency paths, showcasing the potential of recurrent neural networks in understanding the syntactic and semantic nuances of language.",
"<SENTENCE_15>": "In addition to neural network models, the role of semantic compositionality in relation classification is explored in @cite_7.",
"<SENTENCE_16>": "This study presents a recursive neural network model that learns compositional vector representations for phrases and sentences, illustrating the importance of understanding the compositional meaning of language constructs for accurate relation classification.",
"<SENTENCE_17>": "The challenges of extracting semantic relationships amidst sparse data and entity recognition errors are addressed in @cite_5, which employs Maximum Entropy models to combine diverse lexical, syntactic, and semantic features, highlighting the potential for scalable solutions in complex relation classification scenarios.",
"<SENTENCE_18>": "Lastly, the establishment of a new task for multi-way classification of semantic relations between pairs of common nominals as part of SemEval-2010 is introduced in @cite_8, underscoring the community's interest in robust knowledge extraction and the importance of semantic relations in various NLP applications.",
"<SENTENCE_19>": "Additionally, @cite_9's development of a new kernel method for relation extraction emphasizes the ongoing need for robust methods capable of handling POS or parsing errors, particularly in challenging domains like biological corpora.",
"<SENTENCE_20>": "Our work builds upon these foundational studies by proposing a position encoding convolutional neural network (PECNN) that utilizes tree-based position features derived from dependency parse trees.",
"<SENTENCE_21>": "Unlike previous approaches, our method redefines the context for relation classification by designing tree-based convolution kernels that capture both semantic and structural information provided by dependency trees.",
"<SENTENCE_22>": "This innovative approach addresses the gap in existing research by enhancing word representations through the encoding of relative positions in dependency trees and by offering a more nuanced understanding of context in relation classification.",
"<SENTENCE_23>": "In summary, our work is situated within a vibrant scholarly community that seeks to advance relation classification through the innovative use of syntactic information and neural network models.",
"<SENTENCE_24>": "By introducing a novel method that emphasizes the structural and semantic richness of dependency parse trees, we contribute to the ongoing dialogue in the field and propose a new direction for future research."
}

Figure 22: Case study: Summary after Reference Paper Supplement

{
"<SENTENCE_1>": "The burgeoning field of relation classification in natural language processing (NLP) has witnessed significant advancements through the integration of neural network architectures and the strategic utilization of syntactic information.",
"<SENTENCE_2>": "Our work, \"A Position Encoding Convolutional Neural Network Based on Dependency Tree for Relation Classification,\" introduces a novel approach that leverages the syntactic structure of dependency parse trees to enhance relation classification.",
"<SENTENCE_3>": "This section reviews relevant literature that has contributed to the development of relation classification methods, particularly those that incorporate dependency structures and neural network models.",
"<SENTENCE_4>": "Early attempts to improve relation classification have focused on exploiting the syntactic structure provided by dependency trees.",
"<SENTENCE_5>": "@cite_2 presents a pioneering approach that utilizes the shortest path in the dependency graph between two entities as a feature for relation extraction.",
"<SENTENCE_6>": "This method underscores the importance of dependency paths in capturing the essential syntactic relations necessary for identifying semantic relationships between entities.",
"<SENTENCE_7>": "Building on this concept, @cite_1 introduces an augmented dependency path (ADP) that combines the shortest dependency path with subtrees attached to this path.",
"<SENTENCE_8>": "The study demonstrates how this enriched syntactic representation, processed through a dependency-based neural network, can achieve state-of-the-art results in relation classification.",
"<SENTENCE_9>": "The transition from traditional methods to neural network models marks a pivotal evolution in the field, offering new perspectives and methodologies for tackling the complexities of relation classification.",
"<SENTENCE_10>": "Advancements in relation classification methods have also been marked by the exploration of neural network models, which have been instrumental in understanding semantic compositionality and introducing new tasks and methods.",
"<SENTENCE_11>": "@cite_3 and @cite_6 both employ convolutional neural networks (CNNs) but differ in their approach to feature extraction and representation, showcasing the versatility of CNNs in processing textual information for relation classification.",
"<SENTENCE_12>": "@cite_4 introduces a model based on long short-term memory networks (LSTMs) that operate along the shortest dependency paths (SDP-LSTM), effectively integrating heterogeneous information from different sources over the dependency paths.",
"<SENTENCE_13>": "@cite_7 presents a recursive neural network model that learns compositional vector representations for phrases and sentences, illustrating the importance of understanding the compositional meaning of language constructs for accurate relation classification.",
"<SENTENCE_14>": "Furthermore, @cite_5 addresses the challenges of extracting semantic relationships amidst sparse data and entity recognition errors by employing Maximum Entropy models to combine diverse lexical, syntactic, and semantic features.",
"<SENTENCE_15>": "@cite_8 introduces a new task for multi-way classification of semantic relations between pairs of common nominals as part of SemEval-2010, underscoring the community's interest in robust knowledge extraction.",
"<SENTENCE_16>": "@cite_9's development of a new kernel method for relation extraction emphasizes the ongoing need for robust methods capable of handling POS or parsing errors, particularly in challenging domains like biological corpora.",
"<SENTENCE_17>": "Our work builds upon these foundational studies by proposing a position encoding convolutional neural network (PECNN) that utilizes tree-based position features derived from dependency parse trees.",
"<SENTENCE_18>": "Unlike previous approaches, our method redefines the context for relation classification by designing tree-based convolution kernels that capture both semantic and structural information provided by dependency trees.",
"<SENTENCE_19>": "This innovative approach addresses the gap in existing research by enhancing word representations through the encoding of relative positions in dependency trees and by offering a more nuanced understanding of context in relation classification.",
"<SENTENCE_20>": "In summary, our work is situated within a vibrant scholarly community that seeks to advance relation classification through the innovative use of syntactic information and neural network models.",
"<SENTENCE_21>": "By introducing a novel method that emphasizes the structural and semantic richness of dependency parse trees, we contribute to the ongoing dialogue in the field and propose a new direction for future research."
}

Figure 23: Case study: Summary after Structural Rationality Enhancement

{
"<SENTENCE_1>": "The burgeoning field of relation classification in natural language processing (NLP) has witnessed significant advancements through the integration of neural network architectures and the strategic utilization of syntactic information.",
"<SENTENCE_2>": "Our work introduces a novel approach that leverages the syntactic structure of dependency parse trees to enhance relation classification.",
"<SENTENCE_3>": "This section reviews relevant literature that has contributed to the development of relation classification methods, particularly those that incorporate dependency structures and neural network models.",
"<SENTENCE_4>": "Early attempts to improve relation classification have focused on exploiting the syntactic structure provided by dependency trees.",
"<SENTENCE_5>": "@cite_2 presents a pioneering approach that utilizes the shortest path in the dependency graph between two entities as a feature for relation extraction, highlighting the significance of dependency paths in identifying semantic relationships.",
~~"<SENTENCE_6>": "This method underscores the importance of dependency paths in capturing the essential syntactic relations necessary for identifying semantic relationships between entities.",~~
"<SENTENCE_6>": "Building on this concept, @cite_1 introduces an augmented dependency path (ADP) that combines the shortest dependency path with subtrees attached to this path, achieving state-of-the-art results in relation classification through a dependency-based neural network.",
~~"<SENTENCE_8>": "The study demonstrates how this enriched syntactic representation, processed through a dependency-based neural network, can achieve state-of-the-art results in relation classification.",~~
"<SENTENCE_7>": "The transition from traditional methods to neural network models marks a pivotal evolution in the field, offering new perspectives and methodologies for tackling the complexities of relation classification.",
"<SENTENCE_8>": "Advancements in relation classification methods have also been marked by the exploration of neural network models, which have been instrumental in understanding semantic compositionality and introducing new tasks and methods.",
"<SENTENCE_9>": "@cite_3 and @cite_6 both employ convolutional neural networks (CNNs), showcasing the versatility of CNNs in processing textual information for relation classification.",
"<SENTENCE_10>": "@cite_4 introduces a model based on long short-term memory networks (LSTMs) that operate along the shortest dependency paths (SDP-LSTM), effectively integrating heterogeneous information from different sources.",
"<SENTENCE_11>": "@cite_7 presents a recursive neural network model that learns compositional vector representations for phrases and sentences, illustrating the importance of understanding the compositional meaning of language constructs for accurate relation classification.",
"<SENTENCE_12>": "Furthermore, @cite_5 addresses the challenges of extracting semantic relationships amidst sparse data and entity recognition errors by employing Maximum Entropy models to combine diverse lexical, syntactic, and semantic features.",
"<SENTENCE_13>": "@cite_8 introduces a new task for multi-way classification of semantic relations between pairs of common nominals as part of SemEval-2010, underscoring the community's interest in robust knowledge extraction.",
"<SENTENCE_14>": "@cite_9's development of a new kernel method for relation extraction emphasizes the ongoing need for robust methods capable of handling POS or parsing errors, particularly in challenging domains like biological corpora.",
"<SENTENCE_15>": "Our work builds upon these foundational studies by proposing a position encoding convolutional neural network (PECNN) that utilizes tree-based position features derived from dependency parse trees.",
"<SENTENCE_16>": "Unlike previous approaches, our method redefines the context for relation classification by designing tree-based convolution kernels that capture both semantic and structural information provided by dependency trees.",
"<SENTENCE_17>": "This innovative approach addresses the gap in existing research by enhancing word representations through the encoding of relative positions in dependency trees and by offering a more nuanced understanding of context in relation classification."
~~"<SENTENCE_20>": "In summary, our work is situated within a vibrant scholarly community that seeks to advance relation classification through the innovative use of syntactic information and neural network models.",~~
~~"<SENTENCE_21>": "By introducing a novel method that emphasizes the structural and semantic richness of dependency parse trees, we contribute to the ongoing dialogue in the field and propose a new direction for future research."~~
}

Figure 24: Case study: Summary after Content Succinctness Enhancement