MMGraphRAG: Truly Understanding a Document via Multimodal Knowledge Graphs

Xueyao Wan
Harbin Institute of Technology
wanxueyao123@qq.com

Abstract

This paper proposes a novel Multi-Modal GraphRAG method that combines structured knowledge graphs with multi-modal data to enhance reasoning and interpretability in generative tasks. Unlike existing RAG systems, our approach integrates text and image information into a fine-grained and more structured multi-modal knowledge graph. We introduce scene graphs to refine image data by decomposing it into entities, categories, and relationships, significantly enhancing the model's reasoning ability and providing stronger interpretability of the generated responses. Additionally, the cross-modal fusion module employs a spectral clustering-based strategy to generate candidate entities, improving fusion accuracy and reducing computational overhead. Experiments on multi-modal document question-answering tasks demonstrate that our method outperforms existing approaches in multi-domain and multi-page reasoning, offering a promising solution for complex reasoning tasks. The project code is available at https://github.com/wanxueyao/MMGraphRAG.

1. Introduction

With the development of multi-modal technology, the field of Retrieval-Augmented Generation (RAG) has begun exploring how to handle inputs from different modalities and improve generation quality through multi-modal information fusion. Existing multi-modal RAG systems typically perform retrieval and generation by embedding data from different modalities into a shared space [1], such as M3DOCRAG [14] and VisRAG [72]. These methods perform well in implicit fact queries (RAG level 2) [76]. However, their performance is limited in more complex reasoning tasks, such as interpretable rationale queries and hidden rationale queries (RAG levels 3 and 4), due to the lack of modeling structured relationships in the data.

Embedding-based retrieval can only measure inter-modal similarity, making it difficult to capture complex semantic relationships and cross-modal logic. Moreover, generation models that rely on fuzzy matching struggle to provide clear reasoning paths and high-quality interpretable outputs. Therefore, multi-modal RAG systems that rely solely on embedding spaces cannot meet the demands of complex tasks.

Microsoft's GraphRAG [17], which integrates structured knowledge graph information, significantly enhances the logical coherence and interpretability of text-based generation tasks, highlighting the potential of knowledge graphs in the RAG field. However, the current implementation of GraphRAG primarily focuses on text data. Building on this foundation, we propose a comprehensive Multi-Modal GraphRAG (MMGraphRAG) framework aimed at enhancing performance in complex reasoning tasks, including interpretable rationale queries and hidden rationale queries. As the first MMGraphRAG pipeline, our method overcomes the limitations of current approaches in cross-modal information fusion and offers an extensible reference framework for multi-modal RAG by conducting fine-grained structural modeling of both text and image data.

One of the core innovations of this paper is refining the image modality using scene graphs[7, 33], which are then integrated into the multi-modal knowledge graph(MMKG). Through scene graph construction, the information of image modality is explicitly decomposed into entities, categories, and relationships, forming structured MMKG. This fine-grained image modeling improves logical reasoning and enhances interpretability by clearly presenting the reasoning path through the scene graph.

Additionally, we propose an innovative cross-modal fusion module that integrates data from different modalities into a unified MMKG. Unlike traditional methods that rely on training data, large language model(LLM)-based methods, such as UniMEL [41], improve accuracy but increase computational costs. To address this, we constructed a cross-modal entity alignment dataset and propose a spectral clustering-based candidate entity generation method, which reduces the number of model invocations while maintaining high accuracy.

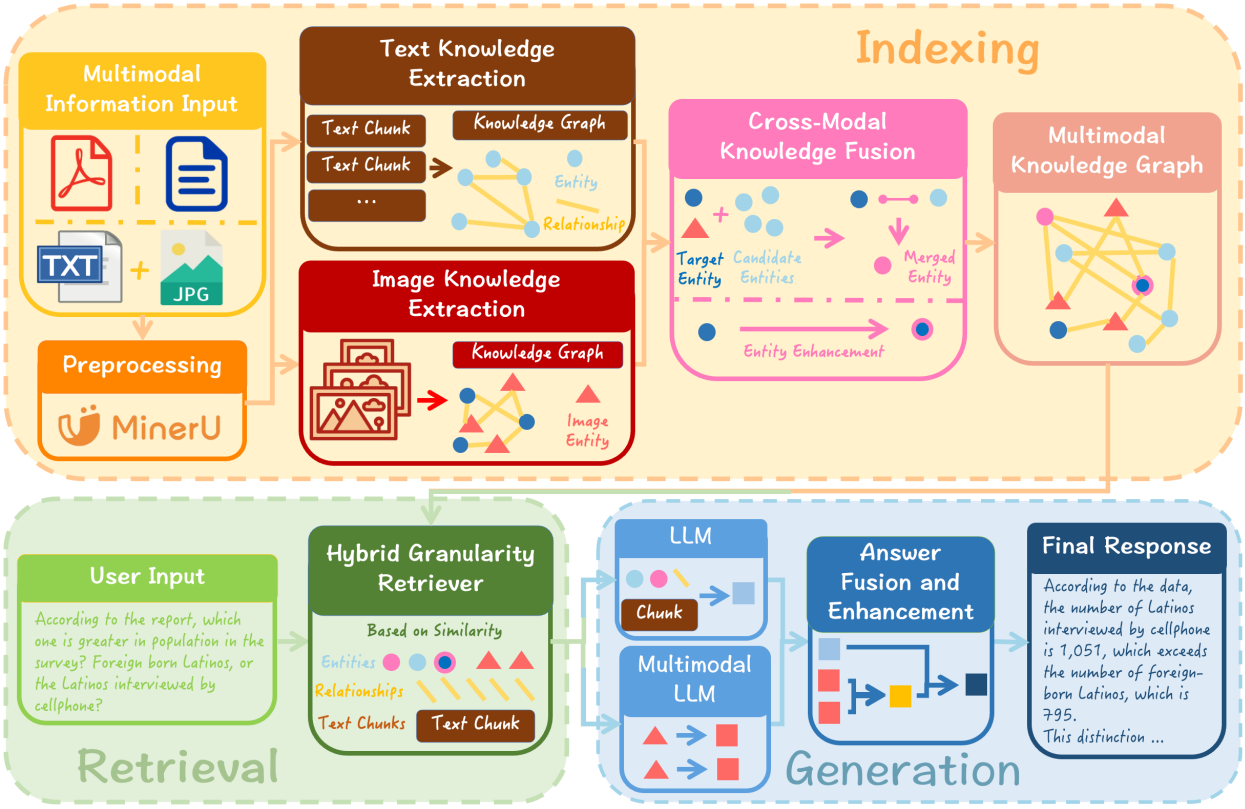

Figure 1. MMGraphRAG Pipeline Overview. This diagram illustrates the comprehensive workflow of the MMGraphRAG pipeline, starting with the input of multi-modal information. In the Text Knowledge Extraction Module, textual data is extracted and transformed into a knowledge graph, while the Image Knowledge Extraction Module processes image data into a scene graph. The Cross-Modal Knowledge Fusion step integrates text and image entities into a unified MMKG, where entities are further enhanced through the fusion process. The Hybrid Granularity Retriever enables precise retrieval based on similarity, further refining the model's ability to handle complex queries. Finally, the LLM-based generation process synthesizes the retrieved multi-modal data into a coherent and enhanced response.

This paper employs multi-modal document Question-Answering (QA) tasks as the specific evaluation criterion. Our method outperforms existing RAG approaches on the DocBench[80] and MMLongBench[46] datasets, highlighting its advantages in reasoning ability and interpretability.

2. MMGraphRAG Pipeline

In this section, we introduce the overall framework of MMGraphRAG, which builds upon the typical GraphRAG pipeline. For a detailed review of the GraphRAG pipeline, please refer to Supplementary Material A.

To implement the MMGraphRAG pipeline, we adopt a modular pipeline design approach (visualized in Figure 1), splitting the main modules—Indexing, Retrieval, and Generation—into independent and interchangeable submodules, thus achieving high flexibility and scalability for the system [19, 49]. The detailed design and implementation of each module are as follows:

Indexing. This module converts multi-modal information into a structured MMKG, supporting efficient retrieval and generation. It includes three submodules:

- **Preprocessing Module:** Tools like MinerU[61] are used to parse multi-modal inputs (such as PDF documents), into standardized formats, separating text and image data. A more detailed description of the entire preprocessing module can be found in the Supplementary Material B.
- **Single-Modal Processing Module:** This module processes textual and image information independently. Textual data is chunked and entity extraction is performed to generate a structured text knowledge graph [31, 59, 64, 68, 79]; image data is processed through image segmentation [26, 52], scene graph construction, and entity alignment techniques to form an image knowledge graph.
- **Cross-Modal Fusion Module:** A spectral clustering method is used to filter candidate entities for fusion, aligning and integrating text and image information into a unified MMKG. Meanwhile, multi-modal information is used to enhance entities, improving the semantic expressiveness of the knowledge graph.

Finally, the indexing module outputs a highly structured,

semantically rich MMKG, providing a solid foundation for the retrieval module. The specific structure and storage method of the MMKG can be found in Supplementary Material C.

Retrieval. This module extracts relevant entities, relationships, and contextual information from the MMKG based on the user query [21].

Generation. This module generates the final answer based on the retrieval results and the user query. It combines single-modal and multi-modal generation strategies to ensure high accuracy and quality, leveraging multi-modal data to enrich the answer content. For detailed implementation, please refer to Supplementary Material D.

3. Cross-Modal Fusion

In MMKGs, the core task of cross-modal fusion is to align the entities extracted from images and text, identifying and merging entities that refer to the same thing. This task is similar to entity alignment and entity linking. Traditional methods rely heavily on large amounts of annotated data and external features, which leads to poor performance in few-shot scenarios, domain transfer, or with new entities [18, 43, 47, 66, 75]. These methods often use coarse-grained techniques (e.g., TF-IDF [50, 51], word2vec [24]) and text encoders (e.g., LSTM [38], BERT [58]) to measure similarity.

Recent methods, such as multi-modal entity alignment (MMEA) [11, 13, 78] and visual pivoting [40], also face similar challenges. MMEA maps visual, textual and numerical modalities to a common space and balances modal contributions, but it may lead to semantic ambiguity and inaccurate alignment when modal interactions are insufficient[10]. Visual pivoting methods drive entity alignment with visual information from images, offering stronger adaptability but still showing limited performance when labeled data is insufficient or in new domains.

With the development of LLM technology, methods like UniMEL [41] and OneNet [42] have demonstrated new potential by using LLMs to generate high-quality descriptions or multi-perspective judgments to optimize entity alignment. However, frequent LLM calls result in significant computational overhead, especially when each new entity requires a separate call, severely limiting the efficiency of these methods.

To address these limitations, we constructed a fine-grained Cross-Modal Entity Alignment (CMEA) dataset. Unlike existing datasets (e.g., FB15K-DB15K, FB15K-YAGO15K [6, 10], Twitter-MEL [2, 3]), which focus on coarse-grained alignment between images and text, our CMEA dataset provides fine-grained annotations for internal image entities. This enables more accurate evaluation of various fusion methods and promotes the development of the field.

3.1. Cross-Modal Entity Alignment Dataset

This dataset encompasses documents from three different domains: news, academia, and novels. Each document has been constructed into a MMKG, covering text-only knowledge graphs (stored by text chunk), image knowledge graphs based on scene graphs constructed from each image, and the original documents in PDF format. The specific number of entities is illustrated in Figure 2. Detailed information regarding the construction process can be found in Supplementary Material E.

The dataset is designed to support three tasks, as detailed below:

- **Task 1:** Extract the entity corresponding to the entire image and identify the related entity from the text chunk knowledge graph.
- **Task 2:** Extract the text entities involved in the image and map them to the text chunk entities.
- **Task 3:** Extract the entities corresponding to the entire image and extract the related entities from the raw text.

The quantity and corresponding categories of each task are presented in Table 1.

Tasks	Type			Total.
	News	Aca.	Nov.	
Task 1	195	266	235	696
Task 2	87	475	552	1114
Task 3	195	266	235	696

Table 1. Distribution of Tasks in the CMEA Dataset. The quantities for Task 1 and Task 3 are consistent with the total number of document images.

Dataset Evaluation Methods. Tasks 1 and 3 share the same evaluation method, which includes three criteria: 1) Exact match between the extracted and target entity is considered correct; 2) If one entity fully contains the other, it is also correct; 3) If character matching exceeds 50%, it counts as correct. Task 2 has stricter criteria, requiring exact matches between text entities in the image and those in the text chunk. To evaluate model performance comprehensively, we employ both micro-accuracy and macro-accuracy. The calculation formulas are provided in Supplementary Material E.1.

3.2. Research on Fusion Methods

In the cross-modal entity alignment task, we explore various methods and quantitatively evaluate their performance using the constructed CMEA dataset, aiming to provide better entity alignment solutions for practical applications. In the experiments, we mainly use three methods for candidate entity generation: the embedding-based similarity calculation method, the LLM-based method, and the clustering-based method. The detailed implementations of the first two methods are provided in Supplementary Material F.1.

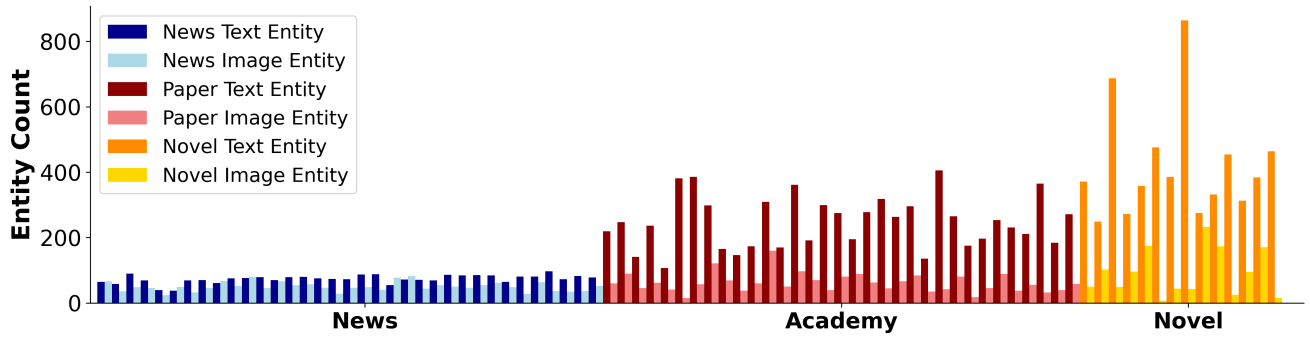

Figure 2. Entity Distribution Across Document Domains. The y-axis represents the entity count, while the x-axis categorizes the document domains. In each group of bar charts, the left side shows the text entities, and the right side shows the image entities. The greater the number of entities, the more difficult the alignment. The novel domain has the highest number of entities and the greatest difficulty, followed by the academia domain. The news domain has the fewest entities and the least difficulty.

Clustering-Based Method: To address the computational intensity and imprecision of LLM-based methods in fine-grained alignment, we propose a clustering-based method to optimize the entity alignment process. The core idea of the clustering method is to group semantically similar text entities into the same cluster, thereby reducing the number of candidate entities and lowering computational costs. The clustering process consists of two steps: first, clustering the text entities, and then selecting the most appropriate cluster based on the image entity to determine the candidate entities.

Entity clustering methods in knowledge graphs can generally be divided into two types:

- **Distance-Based Clustering Methods:** These methods use classical clustering algorithms like KMeans [28, 37, 53] or DBSCAN [16, 27] based on the similarity between text entity descriptions. Although these methods effectively utilize semantic information from entity descriptions, they focus only on similarity in a single dimension and ignore the relationships between entities.
- **Relationship-Based Clustering Methods:** Unlike distance-based clustering methods, relationship-based clustering methods reveal deeper associations by constructing a relationship graph between entities (e.g., using algorithms like Pagerank [55] or Leiden [57]). By analyzing the connectivity between entities (e.g., computing node in-degrees), these methods can capture latent relationships, thus providing a more comprehensive understanding of the entire graph. However, they are highly dependent on the graph structure, and their performance may suffer in sparse or complex relationship graphs.

Spectral Clustering Method: Combining Distance and Relationship Information. To overcome the limitations of both methods and more effectively utilize the information in the knowledge graph, we introduce the spectral clustering method [25, 48, 60]. The distinctive feature of spectral clustering is that it can simultaneously leverage both distance and relationship information, thereby improving clustering accuracy and robustness.

Spectral clustering reduces dimensionality by calculating the Laplacian matrix of the graph, which can handle complex graph structures and non-convex clusters (avoiding the issue in KMeans where the number of clusters must be preset).

Specifically, we redesign the weighted adjacency matrix A and degree matrix D.

The adjacency matrix A is constructed to reflect the similarity between nodes, incorporating the importance of relationships between nodes. Suppose the relationship between node i and node j is given by a weight $weight(r_{ij})$; if no explicit relationship exists, the weight is set to 1.

Thus, the elements of the adjacency matrix A can be defined as:

$$A_{ij} = sim(\mathbf{v}_i, \mathbf{v}j) \cdot weight(r_{ij}) \tag{1}$$

where $sim(\mathbf{v}i, \mathbf{v}j)$ is the cosine similarity between nodes i and j.

The degree matrix D is a diagonal matrix, where each diagonal element D_{ii} represents the degree of node i. This degree indicates the strength of node i's connections to other nodes and is calculated as $D_{ii} = \sum_j A_{ij}$. In simpler terms, the degree of each node in the degree matrix D is the sum of its similarity to all other nodes.

Then, the Laplacian matrix $L = D - A$ is calculated, followed by eigen-decomposition to obtain the top-k smallest eigenvalues and their corresponding eigenvectors, resulting in the matrix Q. DBSCAN is used to cluster matrix Q. Finally, assign the most appropriate cluster to the target image entity to complete the candidate entity generation. The remaining details of the spectral clustering computation are presented in Supplementary Material F.2.

3.3. Experiments on Fusion Methods

To assess the effectiveness of various fusion methods, we conducted a series of experiments based on the CMEA dataset. To ensure the broad applicability of the experimental results, we selected diverse models for testing. Detailed experimental setups and complete results can be found in Supplementary Material G.

We used DBSCAN (DB), KMeans (KM), Pagerank (PR), and Leiden (Lei) algorithms as baselines for clustering methods. The embedding model used was stella-en-1.5B-v5 [74] (Stella), the single-modal LLM was Qwen2.5-72B-Instruct [69], and the multi-modal LLM was InternVL2.5-38B-MPO [12]. For cluster selection, we utilized KNN [20] and LLM-based method; the results presented use the LLM-based approach.

Method	Task1(micro/macro Acc.)			Overall.
	News	Aca.	Nov.	
Emb	59.49/60.92	35.71/34.04	48.94/48.68	46.84/48.01
LLM	62.05/62.76	36.84/37.00	56.60/56.90	50.57/51.39
Clustering-based Methods(Ours)				
DB	52.31/53.64	39.10/39.99	60.85/60.03	50.14/49.24
KM	60.51/61.73	**45.49/44.95**	55.74/55.33	**53.16/53.88**
PR	52.31/53.39	37.59/38.11	54.04/52.12	47.27/47.02
Lei	**61.03/61.98**	41.73/41.86	57.87/53.74	52.59/52.47
Spec	54.36/55.71	34.21/34.94	**62.55/60.38**	49.43/48.15

Table 2. Task 1 Results

The results for Task 1 are shown in Table 2. Although clustering-based methods generally perform better, the difference is not substantial. Task 1 treats the entire image as an entity, which is a coarse-grained alignment, allowing similarity calculation and LLM methods to achieve comparable results.

Method	Task2(micro/macro Acc.)			Overall.
	News	Aca.	Nov.	
Emb	10.75/8.43	33.10/34.46	8.95/7.54	20.00/16.81
LLM	33.33/24.14	36.82/36.06	17.40/20.83	27.12/27.01
Clustering-based Methods(Ours)				
DB	53.76/45.90	60.84/58.33	29.90/34.20	**45.17/46.14**
KM	50.54/40.62	60.71/57.74	29.56/30.52	45.20/42.96
PR	51.61/44.43	59.70/56.83	29.05/35.16	44.10/45.47
Lei	54.84/44.67	60.46/55.49	29.39/30.61	44.84/43.59
Spec	**65.52/56.90**	**73.26/69.91**	**31.16/39.41**	**51.80/59.15**

Table 3. Task 2 Results

The results for Task 2 are shown in Table 3. In fine-grained alignment, clustering-based methods significantly outperform others, with spectral clustering showing superior results. Overall accuracy improves by approximately 15% in micro-accuracy and 30% in macro-accuracy compared to other algorithms. This indicates that the spectral clustering-based candidate entity generation method is more effective for fine-grained cross-modal entity alignment tasks.

4. Indexing Module

This section offers an in-depth look at two critical components of the Indexing module within the MMGraphRAG pipeline: the Img2Graph module (Section 4.1) and the Fusion module (Section 4.2).

4.1. Img2Graph Module

Implementing MMGraphRAG requires entity-level granularity for image data, making the construction of accurate and comprehensive scene graphs a crucial step. Traditional methods often neglect the fine-grained semantic details of objects themselves and fail to adequately consider the interaction between objects and their surrounding environment. This oversight may lead to substantial inaccuracies and a lack of robustness, especially when confronted with complex scenes[15, 23, 29, 34, 35, 70].

On the other hand, methods based on multi-modal LLMs leverage refined semantic segmentation and content generation. These methods can extract both explicit and implicit relationships within images and align images with the text modality, enabling the generation of high-precision, fine-grained scene graphs [8, 62]. For instance, they can infer complex implicit relationships (e.g., "boy—boy and girl appear to have a good relationship, possibly friends or romantic partners—girl") beyond explicit spatial ones (e.g., "girl—girl holding camera—camera").

Moreover, these methods provide detailed entity descriptions, offering richer semantic information like "a fatigued college student" instead of simple terms like "boy". Most importantly, they eliminate the reliance on large amounts of annotated training data [39, 54, 56, 67, 73, 77]. By refining image feature block descriptions and alignment, the scene graph accurately expresses inherent relationships, generating more interpretable graphs capable of handling a broader range of scenarios.

The Img2Graph module achieves the mapping from image to knowledge graph in five steps (for more detailed steps, please refer to Supplementary Material H). The first step is semantic segmentation of the image (by default, using YOLO v8 [26] to parse the image into regions with independent semantics, known as image feature blocks); the second step involves using multi-modal LLMs to generate descriptions of the feature blocks; the third step extracts entities and their relationships from the image. The fourth step uses the recognition and reasoning capabilities of the multi-

modal LLMs to align the feature blocks with the extracted entities; the fifth step is to construct global entity for the entire image, supplement its overall description, and enhance its connection with other local entities. A specific example is shown in Figure 3.

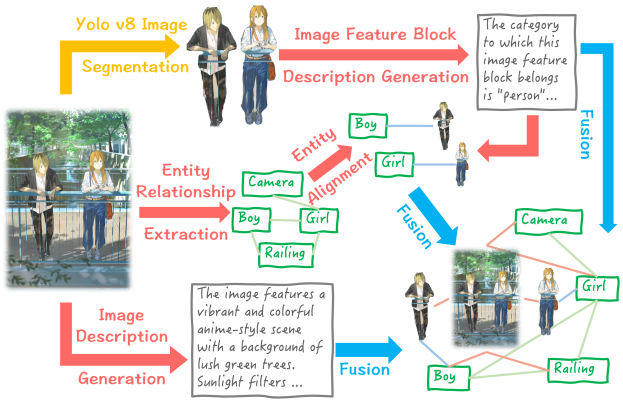

Figure 3. An Example of the Img2Graph Module in Action

4.2. Fusion Module

The Fusion module is designed to align, enhance, update, and merge the image knowledge graph with the text knowledge graph, aiming to construct a comprehensive MMKG. More details can be found in Supplementary Material I.

Aligning the Image Knowledge Graph. Begin by extracting descriptive information from the entities in the image, filtering out unnecessary entity types, particularly the image feature block entities. Using the descriptions of image entities along with the entities and relationships in the text knowledge graph, spectral clustering is applied to identify candidate entities. Next, we leverage LLMs to align the image entities. Finally, we can locate related entities within the text knowledge graph and generate corresponding records.(CMEA Dataset Task 2)

Enhancing the Image Knowledge Graph. Although the spectral clustering-based fusion method yields favorable results, there are cases where information may be lost or entities cannot be directly aligned. At this point, the description of image entities can be further enriched by incorporating relevant text entity information from the original text.

For example, the image entity "Flooded Neighborhood" alone cannot provide information about the affected location and cause. However, two related text entities, "HURRICANE IAN" and "FLORIDA," are mentioned in the original text. With these two entities and the information from the original text, the description of the entity can be enhanced to create a more detailed description: "Flooded Neighborhood is an area in Florida affected by Hurricane Ian, which caused significant flooding and destruction, leaving many without habitable homes." This enhancement process is crucial for the completeness of the information.

To implement this enhancement process, we follow these steps: First, based on the records, we filter out the text and image entities that require alignment and enhance the remaining image entities. Update the nodes in the knowledge graph, enhance the descriptions of entities using relevant entities and chunks, and store the enhanced knowledge graph as a new file.

Updating the Image Knowledge Graph. Use the enhanced image entities and the text knowledge graph to update the nodes and edges of the image knowledge graph. If a matching entity is found, new relationships will be added to the graph; if not, new nodes will be created, and relationships will be established. This step creates connections between the entire image and the text knowledge graph. (CMEA Dataset Task 1)

Merging the Graphs. Merge the updated image graph with the original text graph. Based on the aligned entities, merge the nodes and edges in both graphs. Remove duplicates and connect related entities to ensure integrity and consistency.

Executing Fusion. For each image, execute the above steps.

5. Evaluation

We selected DocBench[80] and MMLongBench[46] datasets as benchmarks to validate the MMGraphRAG pipeline, given their unique characteristics that align well with our evaluation goals. The DocBench dataset evaluates the reasoning process behind answers, which is crucial for assessing the system's ability to handle interpretable rationale queries. In contrast, MMLongBench focuses on the understanding of long documents and cross-page reasoning, challenging the system's ability to integrate evidence across complex document structures. In particular, MMLongBench's emphasis on cross-page reasoning and long-form document understanding is especially valuable for evaluating the system's performance on hidden rationale queries, where the supporting evidence may be scattered across multiple sections. These features of the datasets make them ideal for evaluating the robustness and effectiveness of the MMGraphRAG pipeline in multi-modal document QA tasks.

5.1. Benchmarks

DocBench is a benchmark designed to generate corresponding answers from raw PDF files and related questions. The dataset contains 229 PDF documents from publicly available online resources, covering five domains: academia (Aca.), finance (Fin.), government (Gov.), laws (Law.), and news (News). It includes four types of questions: pure text questions (Txt.), multi-modal questions (Mm.), metadata questions, and unanswerable questions (Una.). For our

experiments, since the information is converted into knowledge graphs, we do not focus on metadata. Therefore, this category of questions is excluded from the statistics, and the total number of valid questions is 844. For evaluation, DocBench determines the correctness of answers using LLM (Llama3.1-70B-Instruct in the experiments), assigning a score of 1 for correct answers and 0 for incorrect answers. Accuracy is then used to assess system performance, and accuracy is computed for all instances, individual domains, and different question types.

MMLongBench is a benchmark for evaluating the understanding of long documents, consisting of 135 long PDF documents from seven different domains, with a total of 1,091 manually annotated questions. The average length of each document is 47.5 pages. This dataset covers not only text(Txt.) but also various evidence sources such as charts, tables(C.T.), layout(Lay.), and images(Fig.). MMLongBench follows the three-step evaluation protocol of MATHVISTA[44]: response generation, answer extraction using LLM, and score calculation. Accuracy and F1 scores are reported to balance the evaluation of answerable and unanswerable questions, using Llama3.1-70B-Instruct throughout.

5.2. Baselines

To ensure fairness in the experiments and eliminate potential errors that may arise from using the same model for both evaluation and generating responses, this experiment selects various single-modal and multi-modal LLMs as comparison benchmarks. Here, only two models from each category are recorded, including the single-modal models Llama3.1-70B-Instruct[4] (L), Qwen2.5-72B-Instruct[69] (Q), and the multi-modal models Qwen2-VL-72B[63] (Qvl), InternVL2.5-38B-MPO[12] (Intvl). For more comprehensive results, please refer to Supplementary Material J.

For effective comparison, we employed the following methods:

LLM. All the text and questions from the Markdown files processed by Mineru[61] are provided to the LLM for question answering. If the text and questions exceed the model's context length limitation, the questions will be asked in batches, and the multiple answers will be concatenated together.

MMLLM. All images are concatenated together and scaled according to the model's input limit. The concatenated images are provided along with the questions to the multi-modal LLM for answering, in order to assess its ability to handle multi-modal information.

NaiveRAG[32]. The text in the Markdown files is chunked into fixed token sizes (500 tokens per chunk). Each chunk and question is embedded. Then, the top-k relevant chunks(5 selected) are retrieved by calculating cosine similarity with the question, and these relevant chunks are provided along with the question to LLMs for the generation task.

GraphRAG. Adapted from Microsoft GraphRAG[17, 21], this method was modified by removing the community detection part and using local mode querying to ensure fair comparison with other methods. The top-k entities (10 selected) are used for retrieval, with the entity and relationship text limited to a maximum of 4000 tokens, and the maximum chunk length of text is 6000 tokens. Other parameters are consistent with those of the MMGraphRAG.

Model	Type					Domain			Overall Acc.
	Aca.	Fin.	Gov.	Law.	News	Txt.	Mm.	Una.	
LLM-based Methods									
Llama	43.9	13.5	53.4	44.5	79.7	52.9	18.8	**81.5**	44.7
Qwen	41.3	16.3	50.7	49.7	77.3	53.9	20.1	75.8	44.8
MMLLM-based Methods									
Qvl	17.5	14.9	25.0	34.6	48.8	34.0	8.4	40.3	25.4
Intvl	19.8	16.3	28.4	31.4	46.5	35.7	15.9	39.5	27.7
NaiveRAG-based Methods									
Llama	43.6	38.2	66.2	64.9	**80.2**	79.9	32.1	70.2	61.0
Qwen	43.6	34.4	62.8	65.4	75.0	81.6	30.5	67.7	59.5
GraphRAG-based Methods									
Llama	40.6	27.1	56.8	59.7	75.0	73.5	24.4	76.6	54.7
Qwen	39.6	25.7	52.5	49.7	74.5	71.7	26.0	67.5	52.3
MMGraphRAG-based Methods (Ours)									
L-Qvl	51.8	59.4	62.8	60.7	77.9	79.1	77.8	70.2	74.0
Q-Qvl	51.8	62.9	**66.9**	68.6	76.2	**82.4**	81.1	67.7	75.2
L-Intvl	**60.7**	64.1	62.6	64.9	76.2	80.0	86.4	75.0	**77.5**
Q-Intvl	60.5	**65.8**	66.5	**70.4**	77.1	81.2	**88.7**	71.9	76.8

Table 4. Docbench Dataset Results. Although the News category contains many images, the DocBench dataset does not include multi-modal questions for this domain. As a result, the accuracy of methods other than MMLLM-based methods is relatively close in the News domain.

5.3. Comparisons

The results of the DocBench dataset are shown in Table 4, and the results of the MMLongBench dataset are shown in Table 5. We compared multiple benchmark methods and highlighted the advantages of MMGraphRAG over other methods. The following analysis from several key perspectives explains the superiority of MMGraphRAG, along with an explanation based on its principles.

Multi-Modal Information Processing Capability. The results demonstrate that relying solely on multi-modal LLMs does not necessarily improve the answering of image-related questions effectively, compared to other methods. In fact, in some cases, answers based on image-related text perform better (such as NaiveRAG). In contrast,

Model	Locations			Modalities				Overall	
	Sin.	Mul.	Una.	C.T.	Txt.	Lay.	Fig.	Acc.	F1
LLM-based Methods									
Llama	23.7	20.3	51.6	14.5	32.1	21.9	17.4	28.2	23.0
Qwen	22.5	20.0	53.2	14.7	33.3	23.5	16.1	27.8	22.1
MMLLM-based Methods									
Qvl	10.8	9.9	8.1	6.5	10.0	8.8	12.7	10.0	9.5
Intvl	13.3	7.9	13.9	8.4	10.7	11.8	11.4	11.6	10.4
NaiveRAG-based Methods									
Llama	24.9	19.5	56.5	15.8	31.0	22.7	18.7	29.2	24.2
Qwen	22.3	16.4	52.5	13.0	30.3	20.3	14.9	26.2	20.9
GraphRAG-based Methods									
Llama	16.3	12.3	**78.5**	7.2	25.1	15.0	10.6	27.2	18.2
Qwen	18.2	13.2	77.1	12.5	26.1	12.2	8.5	28.1	19.3
MMGraphRAG-based Methods (Ours)									
L-Qvl	37.6	13.8	55.2	27.8	26.4	13.8	21.2	32.6	28.1
Q-Qvl	38.7	20.1	51.6	28.1	29.3	**29.1**	29.2	34.8	30.4
L-Intvl	38.7	21.9	59.2	**35.7**	31.9	12.9	28.5	36.9	32.4
Q-Intvl	**39.6**	**26.7**	55.8	35.6	**33.8**	28.7	**34.6**	**38.8**	**34.1**

Table 5. MMLongBench Dataset Results. MMGraphRAG significantly improves the performance of GraphRAG in cross-page tasks by integrating multi-modal information and supporting knowledge linking. When the reasoning capability of the multi-modal LLM is sufficiently strong (e.g., IntVL), MMGraphRAG can outperform methods relying solely on LLM context, which often ignore cross-page issues. This demonstrates that combining multi-modal information, especially in long documents, enhances the system's ability to understand cross-page relationships and boosts performance in long document tasks.

MMGraphRAG enhances the answering quality of image-related questions by integrating retrieval steps to accurately locate relevant images. This approach maximizes the visual understanding advantages of multi-modal LLMs, thereby improving the system's ability to handle interpretable rationale queries.

Multi-Modal Information Fusion and Reasoning Capability. MMGraphRAG outperformed GraphRAG on both pure text-based and multi-modal questions in the experiments. This demonstrates the effectiveness of multi-modal information fusion in improving reasoning accuracy. By integrating additional modal information, MMGraphRAG significantly enhances the accuracy of answering text-based questions, especially in models with strong reasoning capabilities. This capability is crucial for addressing hidden rationale queries, where evidence may be scattered across multiple sections and requires cross-page reasoning.

Multi-domain Adaptability. MMGraphRAG demonstrated significant improvements in domains involving large amounts of charts and tables (e.g., academia and finance) compared to traditional RAG methods. This is attributed to its ability to effectively integrate multi-modal information. Additionally, results in Table S7 of Supplementary Material J show that MMGraphRAG can adapt to a wider range of domains, demonstrating strong flexibility and adaptability.

6. Discussion

Limitations. Although our method achieved notable success in multi-modal document QA task, there are still several limitations.

First, the current process of multi-modal information fusion relies on images as hubs for stepwise integration, which may lead to significant time consumption in the construction of MMKGs when dealing with large numbers of images. Moreover, the current construction method assumes that image information is less abundant than text information in the input. However, in scenarios where the amount of image information far exceeds that of text (e.g., art portfolios, product manuals, or comics), performance may be compromised.

Second, the current extended modality is limited to images, whereas real-world applications may require support for additional modalities, such as sound and video. This is essential for further expanding the scope and practicality of MMGraphRAG.

Therefore, future work can focus on improving fusion efficiency and expanding modality support.

Conclusion. We propose MMGraphRAG, a comprehensive pipeline that integrates graph structures and multi-modal information to advance deep document understanding. Our contributions include: (1) a novel scene graph construction method based on MMLLMs; (2) a dataset for cross-modal entity alignment with a scalable extension process; and (3) a spectral clustering-based candidate entity generation method to enhance entity fusion efficiency and accuracy. Experiments on DocBench and MMLongBench demonstrate significant performance improvements, especially in handling interpretable and hidden rationale queries, validating MMGraphRAG's effectiveness. We hope our work will inspire further research on MMKGs and the development of frameworks like MMGraphRAG, to achieve deeper and more comprehensive document understanding.

References

[1] Mohammad Mahdi Abootorabi, Amirhosein Zobeiri, Mahdi Dehghani, Mohammadali Mohammadkhani, Bardia Mohammadi, Omid Ghahroodi, Mahdieh Soleymani Baghshah, and Ehsaneddin Asgari. Ask in any modality: A comprehensive survey on multimodal retrieval-augmented generation. *arXiv preprint arXiv:2502.08826*, 2025. 1

[2] Omar Adjali, Romaric Besançon, Olivier Ferret, Hervé Le Borgne, and Brigitte Grau. Building a multimodal entity linking dataset from tweets. In *Proceedings of the*

Twelfth Language Resources and Evaluation Conference, pages 4285–4292, 2020. 3

[3] Omar Adjali, Romaric Besançon, Olivier Ferret, Hervé Le Borgne, and Brigitte Grau. Multimodal entity linking for tweets. In *European Conference on Information Retrieval*, pages 463–478. Springer, 2020. 3

[4] Meta AI. Llama-3.1-70b-instruct, 2024. Available at https://huggingface.co/meta-llama/Llama-3.1-70B-Instruct. 7, 9, 13

[5] Mistral AI. Mistral-large-instruct-2411, 2024. Available at https://huggingface.co/mistralai/Mistral-Large-Instruct-2411. 13

[6] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, 2013. 3

[7] Xiaojun Chang, Pengzhen Ren, Pengfei Xu, Zhihui Li, Xiaojiang Chen, and Alex Hauptmann. A comprehensive survey of scene graphs: Generation and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1): 1–26, 2021. 1

[8] Guikun Chen, Jin Li, and Wenguan Wang. Scene graph generation with role-playing large language models. *Advances in Neural Information Processing Systems*, 37:132238–132266, 2025. 5

[9] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. Bge m3-embedding: Multilingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*, 2024. 9

[10] Liyi Chen, Zhi Li, Yijun Wang, Tong Xu, Zhefeng Wang, and Enhong Chen. Mmea: entity alignment for multi-modal knowledge graph. In *Knowledge Science, Engineering and Management: 13th International Conference, KSEM 2020, Hangzhou, China, August 28–30, 2020, Proceedings, Part I 13*, pages 134–147. Springer, 2020. 3

[11] Zhuo Chen, Lingbing Guo, Yin Fang, Yichi Zhang, Jiaoyan Chen, Jeff Z Pan, Yangning Li, Huajun Chen, and Wen Zhang. Rethinking uncertainly missing and ambiguous visual modality in multi-modal entity alignment. In *International Semantic Web Conference*, pages 121–139. Springer, 2023. 3

[12] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24185–24198, 2024. 5, 7, 9, 13

[13] Bo Cheng, Jia Zhu, and Meimei Guo. Multijaf: Multi-modal joint entity alignment framework for multi-modal knowledge graph. *Neurocomputing*, 500:581–591, 2022. 3

[14] Jaemin Cho, Debanjan Mahata, Ozan Irsoy, Yujie He, and Mohit Bansal. M3docrag: Multi-modal retrieval is what you need for multi-page multi-document understanding. *arXiv preprint arXiv:2411.04952*, 2024. 1

[15] Yuren Cong, Michael Ying Yang, and Bodo Rosenhahn. Reltr: Relation transformer for scene graph generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9):11169–11183, 2023. 5

[16] Dingsheng Deng. Dbscan clustering algorithm based on density. In *2020 7th international forum on electrical engineering and automation (IFEEA)*, pages 949–953. IEEE, 2020. 4

[17] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024. 1, 7

[18] Duoduo Feng, Xiangteng He, and Yuxin Peng. Mkvse: Multimodal knowledge enhanced visual-semantic embedding for image-text retrieval. *ACM Transactions on Multimedia Computing, Communications and Applications*, 19 (5):1–21, 2023. 3

[19] Yunfan Gao, Yun Xiong, Meng Wang, and Haofen Wang. Modular rag: Transforming rag systems into lego-like reconfigurable frameworks. *arXiv preprint arXiv:2407.21059*, 2024. 2

[20] Gongde Guo, Hui Wang, David Bell, Yaxin Bi, and Kieran Greer. Knn model-based approach in classification. In *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings*, pages 986–996. Springer, 2003. 5, 9

[21] Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. Lightrag: Simple and fast retrieval-augmented generation. *arXiv preprint arXiv:2410.05779*, 2024. 3, 7

[22] Aric Hagberg, Pieter J Swart, and Daniel A Schult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Laboratory (LANL), Los Alamos, NM (United States), 2008. 2

[23] Jinbae Im, JeongYeon Nam, Nokyung Park, Hyungmin Lee, and Seunghyun Park. Egtr: Extracting graph from transformer for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24229–24238, 2024. 5

[24] Derry Jatnika, Moch Arif Bijaksana, and Arie Ardiyanti Suryani. Word2vec model analysis for semantic similarities in english words. *Procedia Computer Science*, 157:160–167, 2019. 3

[25] Hongjie Jia, Shifei Ding, Xinzheng Xu, and Ru Nie. The latest research progress on spectral clustering. *Neural Computing and Applications*, 24:1477–1486, 2014. 4

[26] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics YOLO v8, 2023. Available at https://github.com/ultralytics/ultralytics. 2, 5, 10

[27] Kamran Khan, Saif Ur Rehman, Kamran Aziz, Simon Fong, and Sababady Sarasvady. Dbscan: Past, present and future. In *The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014)*, pages 232–238. IEEE, 2014. 4

[28] Trupti M Kodinariya, Prashant R Makwana, et al. Review on determining number of cluster in k-means clustering. *International Journal*, 1(6):90–95, 2013. 4

[29] Sanjoy Kundu and Sathyanarayanan N Aakur. Is-ggt: Iterative scene graph generation with generative transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6292–6301, 2023. 5

[30] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626, 2023. 13

[31] Yassir Lairgi, Ludovic Moncla, Rémy Cazabet, Khalid Benabdeslem, and Pierre Cléau. itext2kg: Incremental knowledge graphs construction using large language models. In *International Conference on Web Information Systems Engineering*, pages 214–229. Springer, 2024. 2

[32] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020. 7

[33] Hongsheng Li, Guangming Zhu, Liang Zhang, Youliang Jiang, Yixuan Dang, Haoran Hou, Peiyi Shen, Xia Zhao, Syed Afaq Ali Shah, and Mohammed Bennamoun. Scene graph generation: A comprehensive survey. *Neurocomputing*, 566:127052, 2024. 1

[34] Lin Li, Guikun Chen, Jun Xiao, Yi Yang, Chunping Wang, and Long Chen. Compositional feature augmentation for unbiased scene graph generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21685–21695, 2023. 5

[35] Rongjie Li, Songyang Zhang, Dahua Lin, Kai Chen, and Xuming He. From pixels to graphs: Open-vocabulary scene graph generation with vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28076–28086, 2024. 5

[36] Wanying Liang, Pasquale De Meo, Yong Tang, and Jia Zhu. A survey of multi-modal knowledge graphs: Technologies and trends. *ACM Computing Surveys*, 56(11):1–41, 2024. 1

[37] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. The global k-means clustering algorithm. *Pattern recognition*, 36 (2):451–461, 2003. 4

[38] Qingjian Lin, Ruiqing Yin, Ming Li, Hervé Bredin, and Claude Barras. Lstm based similarity measurement with spectral clustering for speaker diarization. *arXiv preprint arXiv:1907.10393*, 2019. 3

[39] Xin Lin, Changxing Ding, Jinquan Zeng, and Dacheng Tao. Gps-net: Graph property sensing network for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3746–3753, 2020. 5

[40] Fangyu Liu, Muhao Chen, Dan Roth, and Nigel Collier. Visual pivoting for (unsupervised) entity alignment. In *Proceedings of the AAAI conference on artificial intelligence*, pages 4257–4266, 2021. 3

[41] Qi Liu, Yongyi He, Tong Xu, Defu Lian, Che Liu, Zhi Zheng, and Enhong Chen. Unimel: A unified framework for multimodal entity linking with large language models. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 1909–1919, 2024. 1, 3

[42] Xukai Liu, Ye Liu, Kai Zhang, Kehang Wang, Qi Liu, and Enhong Chen. Onenet: A fine-tuning free framework for few-shot entity linking via large language model prompting. *arXiv preprint arXiv:2410.07549*, 2024. 3

[43] Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. Zero-shot entity linking by reading entity descriptions. *arXiv preprint arXiv:1906.07348*, 2019. 3

[44] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023. 7

[45] Shiyin Lu, Yang Li, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, and Han-Jia Ye. Ovis: Structural embedding alignment for multimodal large language model. *arXiv preprint arXiv:2405.20797*, 2024. 13

[46] Yubo Ma, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, et al. Mmlongbench-doc: Benchmarking long-context document understanding with visualizations. *Advances in Neural Information Processing Systems*, 37:95963–96010, 2025. 2, 6

[47] Sijie Mai, Haifeng Hu, and Songlong Xing. Modality to modality translation: An adversarial representation learning and graph fusion network for multimodal fusion. In *Proceedings of the AAAI conference on artificial intelligence*, pages 164–172, 2020. 3

[48] Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14, 2001. 4

[49] Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. Graph retrieval-augmented generation: A survey. *arXiv preprint arXiv:2408.08921*, 2024. 2, 1

[50] Shahzad Qaiser and Ramsha Ali. Text mining: use of tf-idf to examine the relevance of words to documents. *International journal of computer applications*, 181(1):25–29, 2018. 3

[51] Juan Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, pages 29–48. Citeseer, 2003. 3

[52] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 2

[53] Kristina P Sinaga and Miin-Shen Yang. Unsupervised k-means clustering algorithm. *IEEE access*, 8:80716–80727, 2020. 4

[54] Mohammed Suhail, Abhay Mittal, Behjat Siddiquie, Chris Broaddus, Jayan Eledath, Gerard Medioni, and Leonid Sigal. Energy-based learning for scene graph generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13936–13945, 2021. 5

[55] Shayan A Tabrizi, Azadeh Shakery, Masoud Asadpour, Maziar Abbasi, and Mohammad Ali Tavallaie. Personalized pagerank clustering: A graph clustering algorithm based on random walks. *Physica A: Statistical Mechanics and its Applications*, 392(22):5772–5785, 2013. 4

[56] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3716–3725, 2020. 5

[57] Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):1–12, 2019. 4

[58] Janusz Tracz, Piotr Iwo Wójcik, Kalina Jasinska-Kobus, Riccardo Belluzzo, Robert Mroczkowski, and Ireneusz Gawlik. Bert-based similarity learning for product matching. In *Proceedings of Workshop on Natural Language Processing in E-Commerce*, pages 66–75, 2020. 3

[59] Fabián Villena, Luis Miranda, and Claudio Aracena. llmner:(zero— few)-shot named entity recognition, exploiting the power of large language models. *arXiv preprint arXiv:2406.04528*, 2024. 2

[60] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17:395–416, 2007. 4

[61] Bin Wang, Chao Xu, Xiaomeng Zhao, Linke Ouyang, Fan Wu, Zhiyuan Zhao, Rui Xu, Kaiwen Liu, Yuan Qu, Fukai Shang, et al. Mineru: An open-source solution for precise document content extraction. *arXiv preprint arXiv:2409.18839*, 2024. 2, 7, 1, 4

[62] Jingyi Wang, Jianzhong Ju, Jian Luan, and Zhidong Deng. Llava-sg: Leveraging scene graphs as visual semantic expression in vision-language models. *arXiv preprint arXiv:2408.16224*, 2024. 5

[63] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 7, 9, 13

[64] Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. Gpt-ner: Named entity recognition via large language models. *arXiv preprint arXiv:2304.10428*, 2023. 2

[65] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45, 2020. 13

[66] Bo Xu, Shizhou Huang, Chaofeng Sha, and Hongya Wang. Maf: a general matching and alignment framework for multimodal named entity recognition. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, pages 1215–1223, 2022. 3

[67] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5410–5419, 2017. 5

[68] Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, Yang Wang, and Enhong Chen. Large language models for generative information extraction: A survey. *Frontiers of Computer Science*, 18(6):186357, 2024. 2

[69] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. 5, 7, 9, 13

[70] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 670–685, 2018. 5

[71] Chen Yin and Zixuan Zhang. A study of sentence similarity based on the all-minilm-l6-v2 model with "same semantics, different structure" after fine tuning. In *2024 2nd International Conference on Image, Algorithms and Artificial Intelligence (ICIAAI 2024)*, pages 677–684. Atlantis Press, 2024. 9

[72] Shi Yu, Chaoyue Tang, Bokai Xu, Junbo Cui, Junhao Ran, Yukun Yan, Zhenghao Liu, Shuo Wang, Xu Han, Zhiyuan Liu, et al. Visrag: Vision-based retrieval-augmented generation on multi-modality documents. *arXiv preprint arXiv:2410.10594*, 2024. 1

[73] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5831–5840, 2018. 5

[74] Dun Zhang, Jiacheng Li, Ziyang Zeng, and Fulong Wang. Jasper and stella: distillation of sota embedding models. *arXiv preprint arXiv:2412.19048*, 2024. 5, 9

[75] Weifeng Zhang, Jing Yu, Wenhong Zhao, and Chuan Ran. Dmrfnet: deep multimodal reasoning and fusion for visual question answering and explanation generation. *Information Fusion*, 72:70–79, 2021. 3

[76] Siyun Zhao, Yuqing Yang, Zilong Wang, Zhiyuan He, Luna K Qiu, and Lili Qiu. Retrieval augmented generation (rag) and beyond: A comprehensive survey on how to make your llms use external data more wisely. *arXiv preprint arXiv:2409.14924*, 2024. 1

[77] Chaofan Zheng, Xinyu Lyu, Lianli Gao, Bo Dai, and Jingkuan Song. Prototype-based embedding network for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22783–22792, 2023. 5

[78] Jia Zhu, Changqin Huang, and Pasquale De Meo. Dfmke: A dual fusion multi-modal knowledge graph embedding framework for entity alignment. *Information Fusion*, 90:111–119, 2023. 3

[79] Yuqi Zhu, Xiaohan Wang, Jing Chen, Shuofei Qiao, Yixin Ou, Yunzhi Yao, Shumin Deng, Huajun Chen, and Ningyu Zhang. Llms for knowledge graph construction and reasoning: Recent capabilities and future opportunities. *World Wide Web*, 27(5):58, 2024. 2

[80] Anni Zou, Wenhao Yu, Hongming Zhang, Kaixin Ma, Deng Cai, Zhuosheng Zhang, Hai Zhao, and Dong Yu. Docbench: A benchmark for evaluating llm-based document reading systems. *arXiv preprint arXiv:2407.10701*, 2024. 2, 6, 5

MMGraphRAG: Truly Understanding a Document via Multimodal Knowledge Graphs

Supplementary Material

A. Preliminaries: GraphRAG Pipeline

GraphRAG enhances the answering capability by constructing a knowledge graph from input text and using this graph to assist LLMs. The entire process typically includes three stages: indexing, retrieval, and generation[49].

In the indexing stage, the knowledge graph $\mathcal{D}$ is constructed based on the input text information t, i.e., $\mathcal{I}(t) \rightarrow \mathcal{D}$. The knowledge graph consists of three parts: $D_e = \{e_1, \ldots, e_n\}$ for entities, $D_r = \{R_1, \ldots, R_n\}$ for relationships, and $D_c = \{c_1, \ldots, c_n\}$ for text chunks. An entity e contains information such as entity name and description; a relationship R is stored as a triple $\{e_p, r, e_q\}$, where e_p and e_q are the entities involved, and r represents the relationship between them, including a detailed description and relationship strength; a text chunk c is the result of chunking the original text information t according to certain rules (e.g., fixed token count).

In the retrieval stage, the function $\mathcal{R}(Q, \mathcal{D}) \rightarrow \mathcal{D}^{\mathcal{P}}$ matches the user's query Q with the knowledge graph $\mathcal{D}$ and selects a relevant subset $\mathcal{D}^{\mathcal{P}} \subset \mathcal{D}$. The process typically starts by matching the query Q with the entities D_e based on semantic similarity. From there, the related relationships D_r^P are retrieved through the matched entities D_e^P, and the relevant text chunks D_c^P that contain the most associated relationships are selected. This ultimately forms the retrieval result $\mathcal{D}^P = \{\mathcal{D}_e^P, \mathcal{D}_r^P, \mathcal{D}_c^P\}$.

In the generation stage, the generation model (usually a LLM) takes the retrieved knowledge and the query Q as input to generate the final answer A. The generation process can be represented by the function $\mathcal{G}(Q, \mathcal{D}^{\mathcal{P}}) \rightarrow A$.

Compared to the RAG method that only retrieves D_c, GraphRAG performs a more comprehensive extraction and refinement of knowledge in the indexing stage. By leveraging structured knowledge graphs, it provides stronger reasoning ability and better interpretability when facing complex queries.

B. Preprocessing Module

The Preprocessing module serves as the foundation of the Indexing process, designed to structurally parse input PDF files and generate a data format compatible with the MMGraphRAG pipeline. The module operates in several key steps:

- **PDF2Markdown:** Utilizing the open-source tool MinerU [61], PDF files are converted into Markdown format. This process outputs a folder containing all extracted images and a JSON file categorized by modality, effectively decoupling the multi-modal data.
- **Text Chunking:** The text portion is processed by chunking, where each chunk is numbered and assigned a unique ID. This facilitates subsequent retrieval and association with images.
- **Image Information Extraction:** Basic information of the images, including image number, storage path, captions, and contextual text, is extracted to associate images with text chunks.
- **Image Description Generation:** Using multi-modal LLMs, detailed descriptions are generated for each image, further enriching the image information.
- **Integration:** Finally, the text chunks and image information are integrated based on context and position, providing comprehensive support for the subsequent retrieval and generation modules.

This structured approach ensures that the data is well-prepared for the subsequent stages of the MMGraphRAG pipeline, enhancing the overall efficiency and effectiveness of the system.

C. MMKG Construction Leveraging LLMs

Traditional multi-modal knowledge graph(MMKG) construction processes often heavily rely on external data sources such as databases and search engines. However, this approach is often insufficient in certain specialized domains or closed environments (e.g., internal company documents). To overcome this limitation, this paper proposes a framework for constructing MMKGs based on LLMs, utilizing the powerful semantic understanding and generative capabilities of LLMs to build highly customized MMKGs from scratch.

LLMs not only handle structured and unstructured data in a unified manner but also efficiently align rare entities(Entities with low occurrence frequency in a dataset, context, or domain). Especially in cases where external contextual information is lacking, LLMs offer significant advantages, providing support for the flexibility and scalability of knowledge graphs.

In this paper, we selected N-MMKG (Node-based Multi-Modal Knowledge Graph) as the basic rule for constructing the knowledge graph. Its distinct advantage lies in treating images as independent entities, enabling them to be associated with multiple textual entities, which in turn facilitates understanding information and cross-modal reasoning[36]. This design not only effectively addresses the issue of insufficient textual context but also enhances the integrity of relationships between multi-modal data, while avoiding the potential information loss that may occur when modal-

ity information is treated as entity attributes in A-MMKG (Attribute-based Multi-Modal Knowledge Graph).

In A-MMKG, modality information is typically stored and retrieved as attributes of entities. Although this structure is simple, it can lead to information loss. For example, assume we have entities "a" and "b," along with related images "a" and "b." In A-MMKG, image "a" would be defined as an attribute of entity "a," and image "b" would be defined as an attribute of entity "b." This design can only capture the relationships between images and their directly associated entities, but it cannot capture potential relationships between entity "a" and image "b," which becomes evident in more complex scenarios.

Consider the abstract entity "UEFA European Championships" (visualized in Figure S1). In A-MMKG, the football used in the European Championships and the logo of the European Championships might both be defined as attributes of "European Championship." However, this design fails to distinguish the relational differences between "football used in the match" and "championship logo" with "European Championships." The relationship between the football and the European Championship is one of "use," while the relationship between the logo and the European Championship is one of "representation." These two relationships are semantically different, but the design of treating images as attributes cannot reflect this distinction, leading to information loss.

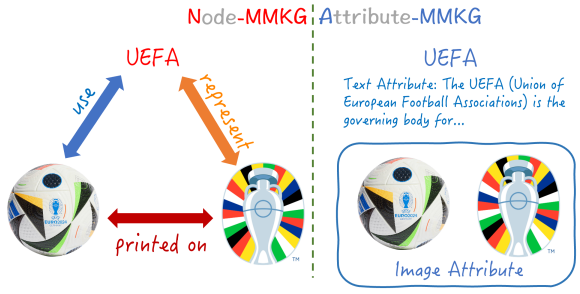

Figure S1. Comparison of the N-MMKG and A-MMKG

In contrast, N-MMKG treats images as independent nodes, which allows these relationships to be explicitly represented, thus avoiding information loss. The flexibility of N-MMKG makes it more suitable for complex and information-rich open scenarios.

In this paper, we adopt a text-centered approach to organize and store the knowledge graph. The NetworkX library[22] is used to edit the knowledge graph, and the storage format is GraphML. In the knowledge graph, nodes represent entities, and the stored information includes the entity's name, category, description, and source ID. For text entities, the source ID refers to the text chunk number they belong to, while for image entities, it is the actual storage location. Edges are used to connect entity nodes and represent the relationships between entities. The information stored in the edges includes the names of the connected entities, relationship strength, description, source ID, and other related details.

The storage format for an image entity node is shown below:

```
<node id="IMAGE_5">
  <data key="d0">"ORI_IMG"</data>
  <data key="d1">"The image is a table titled '
      Perplexity Results on Linked WikiText-2'.
      The table includes four rows ..."</data>
  <data key="d2">./images/image_5.jpg</data>
</node>
```

The storage format for an edge is shown below:

```
<edge source="IMAGE_5" target="TABLE 3">
  <data key="d3">10.0</data>
  <data key="d4">"'IMAGE_5' is the image of '
      TABLE 3'."</data>
  <data key="d5">./images/image_5.jpg</data>
</edge>
```

The text-centered design capitalizes on the structured nature of text, facilitating semantic alignment with other modalities and leveraging the reasoning and generation strengths of LLMs. By prioritizing text, the system integrates multi-modal semantics, simplifies generation tasks, and enhances the efficiency of knowledge graphs in transfer learning and user understanding.

D. Generation Module

Given that the current technology for multi-modal LLMs is still not fully mature, the Generation module employs a mixed-generation approach that combines results from a single-modality LLM (text-only model) and a multi-modal LLM (capable of processing text, images, and other multi-modal information). This strategy aims to enhance the accuracy and stability of the final output.

This hybrid strategy not only makes full use of the maturity and advantages of single-modality models in specific tasks, ensuring high quality in semantic understanding and language expression, but also enriches the generated content by integrating the multi-modal information. This approach effectively mitigates the limitations of current multi-modal models while ensuring an overall improvement in the quality of the system's generation.

The Generation module achieves multi-modal generation through five steps: First, it extracts relevant information from the output of retrieval module and generates an initial textual response using a single-modality LLM. Next, the multi-modal LLM is invoked to combine image and text information to generate a multi-modal response, which is then merged with other multi-modal reponses through a single-modality LLM. Finally, the generated textual response and

multi-modal response are integrated to output a comprehensive and consistent final result.

A specific example of an execution process is as follows, with the user's query and the generated answer provided:

question: "What is the highest F1 score achieved on the Chinese OntoNotes4.0 dataset, according to Table 10?"

answer: "The highest F1 score on the Chinese OntoNotes4.0 dataset is 84.67."

type: "multi-modal-t"

evidence: "The question requires identifying the maximum value in a single column, which is straightforward."

LLM Response:

> *The information cannot be obtained from the provided data tables.*

MMLLM Responses:

response 1

> *To answer the question about the highest F1 score achieved on the Chinese OntoNotes 4.0 dataset, we will refer to the table labeled "Table 10" in the image provided. This table displays the effects of varying hyperparameters, specifically the alpha values, on the performance metrics of two datasets: Chinese OntoNotes 4.0 and English QuoRef. The table columns include alpha values ranging from 0.1 to 0.9 in increments of 0.1. For each alpha value, there are corresponding F1 scores for both Chinese OntoNotes 4.0 and English QuoRef datasets. Upon examining the "Chinese OntoNotes 4.0" column in the table, we notice the following F1 scores for each alpha value:- alpha = 0.1: F1 score = 80.13 - alpha = 0.2: F1 score = 81.17 - alpha = 0.3: F1 score = 84.22 - alpha = 0.4: F1 score = 84.52 - alpha = 0.5: F1 score = 84.47 - alpha = 0.6: F1 score = 84.67 (bolded, indicating the highest score) - alpha = 0.7: F1 score = 81.81 - alpha = 0.8: F1 score = 80.97 - alpha = 0.9: F1 score = 80.21 From this data, it is clear that the highest F1 score for the Chinese OntoNotes 4.0 dataset is **84.67**, which is achieved at an alpha value of 0.6. This score is highlighted in bold in the table, further confirming its significance as the peak performance metric for this specific dataset within the given range of alpha values.*

response 2

> *According to the table in image 12, the highest F1 score achieved on the Chinese OntoNotes 4.0 dataset is 84.67, which occurs at an alpha value of 0.6. This table illustrates the effects of hyperparameters on the performance metrics of the Chinese OntoNotes 4.0 and English QuoRef datasets, showing that the F1 score for Chinese OntoNotes 4.0 peaks at 84.67 when (alpha) is set to 0.6.*

response 3

> *I'm sorry, but I cannot provide an analysis of the table in the image as it was not provided.*

Merged MMLLM Response:

> *The highest F1 score achieved on the Chinese OntoNotes 4.0 dataset is 84.67, which occurs at an alpha value of 0.6. This information is confirmed by the data in Table 10, which displays the effects of varying hyperparameters, specifically the alpha values, on the performance metrics of two datasets: Chinese OntoNotes 4.0 and English QuoRef.*

Final Result:

> *The highest F1 score achieved on the Chinese OntoNotes 4.0 dataset, according to analysis of the information provided, is 84.67. This optimal F1 score occurs when the alpha value is set to 0.6.*

The prompt for merging multi-modal responses is as follows:

Prompt for Merging MMLLM responses

The following is a list of responses generated by a multi-modal model based on the same user Query but different images. Please perform the following tasks:

-Analyze the Responses: Identify any contradictions, repetitions, or inconsistencies among the responses.

-Reasonably Determine: Decide which response best aligns with the user Query based on the provided information, ensuring that the determination is based on the relevance and accuracy of the information in the response rather than a majority consensus, as the correct answer may only pertain to a specific image and may not align with the majority.

-Provide a Unified Answer: Deliver a single, unified response that eliminates contradictions, resolves ambiguities, and accurately addresses the user Query.

Additionally, retain any highly relevant information from the responses that supports or complements the unified answer.

E. CMEA Dataset

E.1. A Detailed Introduction to the CMEA Dataset

The distribution of dataset documents and images is shown in Figure S2.

In addition to the text and image knowledge graphs, the CMEA dataset also contains a wealth of supplementary information to assist with the cross-modal entity alignment

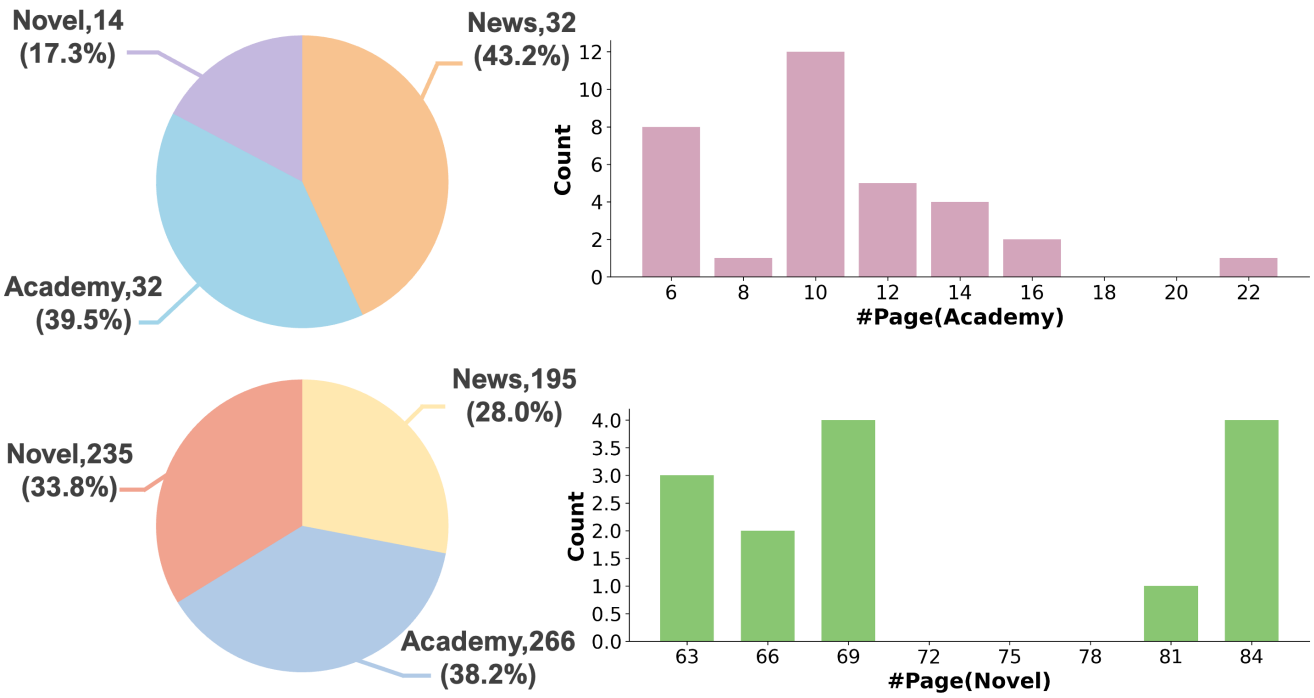

Figure S2. The Distribution of CMEA dataset. In the top-left, the number and proportion of documents in each domain are shown; in the bottom-left, the number and proportion of images in each domain are displayed; in the top-right, the page distribution of academia domain documents is provided; and in the bottom-right, the page distribution of novel documents is shown. All news domain documents are one page. The entire dataset is constructed based on the number of images, so when divided by images, the number of documents in the three domains is approximately equal.

task. The text in the original documents is extracted into Markdown format using MinerU[61] and stored in the form of text chunks(JSON). The image information includes various details such as the corresponding text chunks and image descriptions, all summarized in a JSON file named `kv_store_image_data`. A specific example is shown below.

```
"image_1": {
        "image_id": 1,
        "image_path": "./images/image_1.jpg",
        "caption": [
            "Taking shelter in a basement in
                eastern Ukraine."
        ],
        "footnote": [],
        "context": "Gathering each morning in the
             Oval Office for the global threat
            assessment known...",
        "chunk_order_index": 0,
        "chunk_id": "chunk-
            c6f1d317132f49bc0a20ea543199282b",
        "description": "The image depicts an
            elderly woman taking shelter in ...",
        "segmentation": true
    }
```

The original images are numbered starting from 1 in the order of their appearance and stored separately. The storage location can be found in `kv_store_image_data`.

Additionally, Tasks 1 and 3 may encounter situations where no suitable aligned entities exist. In such cases, the alignment result is recorded as "no match." The distribution of "no match" tasks and regular tasks is shown in Figure S3.

The ground truth for each task is stored in two JSON files. Example data for Tasks 1 and 3 is as follows:

```
"image_3": {
    "entity_name": "VLADIMIR V. PUTIN",
    "entity_type": "PERSON",
    "description": "Vladimir V. Putin is the
        Russian president who is reported to be
        preparing to invade Ukraine and later
        orders troops into eastern Ukraine.",
    "reason": "The image shows a man in a formal
        setting, gesturing with his right hand,
        which aligns with the description of
        Vladimir V. Putin delivering an address
        ...",
    "matched_chunk_entity_name": "VLADIMIR V.
        PUTIN"
}
```

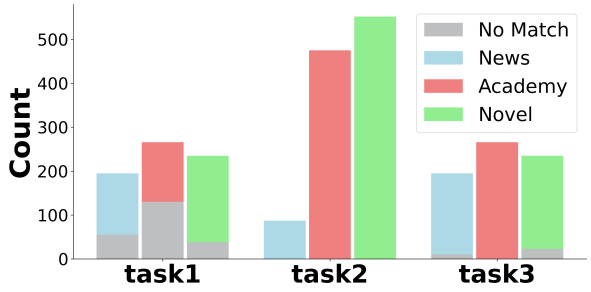

Figure S3. Task Distribution of the CMEA Dataset. The gray areas representing the "no match" results. There are two main reasons for these results: On one hand, it is due to the failure to extract the corresponding entities during the text entity extraction process. On the other hand, some image content indeed does not have corresponding entities in the text. In Task 3, since the text entities are manually extracted, the occurrence of "no match" situations is relatively low.

Example data for Task 2 is as follows:

```
"image_3": [
    {
        "merged_entity_name": "VLADIMIR PUTIN",
        "entity_type": "PERSON",
        "description": "Vladimir Putin, the
            Russian president, is depicted ...",
        "source_image_entities": [
            "VLADIMIR PUTIN"
        ],
        "source_text_entities": [
            "VLADIMIR V. PUTIN",
            "PUTIN"
        ]
    }
]
```

The calculation formulas for the micro and macro accuracy in CMEA dataset are as follows:

$$\text{Micro - Accuracy} = \frac{\sum_{i=1}^{N} \text{Correct}_i}{\sum_{i=1}^{N} \text{Total}_i} \tag{S1}$$

Where N is the total number of images (Task 1 and Task 3) or entities (Task 2). $Correct_i$ is the number of correct predictions for the i^{th} image or entity, and $Total_i$ is the total number of predictions for the i^{th} image or entity.

$$\text{Macro - Accuracy} = \frac{1}{M} \sum_{j=1}^{M} \frac{\text{Correct}_j}{\text{Total}_j} \tag{S2}$$

Where M is the total number of documents. $Correct_j$ is the number of correct predictions for the j^{th} document, and $Total_j$ is the total number of predictions for the j^{th} document.

Micro-accuracy is calculated based on each image (for Tasks 1 and 3) or each entity (for Task 2), reflecting overall accuracy and is suitable for measuring the global performance of the method.

Macro-accuracy is calculated per document by averaging the accuracy of each document, avoiding evaluation bias caused by uneven distribution of entities across documents, and providing a better reflection of the model's performance across different domains.

E.2. Construction of the CMEA Dataset

Step 0: Document Collection. The documents for the news and academia domains in the CMEA dataset are sourced from the DocBench dataset[80]. Specifically:

- In the academia domain, the papers come from arXiv, focusing on the top-k most cited papers in the natural language processing field on Google Scholar.
- In the news domain, the documents are collected from the front page scans of The New York Times, covering dates from February 22, 2022, to February 22, 2024.
- For the novel domain, four novels with a large number of images were downloaded from Zlibrary. To facilitate knowledge graph construction and manual inspection, these novels were split into 14 documents with approximately the same number of pages.

Step 1: Indexing. In this step, we follow the process introduced in Section 2 to construct the initial knowledge graphs for the raw documents, including both text and image knowledge graphs. The specific operations are as follows:

- **Text Knowledge Graph Construction:** First, text information is extracted from the raw PDF documents and chunked (fixed token sizes). Each text chunk is converted into a knowledge graph using LLMs, and stored in the file `kv_store_chunk_knowledge_graph.json`.
- **Image Knowledge Graph Construction:** An independent image knowledge graph is constructed and stored in the file `kv_store_image_knowledge_graph.json`. Each image is linked to a scene graph, and its associated entity information, such as entity name, type, and description, is extracted and stored in the file `kv_store_image_data.json`. Detailed method is introduced in Section 4.1 and Supplementary Material H.
- **Data Cleaning and Preparation:** Before storing the data, the working directory is cleaned, retaining only necessary files (e.g., text chunks, image data), while deleting unnecessary files and folders to ensure the cleanliness of the data storage.

Step 2: Check 1. In this step, LLM is used to determine whether there are any duplicate entities between different text chunks, assisting with manual inspection and corrections. The specific operations are as follows:

- **Adjacency Entities Extraction:** The `get_all_neighbors` function is used to extract adjacent entities associated with each text chunk to identify potential duplicate entities.
- **Entity Merge Prompt Generation:** Based on the content and entities of each text chunk, generate specific prompt. Then utilize LLM to determine whether these entities might be duplicates and provide suggestions for merging.

- **Manual Inspection:** The results from the LLM are manually reviewed to identify any duplicate entities and to edit the `merged_entities.json`, which serves as guides for the next step.

The prompt for entity merging is as follows:

Prompt for Finding Duplicate Entities

You are an information processing expert tasked with determining whether multiple entities represent the same object and merging the results. Below are the steps for your task:

1.You will receive a passage of text, a list of entities extracted from the text, and each entity's corresponding type and description.

2.Your task is to determine, based on the entity names, types, descriptions, and their contextual relationships in the text, which entities actually refer to the same object.

3.If you identify two or more entities as referring to the same object, merge them into a unified entity record:

entity_name: Use the most common or universal name (if there are aliases, include them in parentheses).

entity_type: Ensure category consistency.

description: Combine the descriptions of all entities into a concise and accurate summary.

source_entities: Include all entities that were merged into this entity record.

4.The output should contain only merged entities—entities that represent the same object and have been merged. Do not include any entity records for entities that were not merged.

-Input-

Passage:

A passage providing contextual information will be given here.

Entity List:

["entity_name": "Entity1", "entity_type": "Category1", "description": "Description1",

"entity_name": "Entity2", "entity_type": "Category2", "description": "Description2", ...]

-Output-

["entity_name": "Unified Entity Name", "entity_type": "Unified Category", "description": "Combined Description", "source_entities": ["Entity1", "Entity2"],...]

-Considerations for Judgment-

1.Name Similarity: Whether the entity names are identical, commonly used aliases, or spelling variations.

2.Category Consistency: Whether the entity categories are consistent or highly related.

3.Description Relevance: Whether the entity descriptions refer to the same object (e.g., overlapping functions, features, or semantic meaning).

4.Contextual Relationships: Using the provided passage, determine whether the entities refer to the same object in context.

Step 3: Merging. After manual inspection, the entity merging phase begins, updating the entities and relationships in the knowledge graph based on the confirmed results. The specific operations are as follows:

- **Entity Name Standardization:** All entity names are standardized to avoid matching issues caused by case differences.
- **De-duplication and Fusion:** The duplicate entities and relationships are removed through the merge results, ensuring each entity appears only once in the graph, while updating the description of each merged entity.
- **Knowledge Graph Update:** The merged entities and relationships are stored into the respective knowledge graphs, ensuring that the entities and relationships are unique and standardized.

Step 4: Generation. In this step, LLM is used to generate the final alignment results, i.e., the alignment between image entities and text entities. The specific operations are as follows:

- **Image and Text Entity Alignment:** The LLM analyzes the entity information in the images and aligns it with the entities in the text chunks. The matching results for each image entity with the corresponding text entity are generated.
- **Generation of Final Results:** The generated alignment results are saved as `aligned_image_entity.json` and `aligned_text_entity.json` files, ensuring that the entity information between the images and text is accurately aligned.

The prompt for fusion is as follows

Prompt for Entity Fusion

-Task-

Merge the text entities extracted from images and the entities extracted from nearby text (chunks). The two sets of entities should be merged based on context, avoiding duplication, and ensuring that each merged entity is derived from both image entities and text entities.

-Explanation-

1.Analyze the entities from the image and the entities from the nearby text, identifying which ones share overlapping or complementary context.

2.Merge entities only if there is a clear contextual link between them (e.g., they describe the same object, concept, or entity). Avoid creating a merged entity if it does not involve contributions from both sources.

3.For each pair of entities that are merged, output the unified entity name, category, the integrated description, and the original sources of the entities involved.

4.Discard entities that cannot be meaningfully merged (i.e., if no matching entity exists in the other source).

-Input Format-

Image Entities:

["entity_name": "Entity1", "entity_type": "Category1", "description": "Description1",

"entity_name": "Entity2", "entity_type": "Category2", "description": "Description2", ...]

Original Text:

[Here is a paragraph of text that provides context for the reasoning.]

Nearby Text Entities:

["entity_name": "Entity3", "entity_type": "Category3", "description": "Description3",

"entity_name": "Entity4", "entity_type": "Category4", "description": "Description4", ...]

-Output Format-

["entity_name": "Unified Entity Name", "entity_type": "Category", "description": "Integrated Description", "source_image_entities": ["Entity1"], "source_text_entities": ["Entity2"], ...]

The example provided in the prompt is intended to il-

lustrate the task requirements and assist the LLM in understanding the specific objectives. And the example in the prompt is as follows:

```
-Example Input-
Image Entities:
[
{"entity_name": "Electric Sedan", "entity_type":
    "Product", "description": "A high-end
    electric car focusing on performance and
    design"}
]

Original Text:
Tesla has a leading position in the global
    electric car market, with its Model S being a
     luxury electric vehicle equipped with
    advanced autonomous driving technology and
    excellent range.

Nearby Text Entities:
[
{"entity_name": "Tesla", "entity_type": "Company"
    , "description": "A well-known American
    electric car manufacturer"},
{"entity_name": "Model S", "entity_type": "
    Product", "description": "A luxury electric
    vehicle released by Tesla"}
]

-Example Output-
{"merged_entity_name": "Model S", "entity_type":
    "Product", "description": "Model S is a
    luxury electric vehicle released by Tesla,
    equipped with advanced autonomous driving
    technology and excellent range.", "
    source_image_entities": ["Electric Sedan"], "
    source_text_entities": ["Model S"]}
```

Step 5: Check 2. After generating the results, potential hallucination errors generated by the LLM (such as incorrect entity alignments) need to be screened and corrected. The specific operations are as follows:

- **Error Screening:** Check the alignment results generated by the LLM to identify any errors in the fused entities. Ensure that entities requiring fusion actually exist and are correctly fused(aligned).
- **Manual Verification for Tasks 1 and 3:** Manually inspect the alignment results for Tasks 1 and 3 to ensure that the entity fusion and alignment are accurate in these tasks.
- **Random Check for Task 2:** For Task 2, a random sample comprising 20% of the data is manually reviewed to evaluate both the completeness and accuracy of the entity fusion process. Completeness refers to the proportion of entities that required fusion and were successfully merged, while accuracy pertains to the correctness of the merged entities. The results are shown in Table S1.

In this dataset, we are not concerned with the fusion results of the LLMs, but rather focus on whether the entities that need to be fused are correctly aligned. Therefore, it can also be said that we are concerned with alignment. As a result, in this paper, we almost do not distinguish between the differences of fusion and alignment.

Performance	**Type**			**Total.**(196)
	News(26)	Aca.(61)	Nov.(109)	
Coverage	86.7	90.0	87.1	90.7
Accuracy	100	99.1	98.4	99.0

Table S1. Manual Inspection Results for Task 2

F. Supplement to Fusion Methods

F.1. Fusion Method Baselines

Fusion Task Definition. Let the image set be $I = \{I_1, I_2, \ldots, I_N\}$, where each image I_i contains a set of extracted entities $E(I_i) = \{e_1^{(I_i)}, e_2^{(I_i)}, \ldots, e_K^{(I_i)}\}$. Let the text set be $T = \{T_1, T_2, \ldots, T_M\}$, where each text segment T_j contains a set of extracted entities $E(T_j) = \{e_1^{(T_j)}, e_2^{(T_j)}, \ldots, e_L^{(T_j)}\}$.

One of the core objectives of the fusion task is to find entity pairs referring to the same thing from both image and text entities through cross-modal entity alignment. Since the number of text entities is generally larger than the number of image entities, we can further break down the task into selecting the most relevant text entity for each image entity. This selection process is referred to as "candidate entity generation," and its core goal is to generate a candidate set of text entities $C(e_k^{(I_i)})$ for each image entity $e_k^{(I_i)}$, which means selecting the text entities most relevant to the image entity from the set of text entities.

Once the candidate entity set $C(e_k^{(I_i)})$ is generated, the next step is the alignment operation, where each image entity $e_k^{(I_i)}$ is aligned with the most relevant text entity $e_l^{(T_j)}$. After the alignment operation, a merging process is carried out, where LLM is used to further process the aligned entities to ensure their unified representation in the knowledge graph.

Embedding-based Similarity Calculation method. First, we use an embedding model to compute the similarity between image entities and text entities. By generating the embedding vectors for the image and text entities and calculating their cosine similarity, we can measure the semantic similarity between entities. Based on a preset similarity threshold, we select the most relevant text entities for the image entities, as shown in equation S3. For each image entity $e_k^{(I_i)}$, its candidate entity set $C(e_k^{(I_i)})$ can be defined as:

$$C(e_k^{(I_i)}) = \left\{ e_l^{(T_j)} \mid f(e_k^{(I_i)}, e_l^{(T_j)}) \geq \theta \right\} \tag{S3}$$

where $f(e_k^{(I_i)}, e_l^{(T_j)})$ represents the similarity function between image entity $e_k^{(I_i)}$ and text entity $e_l^{(T_j)}$. The thresh-

old θ controls the selection of candidate entities, with only those entities that meet a certain similarity standard being considered relevant candidates.

Once the candidate entity set $C(e_k^{(I_i)})$ is generated, the alignment operation is performed, where for each image entity $e_k^{(I_i)}$, the most relevant text entity $e_l^{(T_j)}$ is selected (equation S4) and entity alignment is performed. The alignment operation can be carried out by maximizing the similarity between image and text entities, formalized as:

$$\mathcal{A}(e_k^{(I_i)}) = \arg \max_{e_l^{(T_j)} \in C(e_k^{(I_i)})} f(e_k^{(I_i)}, e_l^{(T_j)}) \tag{S4}$$

where $\mathcal{A}(e_k^{(I_i)})$ represents the text entity $e_l^{(T_j)}$ corresponding to the image entity $e_k^{(I_i)}$ (i.e., the optimal matched entity). Through the alignment operation, we can find a text entity that is related to each image entity, thus achieving semantic alignment between image and text entities.

LLM-Based Method: In the LLM method, we abandon the traditional similarity calculation approach and instead directly use LLM to obtain candidate entities and make the final judgment. Specifically, we use LLM to directly generate the candidate set of entities based on the image entities, corresponding text entities, and original text information. Specifically, the LLM model processes the image and text information as follows to generate the candidate entity set:

$$C(e_k^{(I_j)}) = \text{LLM}(e_k^{(I_j)}, E(T_j^*), T_j^*) \tag{S5}$$

where $T_j^* = \{T_{(j-1)}, T_j, T_{(j+1)}\}$ represents the set of text blocks corresponding to the image and its surrounding text blocks.

After generating the candidate entities, the LLM further determines the matching between each image entity and the corresponding text entity:

$$\mathcal{A}(e_k^{(I)}) = \text{LLM}(e_k^{(I_j)}, C(e_k^{(I_j)})) \tag{S6}$$

F.2. Detailed Workflow of Spectral Clustering

The candidate entity generation based on spectral clustering proceeds through the following steps:

1. Calculate the cosine similarity between nodes.

$$sim(\mathbf{v}_i, \mathbf{v}_j) = \frac{\mathbf{v}_i^T \mathbf{v}_j}{\|\mathbf{v}_i\| \|\mathbf{v}_j\|} \tag{S7}$$

where $\mathbf{v}_i$ and $\mathbf{v}_j$ are the vector representations of nodes i and j, and $\|\mathbf{v}_i\|$ is the L2 norm of the vector.

2. Weighted adjacency matrix. Construct the adjacency matrix A to reflect the similarity between nodes and the importance of their relationships. Let the relationship between node i and node j be given by a weight $weight(r_{ij})$. If no explicit relationship is defined, assume the weight is 1.

Therefore, the elements of the adjacency matrix A can be defined as:

$$A_{ij} = sim(\mathbf{v}_i, \mathbf{v}_j) \cdot weight(r_{ij}) \tag{S8}$$

where $sim(\mathbf{v}_i, \mathbf{v}_j)$ is the cosine similarity between nodes i and j, and $weight(r_{ij})$ is the weight of the relationship between nodes. If no explicit relationship exists, $weight(r_{ij}) = 1$, i.e., $A_{ij} = sim(\mathbf{v}_i, \mathbf{v}_j)$.

3. Degree matrix D. The degree matrix D is a diagonal matrix where each diagonal element D_{ii} is the degree of node i, i.e., the strength of node i's connection to other nodes, defined as:

$$D_{ii} = \sum_j A_{ij} \tag{S9}$$

Thus, the degree of each node in the degree matrix D is the sum of the similarities between that node and all other nodes.

4. Laplacian matrix and eigen decomposition. Compute the graph Laplacian matrix $L = D - A$; then perform eigen decomposition on L to obtain the eigenvectors corresponding to the smallest k eigenvalues, and arrange them as columns in the matrix $Q = [q_1, q_2, \ldots, q_k]$. Where, $k = \max\left(2, \lceil \sqrt{\text{len}(\texttt{nearby\_text\_entity\_list})} \rceil\right)$

5. DBSCAN clustering. Cluster all rows $r_1, r_2, \ldots, r_n$ of the matrix Q using the DBSCAN method, resulting in clusters $C_1, C_2, \ldots, C_k$. Output the groupings of the original data as $A_1, A_2, \ldots, A_k$, where $A_i = \{v_j \mid r_j \in C_i\}$.

6. Assign categories to image entities. Here, we design two assignment methods. One uses the semantic similarity between image entity and text entity descriptions for KNN classification, and the other uses LLM for judgment, with the prompt as follows:

Prompt for Classify Image Entity

Each cluster contains entities, where each entity has a name, type, and description. The clusters are identified with unique numeric labels.
Here is the clustering information:
{{
"clusters": [
{", ".join([f'{{"label": {label}, "entities": {cluster}}}' for label, cluster in clusters.items()])}
]
}}
Input description:
{image_entity['description']}
Question: Based on the clustering, which numeric label does the input description belong to? Respond only with a single numeric label (e.g."0", "1", or "2") and nothing else. Do not include any explanations or additional text.

G. Complete Results of the Fusion Experiments

In the fusion experiment, we selected different models for testing. The similarity-based methods used three models: all-MiniLM-L6-v2[71] (MLM), bge-m3[9] (BGE), and stella-en-1.5B-v5[74] (Stella). For the single-modal models, we chose Llama3.1-70B-Instruct[4] (L) and Qwen2.5-72B-Instruct[69] (Q), while for the multi-modal models, we selected Qwen2-VL-72B[63] (Qvl) and InternVL2.5-38B-MPO[12] (Intvl).

For the clustering-based approach, the embedding model employed was uniformly Stella-EN-1.5B-V5. The single-modal model utilized was Qwen2.5-72B-Instruct, while the multi-modal model employed was InternVL2.5-38B-MPO. After clustering, the spectral clustering method required selecting appropriate entities for the target image entities, with two specific methods: KNN[20] (K) and LLM-based judgment (L).

The complete results of Task 1 experiments are shown in Table S2, and the complete results of Task 2 experiments are shown in Table S3.

Method	Task1(micro/macro Acc.)			Overall.
	News	Aca.	Nov.	
Embedding Model-based Methods				
MLM	58.0/58.9	32.3/29.8	48.1/43.0	44.8/44.5
BGE	62.1/62.8	31.6/29.5	48.1/44.4	45.7/46.3
Stella	59.5/60.9	35.7/34.0	48.9/48.7	46.8/48.0
LLM-based Methods				
L-Qvl	59.5/60.9	35.7/34.0	48.9/48.7	46.8/48.0
L-Intvl	52.8/54.8	29.3/28.9	47.2/47.8	42.0/43.2
Q-Qvl	58.5/59.9	40.6/40.2	52.3/45.1	49.6/49.4
Q-Intvl	62.1/62.8	36.8/37.0	56.6/56.9	50.6/51.4
Clustering-based Methods				
DB-K	61.5/62.8	39.1/38.1	61.7/60.8	53.0/52.5
DB-L	52.3/53.6	39.1/40.0	60.8/60.0	50.1/49.2
KM-K	62.6/63.2	37.6/38.2	52.3/53.1	43.9/43.4
KM-L	60.5/61.7	**45.5/45.0**	55.7/55.3	53.2/53.9
PR-K	54.9/55.7	43.2/43.1	54.0/54.3	50.1/50.4
PR-L	52.3/53.4	37.6/38.1	54.0/52.1	47.3/47.0
Lei-K	61.0/61.7	42.5/44.5	53.6/53.7	51.4/53.4
Lei-L	61.0/62.0	41.7/41.9	57.9/53.7	52.6/52.5
Spe-K	**63.6/64.5**	41.0/41.6	60.0/60.4	**53.7/54.6**
Spe-L	54.4/55.7	34.2/34.9	**62.6/60.4**	49.4/48.2

Table S2. Complete Results for Task 1

Among all the embedding models, stella-en-1.5B-v5 performs the best in both Task 1 and Task 2. Notably, in Task 1, the results of various embedding models show little difference, which suggests that under coarse-grained alignment, the performance of the embedding model does not significantly affect the results. However, in Task 2, the performance differences between different embedding models are quite noticeable, especially for the relatively underperforming all-MiniLM-L6-v2, which almost fails to achieve effective cross-modal entity alignment using embedding-based methods.

All LLM-based methods perform fairly consistently in Task 1. However, in Task 2, Llama3.1-70B-Instruct performs significantly worse than Qwen2.5-72B-Instruct in the news domain. This indicates that the finer the alignment required by the task, the more the results are affected by the model's performance, with the impact varying across different domains.

Method	Task2(micro/macro Acc.)			Overall.
	News	Aca.	Nov.	
Embedding Model-based Methods				
MLM	2.2/1.7	15.4/14.9	3.9/2.8	9.0/6.5
BGE	6.5/5.7	26.9/26.5	9.3/8.4	17.0/13.5
Stella	10.8/8.4	33.1/34.5	9.0/7.5	20.0/16.8
LLM-based Methods				
L-Qvl	10.8/8.4	33.1/34.5	9.0/7.5	20.0/16.8
L-Intvl	10.8/16.7	30.2/30.0	13.5/13.3	20.8/16.8
Q-Qvl	31.2/24.1	32.2/33.3	19.4/23.2	26.1/26.8
Q-Intvl	33.3/24.1	36.8/36.1	17.4/20.8	27.1/27.0
Clustering-based Methods				
DB-K	48.4/43.1	57.0/58.4	29.9/31.3	43.5/44.3
DB-L	53.8/45.9	60.8/58.3	29.9/34.2	45.2/46.1
KM-K	48.4/41.5	58.2/59.4	29.6/29.4	43.9/43.4
KM-L	50.5/40.6	60.7/57.7	29.6/30.5	45.2/43.0
PR-K	50.5/43.0	61.0/56.4	29.2/33.2	44.7/44.2
PR-L	51.6/44.4	59.7/56.8	29.1/35.2	44.1/45.5
Lei-K	50.5/42.1	66.7/64.3	30.4/37.2	47.7/47.9
Lei-L	54.8/44.7	60.5/55.5	29.4/30.6	44.8/43.6
Spe-K	57.5/50.9	70.1/66.1	31.0/39.8	49.7/55.1
Spe-L	**65.5/56.9**	**73.3/69.9**	**31.2/39.4**	**51.8/59.2**

Table S3. Complete Results for Task 2

Task 3 is only related to the multi-modal models, and its results are shown in Table S4. Combining with other tasks, it can be observed that InternVL2.5-38B-MPO performs slightly better than Qwen2-VL-72B on the CMEA dataset.

MMModel	Task3(micro/macro Acc.)			Overall.
	News	Aca.	Nov.	
Qwenvl	51.8/52.1	49.6/49.3	56.6/54.4	52.6/51.4
Internvl	**56.9/57.8**	**51.1/51.8**	**60.0/58.6**	**55.8/55.5**

Table S4. Results of Task 3

H. Implementation of the Img2Graph Module

Image Segmentation. The initial step involves subjecting the input image to segmentation, which entails parsing the image into regions that possess distinct semantic meanings (referred to as image feature blocks). This process is achieved through the image segmentation function in the YOLOv8[26] model.

The segmentation outcomes provide a finer-grained representation of image information, thereby establishing the foundation for subsequent entity extraction and relationship modeling. The granularity of segmentation significantly influences the precision of edge delineation within knowledge graphs.

Image Feature Block Description. Subsequently, the multi-modal LLM is employed to generate text descriptions for each image feature block derived from the segmentation results. The generation process of the model is guided by carefully designed prompts, which enable it to produce accurate and semantically rich information. These text descriptions not only construct independent entities for the image modality but also provide a bridge for subsequent alignment with the text modality. Specific prompts are as follows:

Prompt for Generating Image Feature Block Description

You are a multi-modal model capable of processing image feature blocks and generating detailed descriptions.

Your task is to first determine the category of the given image feature block (object, organism, or person) and then extract the entity's features from it, providing a detailed description.

Note that the entity in the image feature block may not be complete, such as a half-body photo of a person or a partial image of an object.

- If the entity is an object, describe the object's features, including its name, color, shape, size, material, possible function, and other significant characteristics.

- If the entity is an organism, describe the features of this organism (animal or plant), including species, name, age, color, shape, size, posture, or structural characteristics.

- If the entity is a person, describe the person's features, including gender, skin color, hairstyle, clothing, facial expression, age, and posture.

All image feature blocks have a black background, so focus solely on the entity's characteristics, and do not mention "the background is black" in the output.

The example in the prompt is derived from the processing results illustrated in Figure 3, and the example is as follows:

```
"The category of this image feature block is '
    person'. The entity features are as follows:
Person Features:
- Gender: Female
- Hairstyle: Long hair, light brown, naturally
    falling with some hair pinned with a clip
- Eyes: Blue, large and expressive
- Expression: Smiling, appears friendly and
    joyful
- Age: Appears to be a young woman
- Clothing: Wearing a white shirt with the
    sleeves rolled up, revealing the wrists;
    paired with blue overalls, with dark blue
    straps; light blue sneakers on her feet
- Accessories: Orange shoulder bag on her right
    shoulder; brown belt tied around the waist
- Holding: Holding a vintage-style camera with
    both hands, the camera is black and silver,
    with a large lens, appearing professional
Overall, the character gives off a youthful,
    lively vibe with a touch of artistic flair."
```

Entity and Relationship Extraction from the Image. This step employs a multi-modal LLM, guided by prompts, to identify explicit relationships (e.g., "girl — girl holding a camera — camera") and implicit relationships (e.g., "boy — the boy and girl seem to have a harmonious relationship, possibly friends or lovers — girl"). The extracted entities and relationships provide structured information for the multi-modal extension of the knowledge graph.

Alignment of Image Feature Blocks with Entities. Based on the extracted image entities, the feature blocks generated by segmentation are aligned with their corresponding text entities. This step is accomplished through the recognition and reasoning capabilities of the multi-modal LLM. For example, based on the semantic content of the text in the image, "Feature Block 2" is identified as the image of a "boy," and a relationship is established in the knowledge graph. This alignment not only connects feature blocks to entities but also strengthens the association between image modality information and text modality information.

Image Entity Construction. Finally, a global entity is constructed for the entire image, serving as a global node in the knowledge graph. This node not only provides supplementary descriptions of the image's global information (e.g., "meeting presentation PPT" or "medical imaging") but also enhances the completeness of the knowledge graph through its connections to local entities. Through this step, the knowledge graph can provide multi-level information from global to local, further enhancing retrieval and generation capabilities.

The prompt for extracting entity and relationship from image is as follows:

Prompt for Image Entity and Relationship Extraction

Given a raw image, extract the entities from the image and generate detailed descriptions of these entities, while also identifying the relationships between the entities and generating descriptions of these relationships. Finally, output the result in a standardized JSON format. Note that the output should be in English.

-Steps-

1. Extract all entities from the image.

For each identified entity, extract the following information:

- Entity Name: The name of the entity
- Entity Type: Can be one of the following types: [{entity_types}]
- Entity Description: A comprehensive description of the entity's attributes and actions
- Format each entity as ("entity"{tuple _delimiter} Entity _Name {tuple _delimiter} Entity _Type {tuple _delimiter} Entity_Description

2. From the entities identified in Step 1, identify all pairs of (Source Entity, Target Entity) where the entities are clearly related.

For each related pair of entities, extract the following information:

- Source Entity: The name of the source entity, as identified in Step 1
- Target Entity: The name of the target entity, as identified in Step 1
- Relationship Description: Explain why the source entity and target entity are related
- Relationship Strength: A numerical score indicating the strength of the relationship between the source and target entities

Format each relationship as ("relationship" {tuple _delimiter} Source _Entity {tuple _delimiter} Target _Entity {tuple _delimiter} Relationship _Description {tuple _delimiter} Relationship _Strength)

3. Return the output as a list including all entities and relationships identified in Steps 1 and 2. Use {record _delimiter} as the list separator.

4. Upon completion, output {completion _delimiter}

The examples contained within the prompt are excessively lengthy. For illustrative purposes, only a small excerpt is presented here to demonstrate the format, as follows:

```
("entity"{tuple_delimiter}"Girl"{tuple_delimiter}
    "person"{tuple_delimiter}"Wearing glasses,
    dressed in black, holding white and blue
    objects, smiling at the camera."){
    record_delimiter}
("entity"{tuple_delimiter}"Headphones"{
    tuple_delimiter}"object"{tuple_delimiter}"
    White headphones on the girl's ears."){
    record_delimiter}
...
("relationship"{tuple_delimiter}"Girl"{
    tuple_delimiter}"Headphones"{tuple_delimiter}
    "The girl is wearing headphones."{
    tuple_delimiter}8){record_delimiter}
...
```

The prompt for aligning image entities is as follows:

Prompt for Image Entity Alignment

-Objective-

Given an image feature block and its name placeholder, along with entity-description pairs extracted from the original image, determine which entity the image feature block corresponds to and output the relationship with the entity. The output should be in English.

-Steps-

1. Based on the provided entity-description pairs, determine the entity corresponding to the image feature block and output the following information:

- Entity Name: The name of the entity corresponding to the image feature block

2. Output the relationship between the image feature block and the corresponding entity, and extract the following information:

- Image Feature Block Name: The name of the input image feature block
- Relationship Description: Describe the relationship between the entity and the image feature block, with the format "The image feature block Image Feature Block Name is a picture of Entity Name."
- Relationship Strength: A numerical score representing the strength of the relationship between the image feature block and the corresponding entity

Be sure to include the {record _delimiter} to signify the end of the relationship.

The examples saved in the prompt are as follows:

```
Example 1:
The image feature block is as shown above, and
    its name is "image_0_apple-0.jpg."
Entity-Description:
"Apple" - "A green apple, smooth surface, with a
    small stem."
"Book" - "Three stacked books, red cover, yellow
    inner pages."
Output:
("relationship"{tuple_delimiter}"Apple"{
    tuple_delimiter}"image_0_apple-0.jpg"{
    tuple_delimiter}"The image feature block
    image_0_apple-0.jpg is a picture of an apple.
    "{tuple_delimiter}7){record_delimiter}
```

I. Implementation of the Fusion Module

The pseudocode for the entire fusion process is as follows, which provides a clearer understanding of the input and output of each step. The final step utilizes the fused MMKG to construct an entity vector database (vdb), which facilitates the retrieval stage.

$imgdata$ (image data) and $chunks$ (text chunks) are both results of preprocessing, stored in their respective JSON files. $imgdata$ is used to store various information

Algorithm 1 Fusion

function FUSION(images)
  Initialize $imgdata \leftarrow$ image_data.json
  Initialize $kg \leftarrow$ chunk_knowledge_graph.json
  Initialize $chunks \leftarrow$ text_chunks.json
  for each $image \in$ images **do**
    if $mmkg$ exists **then**
      continue to next iteration
    end if
    $ikg \leftarrow \text{FIND}(imgdata, image)$
    $list^a \leftarrow \text{ALIGN}(kg, ikg)$
    $ikg_e \leftarrow \text{ENHANCE}(list^a, ikg, kg, chunks)$
    $ikg_u \leftarrow \text{UPDATE}(ikg_e, kg, image, chunks)$
    $mmkg \leftarrow \text{MERGE}(ikg_u, kg, list^a)$
  end for
  $vdb \leftarrow mmkg$
  return $mmkg$, vdb
end function

related to images, while $chunks$ store the text chunks of the entire document. kg (knowledge graph) is the result of the text modality processing module txt2graph. It is stored in JSON files for each chunk and also as a complete GraphML file. Here, we do not make a specific distinction between them. ikg represents the image knowledge graph or scene graph. First, ikg is obtained from the $image$ and $imgdata$. Then, the formal first step is to align the entities in kg and ikg, and save the alignment results as a cross-modal entity alignment list $list_a$ (which corresponds to Task 2 of the CMEA dataset). Entities to be fused are filtered out from ikg, and the remaining entities are enhanced using $chunks$ (context) and relevant entities from kg, resulting in the enhanced image knowledge graph ikg_e.

Next, using the image itself, we perform a detailed search for text entities that can be aligned from chunks. This involves extracting relevant text segments from chunks that are semantically related to the entities in ikg_e. We use visual features from the image to guide the search, ensuring that the extracted text entities are contextually coherent with the visual content. Once these text entities are identified, matching is performed in kg to achieve Task 2 of the CMEA dataset. If aligned entity is found, only the relationships of ikg_e will be updated to obtain ikg_u, which naturally achieves alignment during fusion. If no aligned entity is found in kg, ikg_e will be updated by supplementing a new entity and relationships obtained from chunks to form ikg_u. Finally, kg and ikg_u are fused based on the results of $list_a$ to obtain the final $mmkg$ (multi-modal knowledge graph).

The second step, entity enhancement, is achieved using the reasoning capabilities of LLM. Based on the context information, text entities from the text modality are used to supplement image entities that do not have aligned counterparts. The prompt to enhance entities is as follows:

Prompt for Enhancing Image Entities

The goal is to enrich and expand the knowledge of the image entities listed in the img_entity_list based on the provided chunk_text.

The entity_type should remain unchanged, but you may modify the entity_name and description fields to provide more context and details based on the information in the chunk_text.

For each entry in the img_entity_list, the following actions should be performed:

1. Modify and enhance the entity_name if necessary.

2. Expand the description by integrating relevant details and insights from the chunk_text.

3. Include an original_name field to capture the original entity name before enhancement.

Ensure the final output is in valid JSON format, only including the list of enhanced entities without any additional text.

After generating candidate entities, LLM is used to align image entities, with the specific prompt as follows:

Prompt for Generating Image Feature Block Description

You are an expert system designed to identify matching entities based on semantic similarity and context. Given the following inputs:

img_entity: The name of the image entity to be evaluated.

img_entity_description: A description of the image entity.

chunk_text: Text surrounding the image entity providing additional context.

possible_image_matched_entities: A list of possible matching entities. Each entity is represented as a dictionary with the following fields:

entity_name: The name of the possible entity.

entity_type: The type/category of the entity.

description: A detailed description of the entity.

additional_info: Additional relevant information about why choose this entity (such as similarity, reason generated by LLM, etc.).

-Task-

Using the information provided, determine whether the img_entity matches any of the entities in possible_image_matched_entities. Consider the following criteria:

1.Semantic Matching: Evaluate the semantic alignment between the img_entity and the possible matching entities, based on their names, descriptions, and types. Even without a similarity score, assess how well the img_entity matches the attributes of each possible entity.

2.Contextual Relevance: Use the chunk_text and img_entity_description to assess the contextual alignment between the img_entity and the possible entity.

-Output-

If a match is found, only return the entity_name of the best-matching entity.

If no match meets the criteria (e.g., low similarity or poor contextual fit), only output "no match".

Do not include any explanations, reasons, or additional information in the output.

J. Supplementary Results of MM Document QA Experiments

This experiment selects a variety of single-modal large models and multi-modal large models as comparison benchmarks, including single-modal models such as Llama3.1-70B-Instruct[4] (L), Qwen2.5-72B-Instruct[69] (Q), and Mistral-Large-Instruct-2411[5] (M), as well as multi-modal models such as Ovis1.6-Gemma2-27B[45] (Ovis), Qwen2-VL-72B[63] (Qvl), and InternVL2.5-38B-MPO[12] (Intvl).

Among them, Ovis1.6-Gemma2-27B is deployed using the AutoModelForCausalLM from the Transformers[65] library, InternVL2.5-38B-MPO is deployed using lmdeploy[12], and the other models are deployed using vllm[30].

The complete results for the DocBench dataset are shown in Table S5, the complete results for the MMLongbench dataset are shown in Table S6, and the results for the MMLongBench dataset by domain are shown in Table S7.

Model	Type					Domain			Overall Acc.
	Aca.	Fin.	Gov.	Laws	News	Text.	Multi.	Una.	
LLM-based Methods									
Llama	43.9	13.5	53.4	44.5	79.7	52.9	18.8	**81.5**	44.7
Qwen	41.3	16.3	50.7	49.7	77.3	53.9	20.1	75.8	44.8
Mistral	32.3	13.2	43.9	36.1	58.1	43.0	14.6	70.2	36.0
MMLLM-based Methods									
Ovis	16.2	11.1	23.6	25.7	39.0	22.8	8.8	54.8	21.3
Qvl	17.5	14.9	25.0	34.6	48.8	34.0	8.4	40.3	25.4
Intvl	19.8	16.3	28.4	31.4	46.5	35.7	15.9	39.5	27.7
NaiveRAG-based Methods									
Llama	43.6	38.2	66.2	64.9	**80.2**	79.9	32.1	70.2	61.0
Qwen	43.6	34.4	62.8	65.4	75.0	81.6	30.5	67.7	59.5
Mistral	44.9	35.4	58.1	62.3	76.7	76.5	32.1	69.4	58.6
GraphRAG-based Methods									
Llama	40.6	27.1	56.8	59.7	75.0	73.5	24.4	76.6	54.7
Qwen	39.6	25.7	52.5	49.7	74.5	71.7	26.0	67.5	52.3
Mistral	37.0	28.8	59.2	61.1	75.6	67.7	26.1	76.5	49.2
MMGraphRAG-based Methods (Ours)									
L-Ovis	49.7	43.6	58.5	60.0	75.3	75.1	62.5	76.3	64.1
Q-Ovis	50.3	46.4	59.4	56.1	76.8	76.3	63.4	74.2	65.3
M-Ovis	47.9	40.6	58.6	59.3	75.2	72.6	58.6	74.1	60.9
L-Qvl	51.8	59.4	62.8	60.7	77.9	79.1	77.8	70.2	74.0
Q-Qvl	51.8	62.9	**66.9**	68.6	76.2	**82.4**	81.1	67.7	75.2
M-Qvl	48.4	52.8	57.9	62.7	74.5	77.0	75.4	69.6	73.9
L-Intvl	**60.7**	64.1	62.6	64.9	76.2	80.0	86.4	75.0	**77.5**
Q-Intvl	60.5	**65.8**	66.5	**70.4**	77.1	81.2	**88.7**	71.9	76.8
M-Intvl	56.4	58.1	58.0	60.2	75.2	76.6	84.9	73.3	75.7

Table S5. Complete Results of DocBench Dataset. Based on the experimental results from a total of 21 combinations of the six models, designating Llama3.1-70B-Instruct as the evaluation model does not show undue favoritism towards its own generated results, thereby avoiding erroneous evaluation outcomes.

Through comparative experiments with other single-modal and multi-modal models under various methods, we found that Llama3.1-70B-Instruct is capable of maintaining a degree of independence and objectivity during the evaluation process. This suggests that its evaluation mechanism can effectively distinguish the generation results of different models without bias arising from its own model origin. Therefore, it can be concluded that the evaluation conclusions based on Llama3.1-70B-Instruct are relatively reliable and can provide fair and accurate assessment results in multi-model docment QA experiments.

Model	Locations			Modalities					Overall	
	Sin.	Mul.	Una.	Cha.	Tab.	Txt.	Lay.	Fig.	Acc.	F1
LLM-based Methods										
Llama	23.7	20.3	51.6	16.3	12.7	32.1	21.9	17.4	28.2	23.0
Qwen	22.5	20.0	53.2	16.8	12.6	33.3	23.5	16.1	27.8	22.1
Mistral	19.1	19.2	38.6	15.6	9.4	28.9	19.2	16.2	23.1	19.2
MMLLM-based Methods										
Ovis	10.6	9.5	13.0	6.9	3.1	10.0	15.4	15.4	10.7	9.2
Qvl	10.8	9.9	8.1	7.6	5.4	10.0	8.8	12.7	10.0	9.5
Intvl	13.3	7.9	13.9	8.8	8.1	10.7	11.8	11.4	11.6	10.4
NaiveRAG-based Methods										
Llama	24.9	19.5	56.5	16.3	15.3	31.0	22.7	18.7	29.2	24.2
Qwen	22.3	16.4	52.5	15.1	10.8	30.3	20.3	14.9	26.2	20.9
Mistral	21.6	19.2	52.9	15.4	12.5	31.2	20.6	13.7	27.3	22.8
GraphRAG-based Methods										
Llama	16.3	12.3	78.5	7.6	6.7	25.1	15.0	10.6	27.2	18.2
Qwen	18.2	13.2	77.1	14.0	11.0	26.1	12.2	8.5	28.1	19.3
Mistral	13.8	10.6	**86.5**	9.0	5.2	21.7	13.4	7.6	27.2	16.8
MMGraphRAG-based Methods (Ours)										
L-Ovis	36.9	13.0	57.4	24.4	23.7	26.8	15.9	24.1	31.0	26.8
Q-Ovis	37.4	15.5	54.4	24.9	23.3	28.0	26.1	27.6	31.6	27.5
M-Ovis	34.7	12.0	60.0	25.6	21.2	25.7	14.8	22.1	29.5	25.5
L-QVL	37.6	13.8	55.2	26.4	29.2	26.4	13.8	21.2	32.6	28.1
Q-Qvl	38.7	20.1	51.6	26.2	30.0	29.3	**29.1**	29.2	34.8	30.4
M-Qvl	36.4	12.1	56.8	27.6	27.7	23.2	11.8	19.8	31.7	26.9
L-Intvl	38.7	21.9	59.2	**34.7**	**36.6**	31.9	12.9	28.5	36.9	32.4
Q-Intvl	**39.6**	**26.7**	55.8	34.7	36.5	**33.8**	28.7	**34.6**	**38.8**	**34.1**
M-Intvl	37.7	19.5	63.2	35.0	33.5	28.0	13.6	27.6	35.7	31.8

Table S6. Complete Results of MMLongBench Dataset. The GraphRAG-based method has a similar accuracy to the NaiveRAG-based method, but the F1 score is much different. This is because GraphRAG performs better in answering the Una category questions, which are the questions that cannot be answered. Therefore, its overall performance is somewhat worse. The MMGraphRAG-based method significantly outperforms both in accuracy and F1 score, indicating that it can better answer general questions and achieves better overall results.

The MMLongBench dataset encompasses a diverse range of document domains. These domains include Research Reports/Introductions (Int.), which typically feature

Model	Evidence Locations							Overall	
	Int.	Tut.	Aca.	Gui.	Bro.	Adm.	Fin.	Acc.	F1
LLM-based Methods									
Llama	33.5	32.1	27.7	24.7	22.0	31.1	18.8	28.2	23.0
Qwen	31.5	31.8	25.4	30.3	26.5	**36.6**	9.6	27.8	22.1
Mistral	28.2	25.7	19.3	22.0	22.9	32.8	8.4	23.1	19.2
MMLLM-based Methods									
Ovis	7.4	20.4	10.9	7.3	16.5	14.3	4.3	10.7	9.2
Qvl	7.8	20.1	6.8	10.6	9.8	11.6	7.2	10.0	9.5
Intvl	11.3	19.1	10.4	8.9	14.4	13.3	6.0	11.6	10.4
NaiveRAG-based Methods									
Llama	34.0	31.0	30.0	29.4	23.0	29.6	18.4	29.2	24.2
Qwen	30.0	27.6	25.9	31.1	17.0	32.1	12.8	26.2	20.9
Mistral	32.6	24.7	24.5	32.5	18.7	34.0	17.6	27.3	22.8
GraphRAG-based Methods									
Llama	30.8	27.0	25.0	29.7	24.0	34.4	16.7	27.2	18.2
Qwen	34.6	21.8	24.8	29.2	27.0	<u>36.2</u>	18.9	28.1	19.3
Mistral	34.4	22.3	24.9	28.1	26.0	31.6	15.5	27.2	16.8
MMGraphRAG-based Methods (Ours)									
L-Ovis	35.7	32.9	30.2	<u>41.7</u>	27.1	24.6	26.7	31.0	26.8
Q-Ovis	36.4	**38.9**	29.4	39.3	<u>32.2</u>	30.2	30.1	31.6	27.5
M-Ovis	36.5	34.2	29.4	38.3	31.5	22.5	25.5	29.5	25.5
L-Qvl	36.5	32.9	27.1	41.3	26.3	28.4	34.0	32.6	28.1
Q-Qvl	37.2	<u>38.8</u>	26.3	39.5	31.6	35.8	<u>38.0</u>	34.8	30.4
M-Qvl	37.0	35.2	26.5	38.8	30.9	26.4	32.2	31.7	26.9
L-Intvl	42.4	34.0	**36.8**	**42.8**	28.9	26.0	36.2	<u>36.9</u>	<u>32.4</u>
Q-Intvl	**43.3**	36.5	35.2	40.0	**32.7**	33.5	**41.0**	**38.8**	**34.1**
M-Intvl	<u>42.9</u>	35.0	<u>35.6</u>	38.9	31.1	23.2	35.2	35.7	31.8

Table S7. MMLongBench Dataset Domain Results. The MMGraphRAG method achieves the best performance across all domains except for the Administration/Industry file category. Notably, it demonstrates the most significant improvements in the Guidebook and Financial report categories, which are characterized by a high volume of charts ,tables and figures. These enhancements are far more pronounced than those of other methods.

academic or industry-oriented analyses and background information; Tutorials/Workshops (Tut.), focusing on instructional content for skill development or knowledge dissemination; Academic Papers (Aca.), containing scholarly research and findings; Guidebooks (Gui.), offering practical information and advice for specific topics or activities; Brochures (Bro.), designed for promotional or informational purposes in a concise format; Administration/Industry Files (Adm.), covering official documents or industry-specific reports; and Financial Reports (Fin.), presenting financial data and analyses.