MSG-LLM: A Multi-scale Interactive Framework for Graph-enhanced Large Language Models

Jiayu Ding[1†], Zhangkai Zheng[1†], Benshuo Lin[1], Yun Xue[1*], Yiping Song[2*]
[1]Guangdong Provincial Key Laboratory of Quantum Engineering and Quantum Materials,
School of Electronic Science and Engineering (School of Microelectronics),
South China Normal University, Foshan, China
[2]National University of Defense Technology, Changsha, China
{dingjiayu,zheng_zk,lin_benshuo,xueyun}@m.scnu.edu.cn songyiping@nudt.edu.cn

Abstract

Graph-enhanced large language models (LLMs) leverage LLMs' remarkable ability to model language and use graph structures to capture topological relationships. Existing graph-enhanced LLMs typically retrieve similar subgraphs to augment LLMs, where the subgraphs carry the entities related to our target and relations among the entities. However, the retrieving methods mainly focus solely on accurately matching subgraphs between our target subgraph and the candidate subgraphs at the same scale, neglecting that the subgraphs with different scales may also share similar semantics or structures. To tackle this challenge, we introduce a graph-enhanced LLM with multi-scale retrieval (MSG-LLM). It captures similar graph structures and semantics across graphs at different scales and bridges the graph alignment across multiple scales. The larger scales maintain the graph's global information, while the smaller scales preserve the details of fine-grained sub-structures. Specifically, we construct a multi-scale variation to dynamically shrink the scale of graphs. Further, we employ a graph kernel search to discover subgraphs from the entire graph, which essentially achieves multi-scale graph retrieval in Hilbert space. Additionally, we propose to conduct multi-scale interactions (message passing) over graphs at various scales to integrate key information. The interaction also bridges the graph and LLMs, helping with graph retrieval and LLM generation. Finally, we employ a Chain-of-Thought-based LLM prediction to perform the downstream tasks. We evaluate our approach on two graph-based downstream tasks and the experimental results show that our method achieves state-of-the-art performance.

1 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities in understanding and generalizing unstructured text. Given the powerful representation ability of graphs for structured data, it is natural to combine LLMs with graphs to achieve better performance in the field of natural language processing, including both classification and generation tasks(Li et al., 2024; Chen et al., 2024; Zhang et al., 2024).

Recently, there has been a surge in investigating effective methods for leveraging graph-structured information to enhance LLMs' capabilities. These methods typically employ subgraphs obtained through exact matching to augment the structural information available to LLMs. Some of these studies incorporate graph embeddings with LLM. GNP (Tian et al., 2024) uses soft prompts to provide GNN-modeled structured information to LLMs, and GraphAdapter (Huang et al., 2024) incorporates GNNs as adapters to combine with LLMs. While other studies enable LLMs to reason according to the graph structure and integrate structured information into the reasoning process. ToG (Sun et al., 2023) and MD-QA (Wang et al., 2024) query graphs directly through LLMs. However, the precise matching of the target graph only utilizes local information in the large graph, ignoring more globally related information in the entire graph.

To address these issues, some approaches aim to retrieve multiple relevant subgraphs from the entire graph, rather than relying solely on a single subgraph obtained through exact matching to enhance LLMs. Graph RAG (Edge et al., 2024) suggests partitioning a knowledge graph into community graphs and generating community summaries as retrieval targets. G-retriever (He et al., 2024) frames subgraph retrieval as a prize-collecting Steiner tree (PCST) optimization problem, aiming to include the maximum number of relevant nodes and edges.

[†]These authors contributed equally to this work.
[*]Corresponding author.

Proceedings of the 31st International Conference on Computational Linguistics, pages 9687–9700
January 19–24, 2025.

Additionally, GRAG (Hu et al., 2024) encodes k-hop subgraphs around each node as the target subgraph, retrieving subgraphs similar to the target graph according to the graph embeddings. However, although existing retrieval methods identify more relevant subgraphs, they mainly focus on matching the target subgraph with candidate subgraphs at the same scale. In reality, small and large subgraphs may also match when viewed across different scales. Current approaches overlook the potential for subgraphs of varying scales to share similar semantics or structures, which could provide valuable contextual information. This limitation prevents capturing more comprehensive relationships within the graph and limits the model's ability to leverage the graph's multi-scale semantic richness fully.

In this paper, we introduce a graph-enhanced LLM with multi-scale retrieval (MSG-LLM) and evaluate its effectiveness on two of the most common NLP tasks: text classification (emotion recognition in conversation) and text generation (citation generation). MSG-LLM captures similar graph structures and semantics across different scales, enabling alignment and interaction of graphs at multiple scales. The graphs also interact with LLMs to accomplish some NLP tasks. Specifically, we first construct a multi-scale variation to dynamically adjust the scale of graphs by abandoning unimportant nodes (subgraphs). We employ a graph kernel search to discover subgraphs from the entire graph, where the subgraphs match our target subgraphs across different scales in Hilbert space. Further, we propose a multi-scale interaction module to conduct message passing over graphs at various scales to integrate key information. The interaction module also bridges graphs and LLMs: the graphs step-by-step arouse LLMs via CoT, while LLMs provide textual descriptions about the graphs' semantic information to help graph retrieval. This bidirectional flow allows the graphs to provide structured data that enhances LLM reasoning, while the LLM offers unstructured textual information to guide graph retrieval, ensuring a more comprehensive understanding of the graph's semantics. Finally, we construct a CoT-based LLM prediction module to perform the downstream tasks. The experiments show that our model achieves state-of-the-art performance on two graph-based downstream tasks. We summarize our contributions as follows:

- We propose a novel method to convert and preserve multi-scale graph information, allowing LLMs to effectively utilize structured data across different graph scales.

- We enable full bidirectional interaction between the graph and LLM, where graph retrieval supplies structured information and the LLM's text generation complements this with unstructured data.

- We conduct experiments on datasets for two tasks, demonstrating that our method outperforms both traditional graph-based and language model-based methods.

2 Related work

2.1 Graph-enhanced Text Generation

Graph Retrieval-Augmented Generation is a technique that integrates graph retrieval and generative models to improve the performance of generation tasks, particularly those requiring extensive background knowledge and contextual understanding. By integrating graph structural information into generative models, these approaches improve both the quality and relevance of the generated outputs. To better capture the topological information of graphs, Kang et al. (2023) and Kim et al. (2023) focus on retrieving triples, aiming to represent complex relational data. CBR-KBQA (Das et al., 2021) combines the query and the retrieved pairs for generation. These approaches rely on subgraphs obtained through strict matching to enhance retrieval, but these subgraphs do not necessarily enhance the generation performance of language models. In addition, some methods attempt to improve information coverage by retaining multiple retrieval results. For example, Edge et al. (2024) employs a community detection algorithm to partition the graph into multiple communities, retrieving and aggregating relevant ones to generate the final answer to the query. Similarly, GMT-KBQA (Hu et al., 2022) re-ranks retrieved entities and relations, performing relationship classification and entity disambiguation before generation. Although these methods preserve multiple matching results, they fail to leverage the potential multi-scale information in the graph and do not fully explore relationships across different levels, limiting the model's comprehensive understanding and effective utilization of the graph's information.

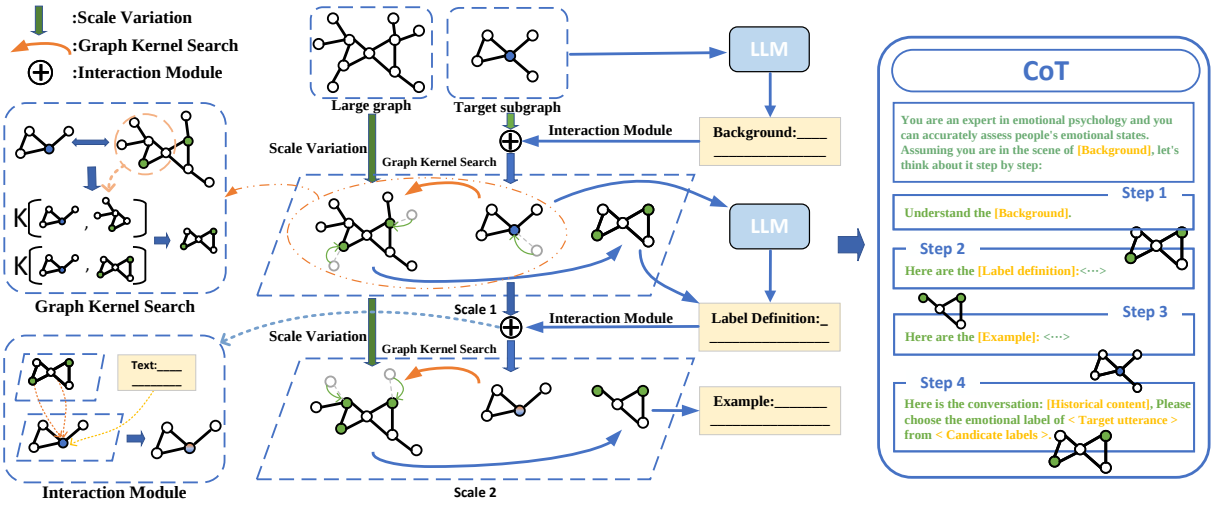

Figure 1: The framework of MSG-LLM with four parts: Multi-Scale Variations, Graph Kernel Search, Multi-Scale Interaction Module, and CoT-based LLM Prediction Module. The green arrows in the figure represent operations of Scale Variations. The sub-figure in the upper-left part shows the Graph Kernel Search module. The sub-figure in the bottom-left part shows the Interaction Module. We feed the subgraphs retrieved at multiple scales into the LLM in the form of Chain-of-Thought, as shown on the right side.

2.2 Emotion Recognition in Conversation

Emotion Recognition in Conversation (ERC) can be broadly categorized into three groups: The first category is graph-based methods, which utilize nodes and edges to model relationships between dialogues and speakers(Ghosal et al., 2019; Ishiwatari et al., 2020). The second category is transformer-based methods, which utilize or adapt transformer blocks to establish long-range emotional dependencies (Zhong et al., 2019; Song et al., 2022; Lei et al., 2023). The third category is graph-transformer methods, which combine the advantages of both graph structures and transformer architectures. S+PAGE (Liang et al., 2022) uses a two-stream conversational Transformer to extract both the self and inter-speaker contextual features for each utterance and employ SPGCN to model the speaker dependency and position information. MFAM (Hou et al., 2023) treats the pre-trained conversation model as a prior knowledge base and forms the weights of edges in the graph neural network. MKFM (Tu et al., 2023) integrates knowledge generated by LLMs and builds a directed graph to model context dependencies. This paper incorporates multi-scale graph information to enhance LLMs. The large-scale graph preserves conversations that are similar to the current conversation and can be used for topic understanding, thereby narrowing down the range of emotional labels; The small-scale only focuses on the context closely related to the current conversation, retaining details that can directly affect emotional judgment.

2.3 Context-Specific Citation Generation

In recent years, the growing volume of scientific literature has increased the demand for citation generation. Previous studies approach this task as a specialized form of text summarization (Hoang and Kan, 2010; Hu and Wan, 2014; Chen and Zhuge, 2019). However, citation generation differs from text summarization, as it requires a context-specific description of the cited paper relative to the citing paper. PTGEN-Cross (Xing et al., 2020) enhances citation generation by embedding citation sentences within specific contexts of the document. AutoCite (Wang et al., 2021) employs a multi-task model to simultaneously infer relevant work and generate citation contexts. DisenCite(Wang et al., 2022) integrates paper text and citation relationships to automatically generate citations tailored to specific contexts. Şahinuç et al. (2024) proposes to systematically explore the task of citation text generation with LLMs. This paper proposes the use of multi-scale citation relationship graphs. The larger-scale identifies papers that are similar to the current paper in only one aspect, such as task background, base model, or experimental setup, which can be used to determine the citation perspective; The smaller scale pays attention to the similarities and differences between the citing and cited pa-

per, which can be directly used for citation content generation.

3 Method

3.1 Overview

Given a large graph G and a target subgraph g_t, our task is to infer the task-specific information p for g_t, leveraging LLM and the information from retrieved subgraph g_r in G. As illustrated in Figure 1, the proposed Multi-Scale Graph-Enhanced LLM (MSG-LLM) involves **Multi-Scale Variations**, **Graph Kernel Search**, **Multi-Scale Interaction Module** and **CoT-based LLM Prediction Module**. MSG-LLM scales the large graph G to generate scaled graphs $\tilde{G} = \{\tilde{G}_1, \tilde{G}_2 \cdots \tilde{G}_k\}$ at different scales, which retain the key information across various scales. The multi-scale interaction module enables the target subgraph g_t to absorb information from different scales and LLMs, yielding g_{inter}. To leverage multi-scale information, MSG-LLM uses graph kernels to retrieve subgraphs similar to g_{inter} across various scales from $\tilde{G}$, resulting in the retrieved subgraphs g_r. MSG-LLM provides g_t and g_r in the form of Chain-of-Thought (CoT) to the LLM, resulting in the final prediction. Algorithm 1 illustrates the overall framework of the model.

We define the crucial notations as follows: The text-attributed graphs are defined as $G = (V, A, S)$, where V represents the set of nodes, A is the adjacency matrix, S represents the natural language attributes of V. Given a target subgraph $g_t = (V_t, A_t, S_t)$, our task is to infer the task-specific information p for g_t by the LLM. We apply scale variations to G, yielding K scaled graphs, denoted as $\tilde{G} = \{\tilde{G}_1, \tilde{G}_2 \cdots \tilde{G}_k\}$. For the target subgraph g_t, let g_r^k denote the retrieved subgraphs from $\tilde{G}_k$ at the k-th scale. Thus, $g_r = \{g_r^1, g_r^2 \cdot g_r^K\}$ represents the set of retrieved subgraphs across all scales.

3.2 Multi-Scale Variations

To capture the multi-scale information of the graph, we apply K scale variations to the entire graph. The scale of a graph refers to the total number of nodes and edges it contains, which can be regarded as its size or capacity. For the initial graph G, we retain the important nodes and aggregate the remaining nodes based on the scaling factor r_k, transforming G into a graph at k-th scale, denoted as $\tilde{G}_k$. Since text embeddings are used to calculate similarity during aggregating, we leverage a pre-trained language model (PLM) to convert text attributes of nodes in G, denoted as X.

To preserve key information in the graph, we first score the nodes based on an importance function, which evaluates the nodes using task-specific indicators, such as information entropy, degree, or other relevant metrics. We retain the top-k nodes with the highest scores based on the scaling factor r_k, denoted as V_{imp}^k, which denotes the set of important nodes retained after scaling. The remaining unimportant nodes are represented as V_{unimp}^k.

To mitigate the graph information loss due to scale variation, we aggregate the features of nodes in V_{unimp}^k to those in V_{imp}^k based on the matching relationships M^k between them:

Algorithm 1: MSG-LLM Algorithm

Input: $G = (V, A, S)$;
Target graph: $g_t = (V_t, A_t, S_t)$;
Number of scale: K;
Intra transformation matrix: W_{intra};
Cross transformation matrix: W_{cross}
Output: Task-specific prediction of g_t

for $k = 1, 2, 3, \ldots, K$ **do**
  Scale variation:
  $\tilde{G}_k = ScaleVariation(G, k)$
  Absorb information from LLM:
  $e_k \leftarrow (LLM, g_t)$
  $W'_{intra} \leftarrow (e_k, W_{intra})$
  if $k = 1$ **then**
    Bidirectional interaction:
    $H_{inter}^k = \sigma(A_{intra} \cdot H_t^k \cdot W'_{intra})$
  end
  if $k! = 1$ **then**
    // Construct cross-scale edge connections A_{cross}
    $A_{cross} \leftarrow (g_t, g_r^{k-1})$
    Bidirectional interaction and scale interaction:
    $H_r^{k-1} \leftarrow g_r^{k-1}$
    $H_{inter}^k = \sigma(A_{intra} \cdot H_t^k \cdot W'_{intra} + A_{cross} \cdot H_r^{k-1} \cdot W_{cross})$
  end
  Graph kernel search:
  $g_r^k = KS(A_t, H_{inter}^k, \tilde{G}_k)$
end
LLM predictor:
$P = LLM(\text{prompt}, g_r^1, g_r^2, \ldots, g_r^K)$

$$D_{ii}^k = \sum_j m_{ij} \tag{1}$$

$$X_k' = (D^k)^{-1} M^k X_k^{unimp} \tag{2}$$

$$M_{sim}^k = \frac{X_k^{imp} X_k'}{|X_k^{imp}||X_k'|} \tag{3}$$

$$\tilde{X}_k = X_k^{imp} + I M_{sim}^k X_k' \tag{4}$$

where $M_{ij}^k = 1$ indicates a matching relationship between the i-th node in V_{imp}^k and j-th node in V_{unimp}^k, while $M_{ij}^k = 0$ indicates no such relationship. X_k' represents the aggregated features of X_k^{unimp} based on the matching relationships. M_{sim}^k is the similarity matrix between X_k^{imp} and X_k'. I is the identity matrix, and $\tilde{X}_k$ is the feature matrix of $\tilde{G}_k$. The adjacency matrix $\tilde{A}_k$ of $\tilde{G}_k$ is obtained by the corresponding edges in V_{imp}^k. The matching strategy is also tailored to the downstream task. For instance, in conversational contexts, unimportant nodes are matched with similar nodes within the same conversation, whereas in citation networks, unimportant nodes are matched with paper nodes involved in citation relationships.

3.3 Graph Kernel Search

To fully leverage information across different scales, we employ graph kernel to retrieve similar subgraphs at K scales with respect to the target subgraph g_t. Before being sent to the graph kernel for retrieval at each scale, g_t is first processed through the interaction module to obtain g_{inter}, which incorporates information from LLM and other scales. Details of the interaction module will be discussed in subsequent sections. We use random walk graph kernel for search, which measures the structural similarity between graphs by counting the number of common paths and evaluates feature similarity through node attributes:

$$g_r = KS(g_{inter}, \tilde{G}) \tag{5}$$

where $KS(\cdot)$ denotes the graph kernel retrieval function, and g_r represents the retrieved subgraph. Given g_{inter} and a subgraph $g' \epsilon \tilde{G}$, We use M_{sim}' to encode the similarity between the attributes of g_{inter} and g', while $A_\times$ denotes the adjacency matrix of the direct product graph. Then, we can compute the kernel that counts the number of common walks of length p between the two graphs as follows:

$$A_\times = A_{inter} \otimes A' \tag{6}$$

$$\mathrm{m} = \mathrm{vec}(M_{sim}') = \mathrm{vec}(H_{inter} X'^{\mathrm{T}}) \tag{7}$$

$$K_p(g_{inter}, g') = \sum_{i,j=1}^{|V_\times|} \mathrm{m}_i \mathrm{m}_j [{A_\times}^p]_{ij} = \mathrm{m}^T {A_\times}^p \mathrm{m} \tag{8}$$

where H_{inter} and X' represent the feature matrices of g_{inter} and g'. M_{sim}' denotes the node similarity matrix, where the (i,j)-th element represents the similarity between the i-th node in g_{inter} and the j-th node in g'. The (i,j)-th element of ${A_\times}^p$ represents the number of paths between the i-th and j-th nodes in the direct product graph. $K_p(g_{inter}, g')$ denotes the similarity score derived from the random walk graph kernel. Based on the similarity scores, we can obtain the retrieved subgraphs g_r at different scales, which will be used to enhance the output of the LLM in the form of CoT. (See Section 3.5 for details).

3.4 Multi-Scale Interaction Module

To facilitate comprehensive information flow between components, we introduce a multi-scale interaction module, which includes both bidirectional interaction and scale interaction components. As previously mentioned, the interaction module allows the target subgraph g_t to absorb additional information and obtain g_{inter}.

Bidirectional interaction. The bidirectional interaction module enables information flow between the graph model and the LLM, allowing their outputs to complement each other for enhanced collaboration. We utilize a PLM to embed the specific information generated by LLM at the current scale, denoted as e^k.

We integrate the information generated by the LLM for the target subgraph g_t into the message aggregation process of GNN. Traditional GNN models utilize a shared transformation matrix W_{intra} to aggregate information. However, since each target subgraph g_t contains distinct information, the globally shared matrix W_{intra} may fail to provide optimal feature aggregation for different target subgraphs. Therefore, we use W_{intra} to capture the common global structure of the graph, while incorporating personalized information e^k generated by the LLM to create specific intra-scale transformation matrix W_{intra}'. Specifically, we modulate the

shared weight W_{intra} by a scale-specific transformation through scaling and shifting. The localized weight matrix of g_t is given by

$$a = \sigma(M_a e^k) + 1, \quad b = \sigma(M_b e^k) \in \mathbb{R}^d \tag{9}$$

$$W'_{intra} = W_{intra} \odot [(a)_{\times d}] + [(b)_{\times d}] \tag{10}$$

where M_a and M_b are learnable parameters, Here the notation $[(X)_{\times n}]$ represents a matrix with n columns, all of which are identical to the vector x, and $\odot$ denotes element-wise multiplication.

Scale interaction. We employ the scale interaction module to ensure that the target subgraph g_t can fully utilize the graph topology at the current scale while also absorbing crucial information from the subgraph $g_r^{(k-1)}$ retrieved at the previous scale. By incorporating information from different scales, we can supplement the current scale with important details that may be overlooked.

For the information at the current scale, we use the topology of g_t for message propagation to highlight the importance of its structure. For cross-scale information, since different scales employ varying node aggregation ratios, discrepancies may arise between scales. So We compute the similarity between the target node at the current scale $v_t \in g_t$ and the nodes from the previous scale $V_r^{k-1} \in g_r^{k-1}$. We then select the top-k most similar nodes to establish connections with v_t, denoted as $Topk_{nodes(v_t)}$. Then we establish edges between v_t and $Topk_{nodes(v_t)}$, resulting in the cross-scale adjacency matrix A_{cross}, where i is the index of v_t, and j is the index of nodes in $Topk_{nodes}$. We utilize the GNN model to aggregate information from neighboring nodes from different scales using different transformation matrices:

$$\begin{aligned} H_{inter}^k = \sigma(&A_{intra} \cdot H_t^k \cdot W'_{intra} \\ &+ A_{cross} \cdot H_r^{k-1} \cdot W_{cross}) \end{aligned} \tag{11}$$

where A_{intra} and A_{cross} represent the adjacency matrix of the intra-scale and cross-scale, W'_{intra} and W_{cross} represent the intra and cross transformation matrices. H_t^k and H_r^{k-1} are the feature matrices of g_t and g_r^{k-1}

For the training of the interaction module, we employ the binary cross-entropy loss as the objective function:

$$L = -\frac{1}{N} \sum_{i=1}^{N} [y_i \cdot \log(p_i) + (1 - y_i) \cdot \log(1 - p_i)] \tag{12}$$

where N is the number of training data, y_i is the task-specific ground truth label.

3.5 CoT-based LLM Prediction Module

Through the multi-scale interaction module and graph kernel search module, We obtained subgraphs g_r retrieved at various scales and LLM-processed text related to the target graph. To fully leverage this information, we utilize the strong text comprehension capabilities of LLMs to process the rich topological information in the retrieval subgraph and make predictions about the target graph.

Specifically, we adopt the CoT method, which explicitly models the reasoning flow, enabling the LLM to gradually understand the multi-scale graph structure through predefined inference steps. We draw inspiration from the human learning process: Reflecting on the learning process, we first summarize the subject to gain a preliminary understanding. Based on this, we refine the specific definition. Through the study of relevant examples, our understanding is corrected and deepened, ultimately enabling a more accurate judgment of the subject's nature. We formulate CoT as four parts: Background understanding, Label scope definition, Example learning, and Context learning.

[Background understanding] The text information obtained from the first bidirectional interaction is used as background information.

[Label scope definition] Use the larger-scale retrieval subgraph to simulate the label scope of the target subgraph and utilize the text information obtained from the second bidirectional interaction to explain the emotional label.

[Example learning] Use the smaller-scale retrieval subgraph as an example to assist LLM reasoning.

[Context learning] Use the historical conversation of the target node as a supplement.

These prompts are fed into the LLM and enable the LLM to progressively learn both textual and structural information of the graph, leading to final inference. For the citation generation task, we use a similar approach. The prompt templates refer to Appendix E.

4 Experiments

4.1 Dataset

We conduct experiments on two tasks: emotion recognition in conversation (ERC) and citation generation. For the ERC task, we use IEMOCAP (Busso et al., 2008), MELD (Poria et al., 2018), and EmoryNLP (Zahiri and Choi, 2018) (only use the textual modality of the multi-modal dataset). For the citation generation task, we use GCite (Wang

Model Class	Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-L
Graph-based Methods	AutoCite	0.4696	0.3348	0.2315	0.1650	0.1700	0.0375	0.1334
	GAT	0.5131	0.3818	0.2684	0.1937	0.1548	0.0382	0.1339
	HGT	0.5252	0.3920	0.2758	0.1982	0.1555	0.0388	0.1359
	DisenCite	0.5418	0.4109	0.2951	0.2175	0.1756	0.0446	0.1515
Transformer-based Methods	SciGEN	0.4959	0.3975	0.2885	0.2110	0.1556	0.0102	0.1348
	PTGEN-Cross	0.3139	0.2343	0.1669	0.1234	0.1641	0.0417	0.1454
	Llama2	0.3884	0.3342	0.2599	0.2024	0.1866	0.0279	0.1200
	Llama3	0.5640	0.4612	0.3464	0.2647	0.2035	0.0283	0.1315
Graph-Transformer Methods	6+A+IF+E(Llama2)	0.4403	0.3758	0.2946	0.2338	0.2170	0.0529	0.1477
	6+A+IF+E(Llama3)	0.5947	0.4857	0.3683	0.2861	0.2214	0.0454	0.1503
	MSG-LLM(Llama3)	**0.6547**	**0.5407**	**0.4271**	**0.3504**	**0.2750**	**0.1238**	**0.2186**

Table 1: Experimental results of citation generation task.

et al., 2022). More details of implementations can be referred to Appendix A.

4.2 Baselines

For ERC task, the baselines are: **1) Graph-based Methods:** DialogueGCN (Ghosal et al., 2019), RGAT (Ishiwatari et al., 2020), DAG-ERC (Shen et al., 2021), DualGATs (Zhang et al., 2023a), SIGAT (Jia et al., 2023). **2) Transformer-based Methods:** KET (Zhong et al., 2019), SPCL+CL (Song et al., 2022), CoG-BART (Li et al., 2022), MPLP (Zhang et al., 2023b), InstructERC (Lei et al., 2023). **3) Graph-Transformer Methods:** S+PAGE (Liang et al., 2022), MKFM (Tu et al., 2023), MFAM (Hou et al., 2023). We choose the weighted-average F1 score as our evaluation metric.

For citation generation task, the baselines are: **1) Graph-based Methods:** GAT (Veličković et al., 2018), HGT (Hu et al., 2020), AutoCite (Wang et al., 2021), DisenCite (Wang et al., 2022). **2) Transformer-based Methods:** PTGEN-Cross (Xing et al., 2020), SciGEN (Luu et al., 2021). **3) Graph-Transformer Methods:** 6+A+IF+E (Şahinuç et al., 2024). We use widely adopted metrics BLEU-1/2/3/4 and ROUGE-1/2/L to measure the similarity between the generated context and the ground truth. For more details on the implementation, please refer to Appendix B.

4.3 Main Results

Table 2 illustrates the results of ERC task. Graph-based methods model conversations and speaker relationships through graph structures, such as DualGATs, which consider the complementary aspects of discourse structure and speaker-aware context. However, these methods do not fully utilize the rich textual information in conversations. Transformer-based methods, like InstructERC, capitalize on powerful text understanding capabilities by constructing instructions and fine-tuning LLMs, outperforming graph-based methods across three datasets. Graph-Transformer methods attempt to combine the strengths of both approaches. For instance, MFAM aligns the structural features of graphs with the semantic features of transformers, while MKFM enhances graphs using LLM-based data augmentation. However, these existing approaches simply combine the two, resulting in suboptimal performance compared to standalone graph-based or transformer-based methods. Our method fully leverages the powerful language understanding of LLMs while integrating multi-scale graph information, creating a more cohesive combination. Our method achieves improvements over the SOTA in IEMOCAP and EmoryNLP, and achieves performance close to InstructERC on MELD. Furthermore, when applied to mainstream large language models like Llama2-7b and Llama3-

Dataset	IEMOCAP	MELD	EmoryNLP
Models	W-F1	W-F1	W-F1
Graph-based Methods			
DialogueGCN	64.18	58.10	-
RGAT	65.22	60.91	34.42
DAG-ERC	68.03	63.65	39.02
DualGATs	67.68	66.90	40.69
SIGAT	70.17	66.18	39.95
Transformer-based Methods			
KET	59.56	58.18	34.39
SPCL+CL	69.74	66.35	40.25
CoG-BART	66.18	64.81	39.04
MPLP	66.65	66.51	-
InstructERC	71.39	**69.15**	41.37
Graph-Transformer Methods			
S+PAGE	68.93	64.67	40.05
MKFM	68.88	65.66	39.76
MFAM	70.16	66.65	41.06
Llama2	29.60	19.50	24.70
Llama2+MSG-LLM	41.73	31.57	31.96
Llama3	33.26	42.26	29.89
Llama3+MSG-LLM	46.60	44.08	31.40
Llama2+LoRA	40.82	58.73	35.82
Llama2+LoRA+MSG-LLM	**72.02**	69.14	**41.48**

Table 2: Experimental results of ERC task.

Task	ERC			Citation generation						
Dataset	IEMOCAP	MELD	EmoryNLP	GCite						
Models	W-F1	W-F1	W-F1	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-L
MSG-LLM	**72.02**	**69.14**	**41.48**	**0.6547**	**0.5407**	**0.4271**	**0.3504**	**0.2750**	**0.1238**	**0.2186**
w/o large-scale	68.01	68.36	38.89	0.6464	0.5324	0.4190	0.3424	0.2716	0.1160	0.2127
w/o small-scale	67.87	67.81	39.84	0.5756	0.4782	0.3783	0.3101	0.2715	0.1198	0.2120
w/o interaction(large)	67.84	67.50	38.85	0.6532	0.5394	0.4261	0.3496	0.2740	0.1215	0.2167
w/o interaction(small)	68.54	68.27	40.03	0.6461	0.5328	0.4195	0.3430	0.2685	0.1169	0.2119

Table 3: The ablation results of ERC and citation generation.

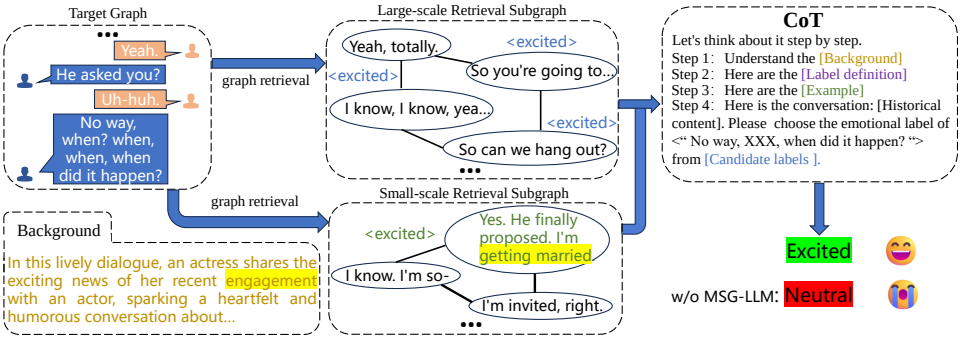

Figure 2: A case study of ERC task.

8b, our approach delivers further improvements, underscoring its effectiveness. The results of citation generation task are presented in Table 1. Similar to the results observed in ERC, our MSG-LLM also achieves state-of-the-art performance.

4.4 Ablation Studies

We conduct an ablation study to investigate the characteristics of the main components in our method. Tabel 3 shows the ablation results, and "w/o" denotes the model performance without a specific module. We have the following observations: The performance of our method drops when removing any one component, which suggests that every part of the design is necessary. Removing any large-scale retrieval subgraphs will cause a significant drop in model performance. This is consistent with our conjecture since large-scale subgraphs preserve their global integrity, which restores the true situation of the target subgraph. Taking away the small-scale retrieval subgraphs resulted in a steady decline on all three datasets, which indicates that small-scale subgraphs preserve the essential substructure details and are beneficial in assisting LLM in reasoning. Removing the interaction module causes obvious performance degradation, demonstrating the critical role of interaction module in enabling the target subgraph to absorb additional information from other scales and the LLM.

4.5 Analyses on Graph Structure

To further verify the capability of our model in effectively utilizing the graph structure, we conducted experiments on both ERC task and the citation generation task. We do not consider the graph structure, but only retrieve relevant sentences or papers based on semantic features and send them to LLM as described in the paper. The results are reported in Figure 3. Performance drop is observed across all datasets when the graph structure is not utilized. This highlights the model's ability to effectively leverage the graph structure.

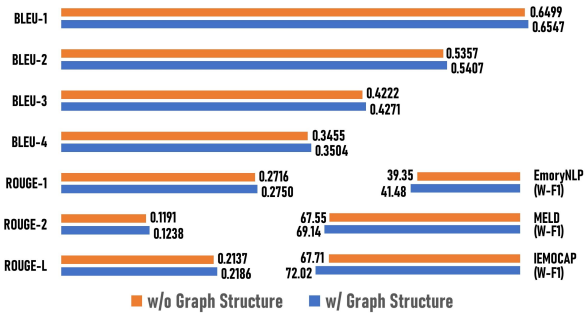

Figure 3: The analyses results of graph structure.

4.6 Case Study

As shown in Figure 2, the target is to judge the emotion of "No way, XXX, when did it happen?" It is difficult to literally judge from the sentence. MSG-LLM obtain two scales of retrieval subgraphs:

The larger-scale contains the conversation with the "hang out" topic, where the emotion is mostly "excited". The small-scale concentrates on the current conversation with the content of "married", so the emotion can be decided as "excited".

5 Conclusion

In this paper, we introduce a Multi-Scale Graph-Enhanced LLM (MSG-LLM). It captures similar graph structures and semantics across graphs in different scales. MSG-LLM first scales the large graph to generate scaled graphs and then employs multi-scale interaction module to enable the target subgraph to absorb information from different scales and LLMs. Finally, we employ graph kernels to retrieve similar subgraphs to the target subgraph at multiple scales and send the retrieved results to the LLM in the form of CoT. Experiment results show that our method achieves state-of-the-art performance on two graph-based downstream tasks.

6 Limation

One potential limitation of our approach is the use of a fixed scale variation parameter, which may not fully accommodate the dynamic or diverse nature of certain graphs. Future work could investigate adaptive scale variation techniques and explore more flexible similarity metrics to further improve the model's performance in varied contexts.

Acknowledgments

We thank all anonymous reviewers for their helpful comments and suggestions. This paper is supported by National Natural Science Foundation of China (NSFC Grant No. 62106275), the Guangdong Basic and Applied Basic Research Foundation under Grant 2023A1515011370, the National Natural Science Foundation of China (32371114), the Characteristic Innovation Projects of Guangdong Colleges and Universities (No. 2018KTSCX049).

References

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. 2008. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42:335–359.

Jingqiang Chen and Hai Zhuge. 2019. Automatic generation of related work through summarizing citations. *Concurrency and Computation: Practice and Experience*, 31(3):e4261.

Zhikai Chen, Haitao Mao, Hongzhi Wen, Haoyu Han, Wei Jin, Haiyang Zhang, Hui Liu, and Jiliang Tang. 2024. Label-free node classification on graphs with large language models (LLMs). In *The Twelfth International Conference on Learning Representations*.

Rajarshi Das, Manzil Zaheer, Dung Thai, Ameya Godbole, Ethan Perez, Jay-Yoon Lee, Lizhen Tan, Lazaros Polymenakos, and Andrew Mccallum. 2021. Case-based reasoning for natural language queries over knowledge bases. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9594–9611.

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

Deepanway Ghosal, Navonil Majumder, Soujanya Poria, Niyati Chhaya, and Alexander Gelbukh. 2019. Dialoguegcn: A graph convolutional neural network for emotion recognition in conversation. 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 154–164.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *arXiv preprint arXiv:2402.07630*.

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

Guiyang Hou, Yongliang Shen, Wenqi Zhang, Wei Xue, and Weiming Lu. 2023. Enhancing emotion recognition in conversation via multi-view feature alignment and memorization. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12651–12663.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Xixin Hu, Xuan Wu, Yiheng Shu, and Yuzhong Qu. 2022. Logical form generation via multi-task learning for complex question answering over knowledge bases. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1687–1696.

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.

Yuntong Hu, Zhihan Lei, Zheng Zhang, Bo Pan, Chen Ling, and Liang Zhao. 2024. Grag: Graph retrieval-augmented generation. *arXiv preprint arXiv:2405.16506*.

Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous graph transformer. In *Proceedings of the web conference 2020*, pages 2704–2710.

Xuanwen Huang, Kaiqiao Han, Yang Yang, Dezheng Bao, Quanjin Tao, Ziwei Chai, and Qi Zhu. 2024. Can gnn be good adapter for llms? In *Proceedings of the ACM on Web Conference 2024*, pages 893–904.

Taichi Ishiwatari, Yuki Yasuda, Taro Miyazaki, and Jun Goto. 2020. Relation-aware graph attention networks with relational position encodings for emotion recognition in conversations. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 7360–7370.

Zhaohong Jia, Yunwei Shi, Weifeng Liu, Zhenhua Huang, and Xiao Sun. 2023. Speaker-aware interactive graph attention network for emotion recognition in conversation. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(12):1–18.

Minki Kang, Jin Myung Kwak, Jinheon Baek, and Sung Ju Hwang. 2023. Knowledge graph-augmented language models for knowledge-grounded dialogue generation. *arXiv preprint arXiv:2305.18846*.

Jiho Kim, Sungjin Park, Yeonsu Kwon, Yohan Jo, James Thorne, and Edward Choi. 2023. Factkg: Fact verification via reasoning on knowledge graphs. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16190–16206.

Shanglin Lei, Guanting Dong, Xiaoping Wang, Keheng Wang, and Sirui Wang. 2023. Instructerc: Reforming emotion recognition in conversation with a retrieval multi-task llms framework. *arXiv preprint arXiv:2309.11911*.

Qian Li, Zhuo Chen, Cheng Ji, Shiqi Jiang, and Jianxin Li. 2024. Llm-based multi-level knowledge generation for few-shot knowledge graph completion. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 2135–2143. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Shimin Li, Hang Yan, and Xipeng Qiu. 2022. Contrast and generation make bart a good dialogue emotion recognizer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 11002–11010.

Chen Liang, Jing Xu, Yangkun Lin, Chong Yang, and Yongliang Wang. 2022. S+ page: A speaker and position-aware graph neural network model for emotion recognition in conversation. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 148–157.

Kelvin Luu, Xinyi Wu, Rik Koncel-Kedziorski, Kyle Lo, Isabel Cachola, and Noah A Smith. 2021. Explaining relationships between scientific documents. 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 2130–2144.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2018. Meld: A multimodal multi-party dataset for emotion recognition in conversations. *arXiv preprint arXiv:1810.02508*.

Furkan Şahinuç, Ilia Kuznetsov, Yufang Hou, and Iryna Gurevych. 2024. Systematic task exploration with LLMs: A study in citation text generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4832–4855, Bangkok, Thailand. Association for Computational Linguistics.

Weizhou Shen, Siyue Wu, Yunyi Yang, and Xiaojun Quan. 2021. Directed acyclic graph network for conversational emotion recognition. 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 1551–1560.

Xiaohui Song, Longtao Huang, Hui Xue, and Songlin Hu. 2022. Supervised prototypical contrastive learning for emotion recognition in conversation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5197–5206.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Heung-Yeung Shum, and Jian Guo. 2023. Think-on-graph: Deep and responsible reasoning of large language model with knowledge graph. *arXiv preprint arXiv:2307.07697*.

Yijun Tian, Huan Song, Zichen Wang, Haozhu Wang, Ziqing Hu, Fang Wang, Nitesh V Chawla, and Panpan Xu. 2024. Graph neural prompting with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19080–19088.

Geng Tu, Bin Liang, Bing Qin, Kam-Fai Wong, and Ruifeng Xu. 2023. An empirical study on multiple knowledge from chatgpt for emotion recognition in conversations. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12160–12173.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Qingqin Wang, Yun Xiong, Yao Zhang, Jiawei Zhang, and Yangyong Zhu. 2021. Autocite: Multi-modal representation fusion for contextual citation generation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 788–796.

Yifan Wang, Yiping Song, Shuai Li, Chaoran Cheng, Wei Ju, Ming Zhang, and Sheng Wang. 2022. Disencite: Graph-based disentangled representation learning for context-specific citation generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11449–11458.

Yu Wang, Nedim Lipka, Ryan A Rossi, Alexa Siu, Ruiyi Zhang, and Tyler Derr. 2024. Knowledge graph prompting for multi-document question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19206–19214.

Xinyu Xing, Xiaosheng Fan, and Xiaojun Wan. 2020. Automatic generation of citation texts in scholarly papers: A pilot study. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6181–6190.

Sayyed M Zahiri and Jinho D Choi. 2018. Emotion detection on tv show transcripts with sequence-based convolutional neural networks. In *Workshops at the thirty-second aaai conference on artificial intelligence*.

Duzhen Zhang, Feilong Chen, and Xiuyi Chen. 2023a. Dualgats: Dual graph attention networks for emotion recognition in conversations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7395–7408.

Ting Zhang, Zhuang Chen, Ming Zhong, and Tieyun Qian. 2023b. Mimicking the thinking process for emotion recognition in conversation with prompts and paraphrasing. *arXiv preprint arXiv:2306.06601*.

Zhengxuan Zhang, Yin Wu, Yuyu Luo, and Nan Tang. 2024. MAR: Matching-augmented reasoning for enhancing visual-based entity question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1520–1530, Miami, Florida, USA. Association for Computational Linguistics.

Peixiang Zhong, Di Wang, and Chunyan Miao. 2019. Knowledge-enriched transformer for emotion detection in textual conversations. 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 165–176.

A Dataset

For the ERC task, we evaluate the efficacy of our method on three standard benchmark datasets: IEMOCAP, MELD, and EmoryNLP. We utilize only the textual modality of the above datasets for the experiments. For the citation generation task, we evaluate the efficacy of our method on three standard benchmark datasets: GCite.

IEMOCAP (Busso et al., 2008) is a dataset recorded as dyadic conversational video clips with eight speakers participating in the training set while two speakers in the testing set.

MELD (Poria et al., 2018) is a multimodal dataset extended from the EmotionLines dataset. MELD is taken from the popular TV show Friends and contains more than 1,400 dialogues and more than 13,000 utterances, each of which is labeled with emotion and sentiment categories.

EmoryNLP (Zahiri and Choi, 2018) is a dataset also collected from the TV series Friends. The dataset comprises utterances that are categorized into seven distinct emotional classes.

GCite (Wang et al., 2022) is a dataset consisting of 25K relationships with different types (7.5K introduction, 8.0K related work, 4.9K model and 4.6K experiment citations) over 4.8K papers extracted from computer science domain of S2ORC (Lo et al. 2020). We randomly select 80% of citation relations to constitute the training set and treat the remaining 10%, 10% as the validation and test set respectively.

B Baselines

For the ERC task, we selected several SOTA baselines for each method:

Graph-based: DialogueGCN (Ghosal et al., 2019) leverage self and inter-speaker dependency of the interlocutors to model conversational context for emotion recognition and addresses context propagation issues present in the current RNN-based methods. **RGAT** (Ishiwatari et al., 2020) propose relational position encodings that provide RGAT with sequential information reflecting the relational graph structure, which can capture both the speaker dependency and the sequential information. **DAG-ERC** (Shen et al., 2021) combines the strengths of conventional graph-based neural models and recurrence-based neural models, providing a more intuitive way to model the information flow between long-distance conversational background and nearby context. **DualGATs** (Zhang

et al., 2023a) concurrently consider the complementary aspects of discourse structure and speaker-aware context, introduce DisGAT and SpkGAT to model discourse dependencies between utterances and capture speaker relationships. **SIGAT** (Jia et al., 2023) modeling the speaker-aware and sequence-aware information in a unified graph and updating them simultaneously to model the interactive influence of them and obtain the final representations.

Transformer-based: KET (Zhong et al., 2019) propose a Knowledge-Enriched Transformer, which interprets contextual utterances by hierarchical self-attention and using a context-aware affective graph attention mechanism to dynamically leveraged external commonsense knowledge. **SPCL+CL** (Song et al., 2022) propose a Supervised Prototypical Contrastive Learning (SPCL) loss to solve the imbalanced classification problem. It designs a difficulty-measure function based on the distance between classes and introduces curriculum learning to alleviate the impact of extreme samples. **CoG-BART** (Li et al., 2022) employs supervised contrastive learning along with an auxiliary response generation task to improve the model's ability to handle context information and better differentiate between similar emotions. **MPLP** (Zhang et al., 2023b) utilize a history-oriented prompt, an experience-oriented prompt, and the label paraphrasing mechanism to improve the understanding of the conversational context, the speaker's background, and the label semantics, respectively. **InstructERC** (Lei et al., 2023) develops instruction-based strategies for ERC and fine-tunes large language models.

Graph-Transformer Methods: S+PAGE (Liang et al., 2022) employs GNN to capture the speaker and position-aware conversation structure information, utilizing a two-stream conversational Transformer presented to extract both the self and inter-speaker contextual features for each utterance. **MFAM** (Tu et al., 2023) adopt supervised contrastive learning to align semantic-view and context-view features, these two views of features work together in a complementary manner, contributing to ERC from distinct perspectives. **MKFM** (Hou et al., 2023) leveraging large models to gain additional knowledge and propose a Multiple Knowledge Fusion Model to integrate knowledge generated by LLMs for ERC.

For the citation generation task, we selected several SOTA baselines for each method:

Graph-based: AutoCite (Wang et al., 2021) involves a novel multi-modal encoder and a multi-task decoder architecture. Based on the multi-modal inputs, the encoder in AutoCite learns paper representations with both citation network structure and textual contexts. **DisenCite** (Wang et al., 2022) propose a novel disentangled representation-based model DisenCite to automatically generate the citation text through integrating paper text and citation graph, empowering the generation of different types of citations for the same paper.

Transformer-based: PTGEN-Cross (Xing et al., 2020) first train an implicit citation text extraction model based on BERT and leverage the model to construct a large training dataset for the citation text generation task. Then it proposes and trains a multi-source pointer-generator network with a cross-attention mechanism for citation text generation. **SCIGEN** (Luu et al., 2021) address the task of explaining relationships between two scientific documents using natural language text.

Graph-Transformer Methods: 6+A+IF+E (Şahinuç et al., 2024) propose a three-component research framework that consists of systematic input manipulation, reference data, and output measurement and uses this framework to explore citation text generation.

C Graph Construction

For the ERC task, we built a graph based on the relationships between speakers. In this paper, the nodes are the utterances in the conversation, i.e., $V = \{u_1, u_2 \cdots u_N\}$, and for each utterance u_i, there is a previous utterance u_τ that is spoken by the same speaker as u_i. We establish edges between u_i and all sentences between u_i and u_τ.

For the citation generation task, following (Wang et al., 2022), we build a graph based on the citation relationship between papers.

D Analyses on Retrieval Efficiency

To verify that scale variation improves retrieval efficiency, we conducted experiments comparing retrieval times at different scales, as shown in Figure 4. The scale variation parameter K indicates the extent of scale variation, with larger values of K retaining more nodes, and smaller values of K resulting in fewer retained nodes. From figure 4, we observe that as K decreases, the time required for each retrieval also decreases. This demonstrates

that scale variation leads to improved retrieval efficiency.

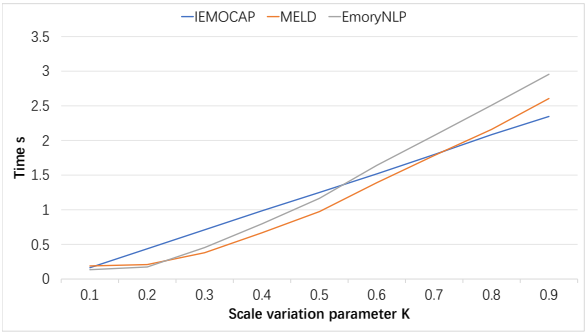

Figure 4: The analysis results of retrieval efficiency.

E Prompt Templates

For the ERC task, we formulate CoT as four parts: Background understanding, Label range definition, Example learning, and Context learning. For the specific CoT template, please refer to Figure 5.

For citation generation, we formulate CoT as three parts: Intention example learning, Citation example study and Context learning. For the specific CoT template, please refer to Figure 6.

F Fine-tuning Implementation Details

Since large language models(LLM) is a generative model and the ERC task is a discriminative task, in order to make LLM familiar with our downstream tasks, we designed an alignment task to fine-tune LLM. It mainly includes three parts: conversation background, label definition and historical content. The conversation background and label definition are generated by the LLM, and the historical window is set to 5.

We use Llama2-7B as our backbone model. Considering the efficiency and effectiveness of Parameter-Efficient-Fine-Tuning (PEFT), we adopt LoRA (Hu et al., 2021) and insert lowrank adapters after self-attention layers. We set the dimension of adapters to 16 and the learning rate to 2e-4. We train with BF16 precision on 80G Nvidia A100 GPUs.

G Citation Position Classification

To further demonstrate the generalization and effectiveness of our approach, following (Wang et al., 2022), we conducted an additional task (Citation Position Prediction) on the GCite dataset using GPT-4o. Table 4 illustrates the results. Given a pair of citing-cited nodes, the goal is to predict which sections the citation could exist. We evaluate the performance using both Micro F1 and Macro F1 scores. Experimental results show that our method significantly outperforms GPT-4o, with improvements of +6.17% in Micro-F1 and +8.1% in Macro-F1% at the 2-scale level, and +7.93% in Micro-F1 and +19.57% in Macro-F1 at the 3-scale level, compared to GPT-4o without MSG-LLM.

Method	Micro-F1	Macro-F1
GPT-4o	0.4075	0.2721
GPT-4o+MSG-LLM(2 scales)	0.4692	0.3531
GPT-4o+MSG-LLM(3 scales)	0.4868	0.4678

Table 4: Position prediction performance.

[SYSTEM]
You are an expert in emotional psychology and you can accurately assess people's emotional states.
[CoT]
Assuming you are in the scene of [Background]:<background>.
Let's think about it step by step:
Step 1: Understand the [Background]
Step 2: Here are the [Label definition]:<label definition>.
Step 3: Here are the [Example]:<example>.
Step 4: Here is the conversation which involves several speakers: <historical content>.
Please choose the emotional label of <target utterance> from <candicate labels>. Just give me the label.

Figure 5: Prompt template of ERC.

[SYSTEM]
As a master in academic writing, capable of digesting the information provided, imagine that dst (the citing paper) cites src (the cited paper) at a specific location in the paper (e.g., intro, method, related work, experiment). Please write a citation sentence for dst.
[CoT]
Let's think about it step by step:
Step 1: For the following examples of citation sentences [Comparative Examples], they all come from the same citation position <citation position>.Please focus only on the intentions behind the citation sentences and completely ignore the content itself (as the content itself is not what we need).[Comparative Examples]: <Comparative Citations>
Step 2: After understanding the citation's position and intent, refer to a specific example [citation example] (where cite_label indicates the citation position, and cite_text is the specific citation text). [citation example]: <citation example>.
Step 3: Fully comprehend [citation position] and [citation example], integrate the following content [target content] (especially focusing on <section list >) and write a professional citation for dst (paying attention to citation length and quality), [target content]: <target content>.
Please generate only the 'citation sentence'.

Figure 6: Prompt template of citation generation.