From References to Insights: Collaborative Knowledge Minigraph Agents for Automating Scholarly Literature Review

Zhi Zhang[1], Yan Liu[1]*, Sheng-hua Zhong[2], Gong Chen[1], Yu Yang[1], Jiannong Cao[1]

[1]The Hong Kong Polytechnic University
[2]Shenzhen University
zhi271.zhang@connect.polyu.hk, yan.liu@polyu.edu.hk, csshzhong@szu.edu.cn, csgchen@comp.polyu.edu.hk, cs-yu.yang@polyu.edu.hk, jiannong.cao@polyu.edu.hk

Abstract

Literature reviews play a crucial role in scientific research for understanding the current state of research, identifying gaps, and guiding future studies on specific topics. However, the process of conducting a comprehensive literature review is yet time-consuming. This paper proposes a novel framework, collaborative knowledge minigraph agents (CKMAs)[1], to automate scholarly literature reviews. A novel prompt-based algorithm, the knowledge minigraph construction agent (KMCA), is designed to identify relationships between information pieces from academic literature and automatically constructs knowledge minigraphs. By leveraging the capabilities of large language models on constructed knowledge minigraphs, the multiple path summarization agent (MPSA) efficiently organizes information pieces and relationships from different viewpoints to generate literature review paragraphs. We evaluate CKMAs on three benchmark datasets. Experimental results demonstrate that the proposed techniques generate informative, complete, consistent, and insightful summaries for different research problems, promoting the use of LLMs in more professional fields.

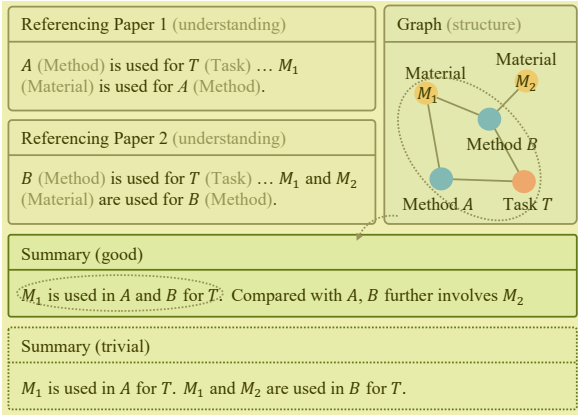

Figure 1: Illustration of relationships between information pieces in scientific papers. Capturing these relationships is essential for composing a coherent story in literature reviews.

Introduction

Artificial intelligence (AI) is being increasingly integrated into scientific discovery to augment and accelerate scientific research (Wang et al. 2023). Researchers are developing AI methods to, e.g., literature understanding, experiment development, and manuscript draft writing (Liu et al. 2022; Wang et al. 2024; Martin-Boyle et al. 2024).

Literature reviews play a crucial role in scientific research, assessing and integrating previous research on specific topics (Bolanos et al. 2024). They aim to meticulously identify and appraise all relevant literature related to a specific research question. Recent advancements in AI have shown promising performance in understanding research papers and generating human-like text (Van Dinter, Tekinerdogan, and Catal 2021). By leveraging AI capabilities, automatic literature review models enable researchers to save time and effort in the manual process of conducting literature reviews, rapidly identify key trends and gaps in recent research outputs, and uncover insights that might be overlooked in manual reviews (Wagner, Lukyanenko, and Paré 2022).

The automatic literature review models typically involve two stages (Shi et al. 2023): (1) selecting relevant reference papers and (2) determining logical relationships to compose a summary that presents the evolution of a specific field (these stages can be applied iteratively). Multiple scientific document summarization (MSDS), aiming to generate coherent and concise summaries for clusters of topic relevant scientific papers, is the representative work in the second stage. Past decades (Jin, Wang, and Wan 2020) have witnessed the development of summarization methods. Extractive methods directly select important sentences from original papers. Abstractive methods can generate new words and sentences but are technically more challenging than extractive methods.

Recently, large language models (LLMs), pre-trained on extensive text data, have transformed abstractive summarization and show human-like performance in understanding and coherent language synthesis. However, ideas arising in research papers often have complex relationships, e.g., conflicting or duplicate. Without explicit instructions, LLMs fall short in capturing the relations between ideas and composing a story that connects related work reflecting the author's understanding of their field (Li and Ouyang 2024). As shown

*Corresponding author.
[1]https://aaai-2025-4471.github.io/

in Fig 1, effective summarization often involves the ability to understand concepts of materials, methods, and tasks in referencing papers, aggregate complementary ideas (e.g., M_1 is used for T) while contrasting differences (e.g., M_2 is additionally used for B compared with A).

To tackle this bottleneck, we propose equipping LLMs with structural knowledge. Different from knowledge graphs, which consist of entities as nodes and their relationships as edges, serving as general-purpose knowledge, we introduce knowledge minigraphs. Knowledge minigraphs are small-scale semantic graphs, comprising research-relevant concepts as nodes and their relationships as edges, specially designed to capture the structural information between ideas in references.

To automatically construct knowledge minigraphs, we propose a prompt-based algorithm, the knowledge minigraph construction agent (KMCA) to elicit LLMs to identify research-relevant concepts and relationships based on references. Benefiting from the designed iterative construction strategy, key information and relationships are iteratively extracted and stored from numerous references into minigraphs.

By leveraging the capabilities of LLMs on knowledge minigraphs, the multiple path summarization agent (MPSA) is designed to organize the generated literature review. The MPSA samples multiple summaries from different viewpoints and logical paths in the knowledge minigraph, utilizing the mixture of experts technique. A self-evaluation mechanism is then employed to automatically route to the most desirable summary as the final output.

Related Work

Graphs in MSDS Tasks

To generate a summary that is representative of the overall content, graph-based methods construct external graphs to assist document representation and cross-document relation modeling, achieving promising progress. In this regard, LexRank (Erkan and Radev 2004) and TextRank (Mihalcea and Tarau 2004) first introduced graphs to extractive text summarization in 2004. They compute sentence importance using a graph representation of sentences to extract salient textual units from documents as summarization.In 2020, Wang et al. (Wang et al. 2020) propose to extract salient textual units from documents as summarization using a heterogeneous graph consisting of semantic nodes at several granularity levels of documents. In 2022, Wang et al. (Wang et al. 2022) incorporate knowledge graphs into document encoding and decoding, generating the summary from a knowledge graph template to achieve state-of-the-art performance.

However, to the best of our knowledge, no existing work integrates LLMs into graph-based methods to leverage their natural language understanding capabilities for improved graph construction and summary generation.

Pre-trained Language Models in MSDS Tasks

In recent years, pre-trained language models (PLMs) have demonstrated promising results in multiple document summarization. Liu et al. (Liu and Lapata 2019) propose fine-tuning a pre-trained BERT model as the encoder and a randomly initialized decoder to enhance the quality of generated summaries. Building upon BART (Lewis et al. 2020), Beltagy et al. (Beltagy, Peters, and Cohan 2020) propose LED for lengthy text summarization, which is directly initialized from bart-large but employs global-local attention to better handle long context inputs. Xiao et al. (Xiao et al. 2022) introduce PRIMERA, a pre-trained encoder-decoder multi-document summarization model, by improving aggregating information across documents. More recently, pre-trained large language models (LLMs) show promising generation adaptability by training billions of model parameters on massive amounts of text data (Zhao et al. 2023; Minaee et al. 2024). Zhang et al. (Zhang et al. 2024a) utilize well-designed instructions to extract key elements, arrange key information, and generate summaries. Zakkas et al. (Zakkas, Verberne, and Zavrel 2024) propose a three-step approach to select papers, perform single-document summarization, and aggregate results.

PLMs can provide fluent summary results for referencing papers. However, they fall short in capturing the relations between ideas from multiple related papers.

Method

Fig. 2 illustrates the architecture of the proposed collaborative knowledge minigraph agents (CKMAs). CKMAs consists of two key components: the knowledge minigraph construction agent and the multiple path summarization agent.

Knowledge Minigraph Construction Agent

In this module, we are given T reference documents $\{C_1, \ldots, C_t\}$'s abstracts. We aim to construct a knowledge structure O that captures the relationships between concepts in the referenced papers.

Past decades have witnessed knowledge graphs become the basis of information systems that require access to structured knowledge (Zou 2020). Knowledge structures are represented as semantic graphs, where nodes denote entities and are connected by relations denoted by edges. However, the general-purpose knowledge graphs are unsuitable for scientific document summarization, as they do not necessarily involve the main ideas of research papers and are not suitable for identifying gaps not addressed by prior work. Thus, in this paper, we propose establishing a knowledge minigraph, defined as as a small set of research-relevant concepts and their relationships. The construction steps of the knowledge minigraph are as follows:

Reference chunking Given a total of T reference documents, we first divide them into I chunks, each containing at most k reference documents. Here, $\{C_1^i, \ldots C_k^i\}$ represents the k reference papers in the i-th chunk. MSDS usually involves numerous reference papers, forming a long context. LLMs either fail to process the entire context exceeding the acceptable length or suffer from missing crucial information positioned amidst a lengthy context (Zhang et al. 2024b). Thus, we chunk related works and use LLMs to construct the knowledge structure step by step with k reference papers each time.

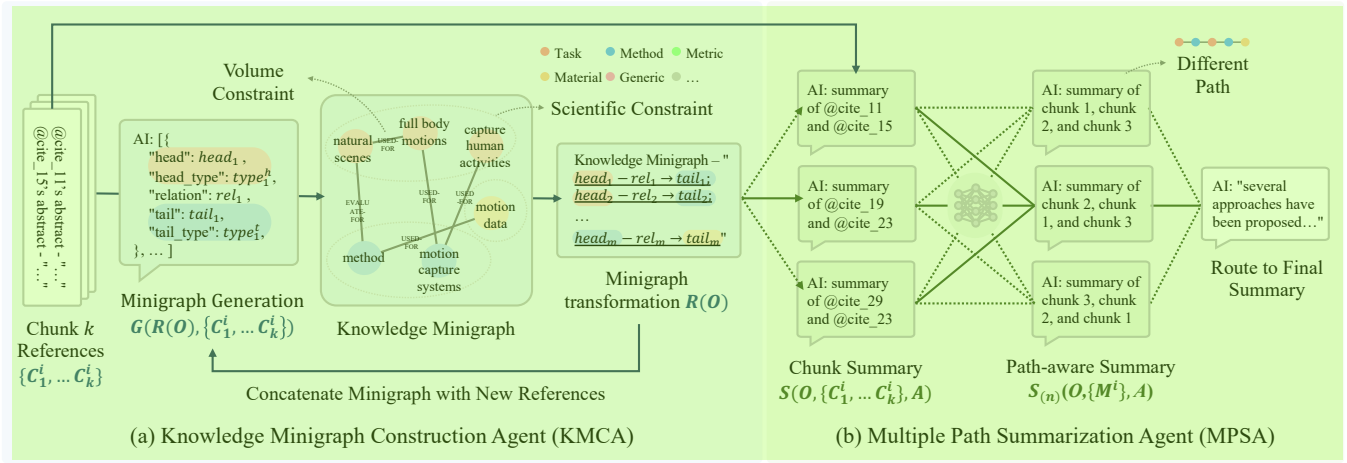

Figure 2: The overall architecture of the proposed collaborative knowledge minigraph agents (CKMAs).

Minigraph generation We employ a knowledge minigraph prompt $G(\{C_1^i, \dots C_k^i\})$ to construct the knowledge structure of interest based on the abstracts of referenced papers $\{C_1^i, \dots C_k^i\}$. It is known that constructing knowledge graphs from raw text data requires entity recognition and relation extraction (Trajanoska, Stojanov, and Trajanov 2023). Tailoring recent advancements in prompt techniques, we design instructions for the integration of LLM into these tasks within a single round of dialogue (using the prompt for query and LLMs for response). As shown in Table 1, the prompt involves three special designs:

(1) Output Constraint: LLMs are well known for being relatively verbose and free-form in their output, making it hard for automated graph construction programs (Tan and Motani 2023). Thus, we prompt LLM to constrain output in a machine-understandable JSON format. (2) Scientific Constraints: To ensure the constructed knowledge structure revolves around the main idea of research topics, we design constraints on entities of interest and relationships of interest. Inspired by DYGIE++ (Wadden et al. 2019), we extract entities classified in six types, Task, Method, Metric, Material, Generic, and OtherScientificTerm, and relations are in seven types, Compare, Used-for, Feature-of, Hyponym-of, Evaluate-for, Part-of, and Conjunction. (3) Volume Constraints: Redundant relationships will lead to length issues. To ensure the constructed knowledge structure is concise and informative, we constrain the focus to the top-m significant relationships.

Minigraph transformation To enable LLMs to understand the derived knowledge minigraph, we design a function $R(O)$ to transform the knowledge minigraph O into a text representation for subsequent processing. In detail, available relationships whose type meet the constraints are transformed into a line of text in the format $head_p - rel_p \rightarrow tail_p$, where $head_p$ and $tail_p$ are the head and tail entities, and rel_p is the relationship.

Finally, iteratively employing these three steps, the knowledge structure O^i is constructed as shown in Eq. 1.

$$O^i = \begin{cases} G(R(O^{i-1}), \{C_1^i, \dots C_k^i\}), i \geq 2 \\ G(\{C_1^i, \dots C_k^i\}); i = 1 \end{cases} \quad (1)$$

In the first iteration, we apply the knowledge minigraph prompt G to the first chunk of reference papers to derive the initial knowledge minigraph O^1. For subsequent iterations, we transform the intermediate knowledge minigraph O^{i-1} into a text representation $R(O^{i-1})$. This text representation is then input along with the i-th chunk of reference papers $\{C_1^i, \dots C_k^i\}$, allowing for dynamic updates to the knowledge minigraph.

Multiple Path Summarization Agent

In this module, we are given the knowledge structure O, the referencing paper's abstract A, and the chunked referencing papers $\{C_1^i, \dots C_k^i\}$'s abstracts. We aim to generate a summary following the knowledge structure.

Even given O as guidance, generating a summary remains an ill-posed problem, i.e., the solution is not unique and depends on the specific discussion viewpoints. For example, one can highlight different concepts or choose different writing logic for different situations. Can we harness different understandings of the knowledge structure to create a more capable summary? Inspired by the mixture of experts approach (Shazeer et al. 2017), a machine learning technique to leverage diverse model capabilities where multiple expert networks specialize in different skill sets, we propose using LLMs with different hinted paths to understand the knowledge minigraph for generating multiple summaries and selecting the best viewpoint. The steps of the multiple path summarization agent are as follows:

Chunk summarization As mentioned before, MSDS usually involves numerous reference papers, forming a long context problem. We chunk them into I chunks and formulate a hierarchical summarization process, first generating summaries for each chunk of referenced papers. With prompt engineering, we elicit the behavior of LLMs to generate summaries for each chunk under the guidance of de-

Table 1: Prompts in the knowledge minigraph construction agent and the multiple path summarization agent.

Knowledge Minigraph Construction Agent (KMCA)		Multiple Path Summarization Agent (MPSA)	
Description	Prompt	Description	Prompt
Role Play	You are an advanced algorithm designed to extract information in structured formats for building a knowledge graph. Your task is to identify entities and relations from a given text based on the user's prompt.	Default Requirement	You are the most famous researcher in writing the related work section of a given scientific article. Your summaries are concise, informative and of high quality. You are the author of a scientific article. You have already written the abstract of the article, and you are currently writing the related work section of the article. You want to write a paragraph of at most 200 words, which will be used without modification as a paragraph in the related work section that refers to the referenced documents, either to base on their ideas or to challenge them. Be fluent. Avoid repetitive information. Refer to the referenced documents of the list using their $id in this format "@cite_$id". All documents should be cited. You are encouraged to cite more than one document in one place if you are sure that the citation is supported by their summaries.
Output Constraint	Generate the output in JSON format, containing a list of objects with keys: "head", "head_type", "relation", "tail", and "tail_type".		
Scientific Constraints	The "head" and "tail" keys should contain the extracted entity text. The "head_type" and "tail_type" keys must be one of the following: "Task", "Method", "Metric", "Material", "Other-Scientific-Term", "Generic". The "relation" key must be one of these types: "Compare", "Conjunction", "Evaluate-For", "Used-For", "FeatureOf", "Part-Of", "Hyponym-Of".		
Volume Constraints	Extract up to 32 of the most important relations, prioritizing significant and relevant information. Ensure entity consistency by using the most complete identifier for entities mentioned multiple times in different forms.	Input	Scientific article's abstract: {{abstract}} Referenced documents' summaries: {{reference_index}}'s abstract - "{{reference_abstract}}" Referenced documents' knowledge graphs: {{knowledge_graphs}} Written paragraph:
Input	Text: {{knowledge_graph}} {{reference_index}}'s abstract – "{{reference_abstract}}" Extracted entities and relations:		

rived knowledge minigraphs O. As illustrated in Table 1, we instruct LLMs to take into consideration three kinds of information: the scientific article's abstract A, summaries of referenced paper $\{C_1^i, \dots C_k^i\}$, and the knowledge minigraphs of referenced paper O. After understanding this information, the LLM is tasked with writing a summary for each chunk, where A and $\{C_1^i, \dots C_k^i\}$ provide textual details locally and O provide structural knowledge globally. Flexible to customize other instruction details, such as specifying writing style in real-world applications, the default instructions follow the prompts designed by Zakkas et al. (Zakkas, Verberne, and Zavrel 2024) for fair comparison. Mathematically, the chunk summarization can be denoted as:

$$M^i = S(A, \{C_1^i, \dots C_k^i\}, O) \tag{2}$$

where S is the prompt function for chunk summarization.

Path-aware summarization We employ E experts to merge all chunk summaries $\{M^i\}$ and generate final summaries, with each expert aware of different hinted paths to understand the knowledge minigraph. Given consistent knowledge, different human researchers may have varying understandings, selectively emphasizing concepts in a logical order. To automatically mimic human researchers and generate summaries of different understanding, we leverage the observation that LLMs are sensitive to prompt wording and their order (Pezeshkpour and Hruschka 2023). We find that the order of given referencing papers impacts the generated summary, affecting the order of introducing information from references and the highlighting of concepts. Thus, we propose sampling E permutations from the full permutations of referenced papers to serve as the order of input in the instructions as a hint of the potential logical path in knowledge minigraph, which are then used to prompt the LLM to generate summaries. We use the same prompt as in chunk summarization, except that summaries of referenced papers $\{C_1^i, \dots C_k^i\}$ are replaced by a permutation of chunk summaries $\{M^i\}$. Mathematically, the summaries generated by the e-th expert can be denoted as:

$$Y_{(e)} = S_{(e)}(A, \{M^i\}, O) \tag{3}$$

where $S_{(e)}$ represents the prompt function for the e-th expert with the sampled e-th permutation of referencing papers. For instance, given chunk summarizations $\{M_1, M_2, M_3\}$, where M_1, M_2, and M_3 are summaries of the first, second, and third chunks of referencing papers, respectively, three experts can be fed $[M_1, M_2, M_3]$, $[M_2, M_1, M_3]$, and $[M_3, M_2, M_1]$ as input. The remaining parts of the instructions remain consistent with S.

Summarization router We design a router to evaluate different experts' summaries and automatically select the most desirable summary $Y_{(e)}$ as the final output. Without requiring additional side information, this paper proposes a self-evaluation strategy. In detail, we observe that there are agreements between the experts' viewpoints and their generated summaries. We propose to quantify the degree of agreement for each summary using the ROUGE-1 score (Lin 2004), which measures the overlap between a generated summary and other summaries. We then select the summary with the highest degree of agreement, which indicates that its understanding has the highest likelihood of being supported by other experts, or in other words, is relatively more acceptable. Mathematically, the final summary can be denoted as:

$$Y = \arg\max_e \sum_{j \neq e} \text{rouge1}(Y_{(e)}, Y_{(j)}) \tag{4}$$

where $\text{rouge1}(Y_{(e)}, Y_{(j)})$ is the an 1-gram recall (ROUGE-1 score) between e-expert's generated summary $Y_{(e)}$ and j-expert's generated summary $Y_{(j)}$.

Table 2: Comparasion of CKMAs with state-of-the-art methods on Multi-Xscience, TAS2, and TAD datasets. 1

Method		Multi-Xscience		TAD		TAS2	
		ROUGE-1	ROUGE-2	ROUGE-1	ROUGE-2	ROUGE-1	ROUGE-2
Graphs	LexRank (2004)	30.19	5.53	27.29	3.50	27.04	3.18
	TexRank (2004)	31.51	5.83	26.80	3.61	26.19	3.14
	HeterSumGraph (2020)	31.36	5.82	27.85	3.88	27.56	3.62
	GraphSum (2020)	29.58	5.54	26.12	4.03	25.01	3.23
	TAG (2022)	33.45	7.06	30.48	6.16	28.04	4.75
	KGSum (2022)	35.77	7.51	32.38	5.19	30.67	4.76
PLMs	Pointer-Generator (2017)	34.11	6.76	31.70	6.41	28.53	4.96
	BertABS (2019)	31.56	5.02	27.42	4.88	25.45	3.82
	SciBertABS (2019)	32.12	5.59	27.88	5.19	26.01	4.13
	HiMAP (2019)	31.66	5.91	30.49	6.21	28.37	5.07
	BART (2020)	32.83	6.36	25.39	4.74	27.73	4.80
	MGSum (2020)	33.11	6.75	27.49	4.79	25.54	3.75
	PRIMERA (2022)	31.90	7.40	32.04	5.78	29.99	5.07
	GPT-3.5-turbo (2023)	31.11	7.38	30.77	4.78	26..97	4.14
	GPT-4 (2023)	33.21	7.61	32.50	4.90	30.71	4.25
	3A-COT (2024)	23.65	4.85	23.02	3.73	22.65	3.43
	SumBlogger (2024)	35.40	8.40	33.90	5.51	30.69	3.92
Proposed		**36.41**	**8.78**	**34.16**	**6.22**	**32.31**	**5.36**

2

Experiments 3

We evaluate our method on three public MSDS datasets: Multi-Xscience (Lu, Dong, and Charlin 2020), TAD (Chen et al. 2022), and TAS2 (Chen et al. 2022). Multi-Xscience is the first large-scale and open MDSS dataset, collected from arXiv and the Microsoft Academic Graph (MAG). It contains 5,093 instances for testing, primarily focusing on the computer science field. TAD and TAS2 are collected from the public scholar corpora S2ORC and Delve, respectively. While TAD contains papers from multiple fields, TAS2 focuses on the computer science field. Both TAD and TAS2 contain 5,000 instances for testing. The input and output format of the three datasets are consistent: each instance contains the abstract of a query paper and the abstracts of reference papers it cites as input, with a paragraph from the related work section of the query paper serving as the gold summary. 4

For a fair comparison, we follow relevant studies (Zakkas, Verberne, and Zavrel 2024) of prompt-based methods to use the same LLM, gpt-3.5-turbo, as the backbone model. We set the temperature of the sampling to 0.0 for reproducibility. We set the chunk size k to 3 and the number of experts E to 3. We set the volume constraint m to 32. Following previous work, we automatically evaluate the summarization quality using ROUGE scores (Lin 2004). We employ ROUGE-N to calculate the N-grams overlap between the output and gold summary, assessing the summary informativeness: 5

$$\text{ROUGE} - \text{N} = \frac{\sum_{X \in U} \sum_{\text{gram}_n \in X} \text{C}_{\text{match}}(\text{gram}_n)}{\sum X \in U \sum_{\text{gram}_n \in X} \text{C}(\text{gram}_n)} \quad (5)$$

where X denotes a reference summary sampled from the reference summary collection U, n represents the length of the n-gram, $\text{C}(.)$ is the count of the n-gram, and $\text{C}_{\text{match}}(.)$ is the maximum number of n-grams co-occurring in a candidate summary and a set of reference summaries. We report unigram and bigram co-occurrence (ROUGE-1 and ROUGE-2). 6 7

Comparasion Experiments 8

Table 2 compares the proposed model with graph-based methods including LexRank (Erkan and Radev 2004), TextRank (Mihalcea and Tarau 2004), HeterSumGraph (Wang et al. 2020), GraphSum (Li et al. 2020), TAG (Chen et al. 2022), and KGSum (Wang et al. 2022) and pre-trained language model-based methods, including Pointer-Generator (See, Liu, and Manning 2017), BertABS (Liu and Lapata 2019), SciBertABS (Beltagy, Lo, and Cohan 2019), HiMAP (Fabbri et al. 2019), BART (Lewis et al. 2020), MGSum (Jin, Wang, and Wan 2020), PRIMERA (Xiao et al. 2022), GPT-3.5-turbo (Ouyang et al. 2022), and GPT-4 (Achiam et al. 2023). The proposed CKMAs achieves state-of-the-art performance on all three datasets in terms of ROUGE-1 and ROUGE-2 scores. CKMAs also outperforms the latest prompt-powered MSDS, e.g., 3A-COT (Zhang et al. 2024a) and SumBlogger (Zakkas, Verberne, and Zavrel 2024). 9

Ablation Studies 10

This section presents ablation studies to investigate the performance gains brought by the designed modules in CKMAs. We first ablate the knowledge minigraph construction agent (KMCA) and the multiple path summarization agent (MPSA), respectively. When ablating KMCA, we no longer construct the knowledge minigraph and do not include it as part of the MPSA instruction. When removing MPSA, we use the instruction in Table 1 to directly generate a single summary as the final summary. We report the performance 11

Table 3: Ablation study of the proposed collaborative knowledge minigraph agents (CKMAs) on the Multi-Xscience dataset.

Knowledge Minigraph Construction Agent (KMCA)					Multiple Path Summarization Agent (MPSA)				
Scientific Constraint	Volume Constraint	Iterative Construction	ROUGE-1	ROUGE-2	Chunk Summary	Path Permutation	Summary Router	ROUGE-1	ROUGE-2
×	×	×	34.90	8.56	×	×	×	32.04	5.54
×	✓	✓	35.69	8.62	×	✓	✓	33.29	6.47
✓	×	✓	35.50	8.59	✓	×	✓	34.00	7.09
✓	✓	×	35.04	8.57	✓	✓	×	32.18	6.32
✓	✓	✓	**36.41**	**8.78**	✓	✓	✓	**36.41**	**8.78**

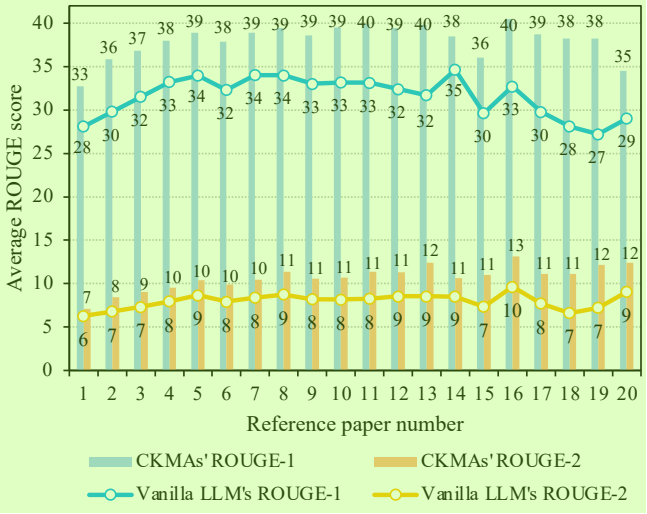

Figure 3: ROUGE score comparison of the proposed CKMAs with its backbone models (vanilla GPT-3.5-turbo) group by reference paper number on the Multi-Xscience dataset.

of the ablated version of the model in the first line of Table 3.

Then, we ablate the modules in MPSA and KMCA. For the knowledge minigraph construction agent, when ablating the scientific constraint or volume constraint ablation in minigraph generation, we remove the corresponding instruction in Table 1. To ablate iterative construction, we remove Eq. 1 and directly use all references as input. Due to the length issue, the over-length context is truncated. For the multiple path summarization agent, when ablating chunk summarization, we directly use all references' abstracts as input. To ablate the path-aware summarization strategy, we use the references with original order as input, adjusting temperatures from 0.0 to 0.7. When removing the summarization router, we randomly select a generated summary.

Table 3 shows the results of ablation studies, with the full version of the proposed model reported in the last line. We find that removing any module leads to performance degradation. This indicates that all designs contribute to the final performance. The MPSA brings a 4% performance gain, and KMCA brings a 2% performance gain. For designs in the knowledge minigraph construction agent, the performance gain brought by iterative construction is the largest, indicating its effectiveness in understanding long contexts. For designs in the multiple path summarization agent, the performance gain brought by the summarization router is the largest, indicating the importance of selecting the most desirable summaries from different logical paths.

Case Studies

This section conducts case studies to provide further insights into our model's performance. We first perform statistical analysis to validate in which cases the model succeeds and in which it fails. We group the test samples based on the number of references contained in the gold summary. In every group, we calculate the average ROUGE-L scores for CKMAs and its backbone model (vanilla GPT-3.5-turbo), comparing generated summaries with gold summaries. The results are presented in Fig. 3. We observe that CKMAs consistently outperforms vanilla GPT-3.5-turbo regardless of the number of referencing papers. As the number of reference papers increases, the performance gap between CKMAs and vanilla GPT-3.5-turbo widens. CKMAs demonstrates the capability to model complex relationships within long context, even achieving better results due to more references benefiting the understanding of the research, in contrast to the performance decrease observed in vanilla GPT-3.5-turbo.

For a detailed comparison, we sample an instance from the Multi-Xscience dataset and use well-known LLMs, GPT-3.5-turbo and GPT 4.0, to generate summaries with the default requirements listed in Table 1. The generated results are shown in Table 4. We find that GPT-3.5-turbo suffers from information loss, omitting citation 13. GPT 4.0 shows improvement but lists facts in parallel without logical connections. For example, citations 37 and 14 are listed side by side, but show no parallel logical relationship. It can even lead to hallucination problems, as no evidence shows citation 16 is a statistical method, while citation 16 and citations 3, 6, 5 are all categorized as statistical methods. We then use different versions of the proposed CKMAs to generate a summary. It can be observed that without the knowledge minigraph construction agent (KMCA), the multiple path summarization agent (MPSA) contributes to highlighting different categories of algorithms from the desired viewpoint. Without MPSA, the KMCA contributes to organizing algorithms logically, e.g., from probabilistic to statistical approaches, and then to example-based learning methods. With

Table 4: Case study of the proposed collaborative knowledge minigraph agents (CKMAs) on the Multi-Xscience dataset.

ChatGPT 3.5 (ROUGE-1: 29.50)	GPT 4.0 (ROUGE-1: 33.46)	Proposed w/o KMC (ROUGE-1: 38.91)
Information missing: @cite_13 is missing … word sense disambiguation … the naive mix algorithm as introduced by @cite_37 … @cite_14 presents a novel probabilistic modeling technique … the findings of @cite_16 highlight … @cite_5's comparative analysis of learning algorithms underscores … @cite_24 provides empirical support for …	Hallucinated logic: @cite_16's parallel … @cite_24 noted the relationship between sense and collocation while @cite_13 emphasized the value of context … @cite_37 and @cite_14 … the former unveiling the naive mix algorithm ... and the latter … focusing on systematic variable interactions … the exemplar-based learning algorithm … @cite_3 … @cite_6 @cite_5 and @cite_16 further broaden the conversation through the use of statistical methods …	Highlights key information, e.g., models word sense disambiguation … for instance, @cite_13 demonstrated the effectiveness of … while @cite_16 presented a model selection approach … similarly @cite_3 presents an exemplar-based learning algorithm … furthermore @cite_24 shows that a polysemous word … @cite_37… @cite_14 … probabilistic models … statistical methods … @cite_6 …
Proposed w/o MAG (ROUGE-1: 34.01)	**Proposed (ROUGE-1: 41.88)**	**Gold Summary**
Organized logic, e.g., probabilistic to statistical … word sense disambiguation … for instance @cite_37 and @cite_14 proposed probabilistic models … cite 6 on the … statistical sense resolution methods … furthermore @cite_3 presented exemplar-based learning … @cite_24 showed that a polysemous word … @cite_5 compared seven different learning algorithms … finally @cite_16 expanded existing model selection methodology …	Organized logic, highlight key information … word sense disambiguation including supervised learning algorithms such as … @cite_37 … probabilistic models … @cite_14 and … statistical methods @cite_6 … other approaches include exemplar-based learning … @cite_3 model selection … @cite_16 and … @cite_13 additionally …. @cite_24 and … @cite_5 these approaches … evaluated using various criteria …	Human written summary, special focus word sense disambiguation has more commonly been cast as a problem in supervised learning, e.g., @cite_13 @cite_2 @cite_24 @cite_6 @cite_14 @cite_5 @cite_3 @cite_16 @cite_37 …

both modules, CKMAs generates the best summary, categorizing algorithms as supervised learning as in the gold summary and detailing subcategories in a logical order.

We then analyze the differences between queried public knowledge graphs and the constructed knowledge minigraphs. We sample an instance from the Multi-Xscience dataset for this comparison. To query the knowledge graph, we use SPARQL to access Wikidata, a collaborative knowledge base. The queried knowledge graph is shown in the upper part of Fig. 4. For the knowledge minigraph, we employ the proposed method with knowledge minigraph construction, with the result displayed in the lower part. We observe that the entities in the queried knowledge graph are general-purpose and lack specific insights into research problems. The minigraph clearly presents tasks and methods (some including metrics and materials), making it more informative for summarization purposes.

Conclusions and Future Work

This paper aims to provide an intelligent research copilot to assist in writing literature reviews based on given references. While recent LLMs excel at natural language understanding and generation, they struggle to explicitly model complex relationships between ideas from multiple papers. To address this challenge, we propose collaborative knowledge minigraph agents (CKMAs). The contributions of this work are threefold: (1) We propose scientific document-oriented knowledge minigraphs and, for the first time, equip LLMs with knowledge minigraphs for multiple scientific document summarization. (2) We are the first to develop a prompt-based iterative algorithm to process a vast amount of literature and automatically construct knowledge minigraphs for multiple scientific document summarization. (3) We firstly introduce a mixture of experts' mechanisms to attempt the organization of literature reviews with different logical paths on minigraphs and derive the best one via self-evaluation.

We conduct comparison experiments, ablation studies,

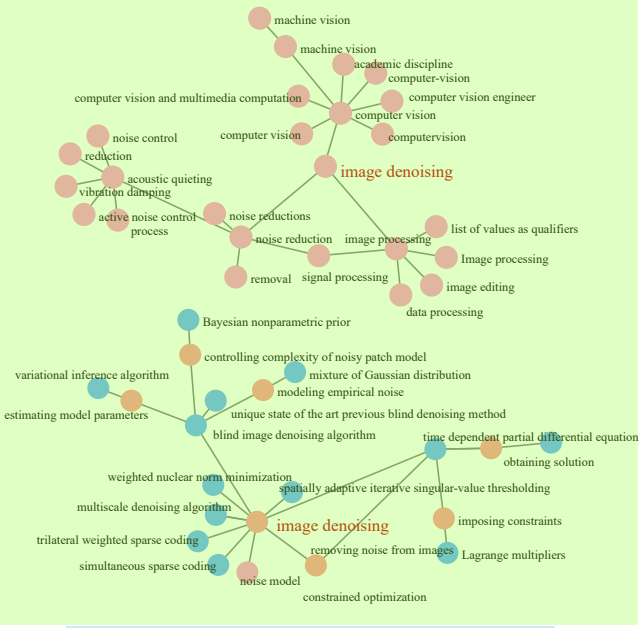

Figure 4: Case study of knowledge graphs queried from Wikidata (top) and knowledge minigraph constructed by CKMAs (bottom) for the topic "image denoising".

and case studies. The results show the effectiveness of the proposed method. For future work, we plan to explore fine-tuning LLMs with the proposed CKMAs to better follow instructions and approximate human-written literature reviews in collected datasets. We also intend to investigate the possibility of generating full survey papers with multiple paragraphs, which involve more scientific documents and more complex relationships and require planning of the survey paper's organization.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Beltagy, I.; Lo, K.; and Cohan, A. 2019. SciBERT: A Pretrained Language Model for Scientific Text. In *Conference on Empirical Methods in Natural Language Processing*, 3615–3620.

Beltagy, I.; Peters, M. E.; and Cohan, A. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Bolanos, F.; Salatino, A.; Osborne, F.; and Motta, E. 2024. Artificial intelligence for literature reviews: Opportunities and challenges. *arXiv preprint arXiv:2402.08565*.

Chen, X.; Alamro, H.; Li, M.; Gao, S.; Yan, R.; Gao, X.; and Zhang, X. 2022. Target-aware abstractive related work generation with contrastive learning. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, 373–383.

Erkan, G.; and Radev, D. R. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22: 457–479.

Fabbri, A. R.; Li, I.; She, T.; Li, S.; and Radev, D. 2019. Multi-News: A Large-Scale Multi-Document Summarization Dataset and Abstractive Hierarchical Model. In *Annual Meeting of the Association for Computational Linguistics*, 1074–1084.

Jin, H.; Wang, T.; and Wan, X. 2020. Multi-granularity interaction network for extractive and abstractive multi-document summarization. In *Annual Meeting of the Association for Computational Linguistics*, 6244–6254.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Annual Meeting of the Association for Computational Linguistics*, 7871–7880.

Li, W.; Xiao, X.; Liu, J.; Wu, H.; Wang, H.; and Du, J. 2020. Leveraging Graph to Improve Abstractive Multi-Document Summarization. In *Annual Meeting of the Association for Computational Linguistics*, 6232–6243.

Li, X.; and Ouyang, J. 2024. Related Work and Citation Text Generation: A Survey. *arXiv preprint arXiv:2404.11588*.

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

Liu, S.; Cao, J.; Yang, R.; and Wen, Z. 2022. Generating a structured summary of numerous academic papers: Dataset and method. In *International Joint Conference on Artificial Intelligence*, 4259–4265.

Liu, Y.; and Lapata, M. 2019. Text Summarization with Pretrained Encoders. In *Conference on Empirical Methods in Natural Language Processing*, 3730–3740.

Lu, Y.; Dong, Y.; and Charlin, L. 2020. Multi-XScience: A Large-scale Dataset for Extreme Multi-document Summarization of Scientific Articles. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 8068–8074.

Martin-Boyle, A.; Tyagi, A.; Hearst, M. A.; and Kang, D. 2024. Shallow Synthesis of Knowledge in GPT-Generated Texts: A Case Study in Automatic Related Work Composition. *arXiv preprint arXiv:2402.12255*.

Mihalcea, R.; and Tarau, P. 2004. Textrank: Bringing order into text. In *Conference on Empirical Methods in Natural Language Processing*, 404–411.

Minaee, S.; Mikolov, T.; Nikzad, N.; Chenaghlu, M.; Socher, R.; Amatriain, X.; and Gao, J. 2024. Large language models: A survey. *arXiv preprint arXiv:2402.06196*.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744.

Pezeshkpour, P.; and Hruschka, E. 2023. Large language models sensitivity to the order of options in multiple-choice questions. *arXiv preprint arXiv:2308.11483*.

See, A.; Liu, P. J.; and Manning, C. D. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *Annual Meeting of the Association for Computational Linguistics*.

Shazeer, N.; Mirhoseini, A.; Maziarz, K.; Davis, A.; Le, Q.; Hinton, G.; and Dean, J. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.

Shi, Z.; Gao, S.; Zhang, Z.; Chen, X.; Chen, Z.; Ren, P.; and Ren, Z. 2023. Towards a Unified Framework for Reference Retrieval and Related Work Generation. In *Findings of the Association for Computational Linguistics*, 5785–5799.

Tan, J. C. M.; and Motani, M. 2023. Large language model (llm) as a system of multiple expert agents: An approach to solve the abstraction and reasoning corpus (arc) challenge. *arXiv preprint arXiv:2310.05146*.

Trajanoska, M.; Stojanov, R.; and Trajanov, D. 2023. Enhancing knowledge graph construction using large language models. *arXiv preprint arXiv:2305.04676*.

Van Dinter, R.; Tekinerdogan, B.; and Catal, C. 2021. Automation of systematic literature reviews: A systematic literature review. *Information and Software Technology*, 136: 106589.

Wadden, D.; Wennberg, U.; Luan, Y.; and Hajishirzi, H. 2019. Entity, Relation, and Event Extraction with Contextualized Span Representations. In *Conference on Empirical Methods in Natural Language Processing*.

Wagner, G.; Lukyanenko, R.; and Paré, G. 2022. Artificial intelligence and the conduct of literature reviews. *Journal of Information Technology*, 37(2): 209–226.

Wang, D.; Liu, P.; Zheng, Y.; Qiu, X.; and Huang, X.-J. 2020. Heterogeneous Graph Neural Networks for Extractive Document Summarization. In *Annual Meeting of the Association for Computational Linguistics*, 6209–6219.

Wang, H.; Fu, T.; Du, Y.; Gao, W.; Huang, K.; Liu, Z.; Chandak, P.; Liu, S.; Van Katwyk, P.; Deac, A.; et al. 2023. Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972): 47–60.

Wang, P.; Li, S.; Pang, K.; He, L.; Li, D.; Tang, J.; and Wang, T. 2022. Multi-Document Scientific Summarization from a Knowledge Graph-Centric View. In *International Conference on Computational Linguistics*, 6222–6233.

Wang, Q.; Edwards, C.; Ji, H.; and Hope, T. 2024. Towards a Human-Computer Collaborative Scientific Paper Lifecycle: A Pilot Study and Hands-On Tutorial. In *Joint International Conference on Computational Linguistics, Language Resources and EvaluationSummaries*, 56–67.

Xiao, W.; Beltagy, I.; Carenini, G.; and Cohan, A. 2022. PRIMERA: Pyramid-based Masked Sentence Pre-training for Multi-document Summarization. In *Annual Meeting of the Association for Computational Linguistics*, 5245–5263.

Zakkas, P.; Verberne, S.; and Zavrel, J. 2024. SumBlogger: Abstractive Summarization of Large Collections of Scientific Articles. In *European Conference on Information Retrieval*, 371–386.

Zhang, Y.; Gao, S.; Huang, Y.; Yu, Z.; and Tan, K. 2024a. 3A-COT: an attend-arrange-abstract chain-of-thought for multi-document summarization. *International Journal of Machine Learning and Cybernetics*, 1–19.

Zhang, Z.; Chen, R.; Liu, S.; Yao, Z.; Ruwase, O.; Chen, B.; Wu, X.; and Wang, Z. 2024b. Found in the middle: How language models use long contexts better via plug-and-play positional encoding. *arXiv preprint arXiv:2403.04797*.

Zhao, W. X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; Zhang, J.; Dong, Z.; et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Zou, X. 2020. A survey on application of knowledge graph. In *Journal of Physics: Conference Series*, volume 1487, 012016.