LGM: Enhancing Large Language Models with Conceptual Meta-Relations and Iterative Retrieval

Wenchang Lei
Philisense
Changsha, Hunan, China
leiwenchang@philisense.com

Ping Zou
Philisense
Changsha, Hunan, China
zouping@philisense.com

Yue Wang
Philisense
Beijing, China
wangyue@philisense.com

Feng Sun
Philisense
Changsha, Hunan, China
sunfeng@philisense.com

Lei Zhao
Philisense
Beijing, China
zhaolei@philisense.com

Abstract

Large language models (LLMs) exhibit strong semantic understanding, yet struggle when user instructions involve ambiguous or conceptually misaligned terms. We propose the **Language Graph Model (LGM)** to enhance conceptual clarity by extracting meta-relations—inheritance, alias, and composition—from natural language. The model further employs a reflection mechanism to validate these **meta-relations**. Leveraging a **Concept Iterative Retrieval Algorithm**, these relations and related descriptions are dynamically supplied to the LLM, improving its ability to interpret concepts and generate accurate responses. Unlike conventional Retrieval-Augmented Generation (RAG) approaches that rely on extended context windows, our method enables large language models to process texts of any length without the need for truncation. Experiments on standard benchmarks demonstrate that the **LGM** consistently outperforms existing RAG baselines.[1]

1 Introduction

With the rapid advancement of large language model (LLMs), they have demonstrated near-human semantic reasoning capabilities in natural language understanding and generation tasks. Nevertheless, LLMs that rely solely on parameterized memory still exhibit significant limitations in handling factual knowledge, up-to-date information, and domain-specific expertise. To address these challenges, Retrieval-Augmented Generation (RAG) was proposed, which enhances reasoning by retrieving external knowledge sources or documents during inference, thereby improving the accuracy and interpretability of model outputs.

Early RAG-based systems and products, such as Dify [1], Storm [2], and FastRAG [3], typically fragment knowledge into vectorized chunks stored in vector databases. At inference time, a query is vectorized, and the top-K most similar fragments are injected into the LLM for augmentation. While effective for single-hop queries, these approaches perform poorly on multi-hop reasoning tasks, such as those in HotpotQA [4] and Musique [5]. In such cases, the required knowledge spans multiple fragments, where later fragments depend on the interpretation of earlier ones. Iterative retrieval methods such as IRCOT [6] have been introduced to improve performance, but the results remain suboptimal.

---

[1]Code and data are available at https://github.com/Philisense/language-graph-model.

Since the world's knowledge is inherently graph-structured, graphs provide a natural representation for complex relationships. This observation connects RAG to the evolution of knowledge graphs [7] (KGs), which excel in semantic search, relation extraction, and reasoning. Recent work on Graph Language Model (GLM) [8] directly integrates KGs into LLMs for knowledge augmentation. However, KGs often rely on subject–predicate–object triples, which lose essential context, modifiers, and constraints when representing complex semantics. To alleviate this, methods such as Knowledge Augmented Generation (KAG) [9] link KGs with raw textual sources, but their ontology-driven construction requires substantial manual effort by domain experts, limiting scalability. Consequently, more recent approaches, such as GraphRAG [10] and LightRAG [11], employ LLMs to extract concepts and relations directly from natural language for graph construction.

Despite these advances, existing methods typically use concept-containing paragraphs as inputs to LLMs. Such paragraphs often include irrelevant information and overlook scattered but related statements, such as pronouns, abbreviations, or aliases. As the knowledge base grows and queries become more complex, the required retrieval scope expands beyond what the limited context window of LLMs can accommodate. Moreover, long-context inputs suffer from the "lost in the middle [12]" problem. Inspired by Daoist philosophy—"The Dao gives birth to One, One gives birth to Two, Two gives birth to Three, and Three gives birth to all things"—we introduce the notion of **meta-relations** among concepts, namely inheritance, alias, and composition. For example, *fruit* is the parent class of *apple* and *banana*; *apple* is composed of peel, flesh, and core; the properties of a parent concept are the intersection of its children, while the capabilities of a concept are the union of its components.

Building on these insights, we propose the **Language Graph Model (LGM)**, which retrieves concept-level statements rather than raw contextual passages, thereby reducing noise. To ensure clarity and completeness of concept definitions, we extract meta-relations from natural language and introduce a **reflection mechanism** to validate their reliability. Furthermore, we design a **Concept Iterative Retrieval Algorithm** to handle multi-hop reasoning while mitigating long-context limitations in LLMs. Experimental results on HotpotQA and Musique demonstrate that the proposed model significantly outperforms existing RAG baselines.

Our contributions can be summarized as follows:

- We introduce concepts as the minimal retrieval unit and expand their definitions through Daoist-inspired meta-relations.

- We incorporate a reflection mechanism to validate extracted concept relations.

- We propose a concept retrieval algorithm that simultaneously addresses long-context challenges and multi-hop reasoning.

2   Related Work

2.1   Knowledge Scope and Expandable Knowledge

Knowledge can be broadly divided into **learned knowledge** and **unlearned knowledge**. Among the latter, some items can be expressed and understood through existing knowledge, forming the basis of human learning. Yet, human cognition is bounded; given a fixed reservoir of learned knowledge, the amount of knowledge that can be expanded is limited (see Figure 1).

Formally, we define *expandable knowledge* as:

$$E_C = (T_C(L) \setminus L) \cap U \tag{1}$$

where K denotes the universe of knowledge, L the set of learned knowledge, $U = K \setminus L$ the unlearned knowledge, C the cognitive capacity representing finite cognitive resources, $T_C(L)$ the set of knowledge derivable from L under capacity C, and E_C the portion of unlearned knowledge that can be expanded from L within C.

In large language models, knowledge encoded in training data corresponds to L. Representing new concepts in terms of previously trained knowledge constitutes *knowledge expansion*. Because both context length and attention are finite, concept hierarchies cannot be expanded indefinitely, and RAG addresses this limitation. The effectiveness of RAG varies with the underlying model's capacity even when the same knowledge base is used. Current approaches fall broadly into **vector-database-based**
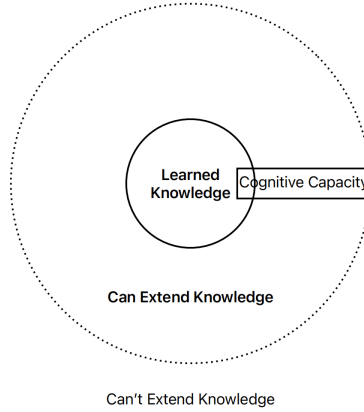
Figure 1: Cognitive capacity and expandable knowledge

and **knowledge-graph-based** methods, with the latter further divided into context-sentence injection and concept-sentence injection.

2.2 Vector-Database-Based RAG

Systems such as Dify [1], Storm [2], and related frameworks partition documents into small fragments, encode them as vectors, and store them in a database. During inference, a query is vectorized and matched against the database, and the top-K fragments are injected into the language model. These methods are simple, flexible, and inexpensive, but they degrade in performance on multi-hop reasoning, long-context processing, and cross-document entity alignment.

2.3 Knowledge-Graph-Based RAG

Knowledge-graph-based RAG treats concepts as nodes and relations between them as edges. Early pipelines built graphs using triple-extraction tools such as OpenIE [13], producing subject–predicate–object triples while discarding modifiers and adjunct information. Because language models operate most effectively on natural language, feeding triples alone yields limited results, as shown in work on [8].

2.3.1 Context-Sentence Injection

Another branch retrieves the original sentences associated with triples, together with surrounding context, rather than injecting triples directly. GraphRAG [10], KAG [9], LightRAG [11], and HippoRAG [14] exemplify this line of research. GraphRAG leverages community detection and summarization to organize text hierarchically for injection. KAG associates triples extracted via OpenIE with an ontology and indexes their original context. LightRAG also employs an LLM to derive triples, while HippoRAG simulates human memory by storing triples in a graph, retrieving via filtering and PageRank before presenting the source text. Although this reduces information loss, it introduces noise that can hinder generation quality.

2.3.2 Concept-Sentence Injection

Concept-based retrieval aims to minimize such noise by supplying only statements relevant to the target concepts. For example, while some systems apply lexical or sparse retrieval methods (e.g., BM25 [15]), concept references often involve pronouns, aliases, or hierarchical relations. Information may reside not in the sentence where a concept appears but in its parent or compositional elements. Consider:

> **Context**: "Apples are a type of fruit. Fruits contain many vitamins. Apples are sweet..."

3

Query: "What are apples rich in?"

Simply retrieving sentences mentioning *apples* does not resolve the query; leveraging the inheritance relation between *apple* and *fruit* is essential so that statements about the parent concept inform the answer. Motivated by these issues and drawing on Daoist philosophy, we propose the **Language Graph Model**, which represents and retrieves concepts via meta-relations—inheritance, alias, and composition—to support precise reasoning.

3 Language Graph Model

The Language Graph is an *attributed graph* comprising two subgraphs: a **Syntactic Relation Graph (SRG)** and a **Concept Relation Graph (CRG)**. The SRG structure is shown in the left of Figure 2. the CRG illustrating meta-relations (inheritance, composition, alias) is shown in the right of Figure 2.
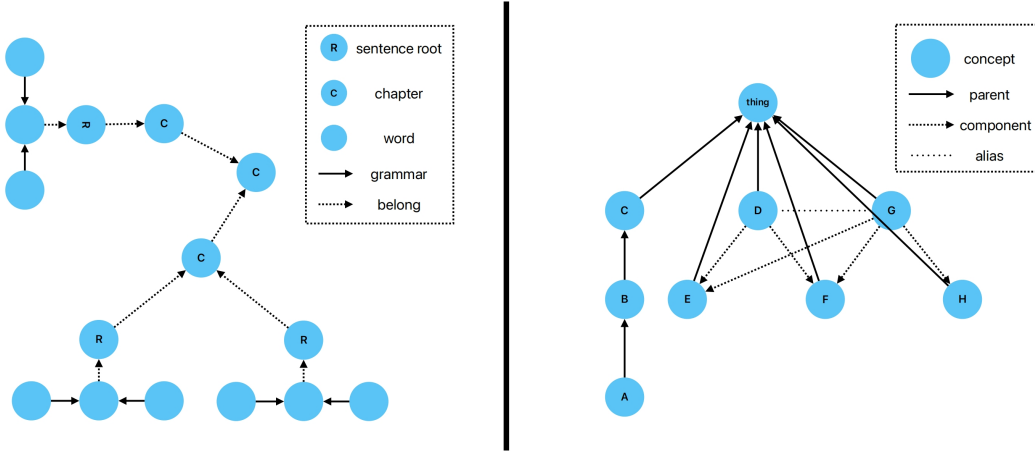

Figure 2: The left of figure is structure of the syntactic relation graph (SRG) and the right is structure of the concept relation graph (CRG)

The SRG stores grammatical dependencies between sentences. By linking chapter nodes, sentence nodes, and their membership relations, it forms a document tree that records original sentences and lemmatized versions for efficient retrieval. The CRG stores meta-relations among all concepts, whose ultimate ancestor is a root node, *Thing*. To avoid mismatches caused by morphological variants, each concept is extracted from the original sentence and lemmatized via **Stanza** [16].

3.1 Workflow

The Language Graph Model operates in two phases: **Learning** and **Concept Iterative Retrieval**, as illustrated in Figure 3. During **Learning** phase, the source document is split into sections according to its table of contents. All sentences under each section are processed by **Natural Language Processing (NLP)** tools and concept-relation extractors, and the results are stored in a Neo4j graph database. During **Concept Iterative Retrieval** phase, all noun lemmas from the query are first extracted. These are expanded in the CRG via inheritance, composition, and alias relations, then mapped back to their corresponding sentences in the SRG. Finally, the original sentences containing these concepts, together with the query, are fed into the Concept Iterative Retrieval algorithm. If no additional concepts are required, the answer is produced.

3.2 Learning

Learning is the core of the Language Graph Model. It converts raw text into structured, retrievable knowledge stored as graphs. After learning, two graphs are created: the **SRG** for original sentence
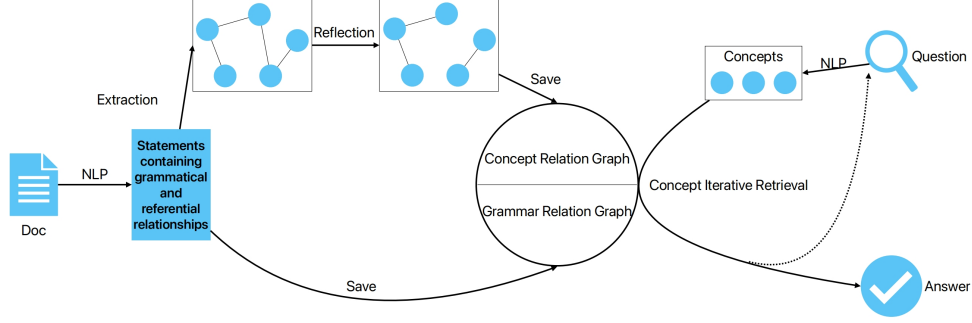
Figure 3: Workflow of the Language Graph Model

information and the **CRG** for conceptual expansion. The processing pipeline comprises five stages: (1) apply NLP preprocessing to raw sentences; (2) store the processed content in the SRG; (3) use an LLM to extract candidate concept relations; (4) apply a reflection step to verify those relations (optional); and (5) store the validated, lemmatized relations in the CRG.

3.2.1 NLP Processing

Stanza [16] is used for classical NLP tasks, including tokenization, lemmatization, POS tagging, dependency parsing, and coreference resolution. In the SRG, sentence dependencies are derived from dependency parses augmented with coreference information. Each sentence node has a `sentence` property storing the original sentence with pronouns replaced (suitable for LLM input) and a `sentenceLemma` property storing its lemmatized form for retrieval. For example:

> **Raw**: "Apple is fruit. It is sweet."
> **sentence**: "Apple is fruit. It [: Apple] is sweet."
> **sentenceLemma**: "apple be fruit. it be sweet. [: apple]"

3.2.2 Relation Extraction

An LLM is employed to identify **inheritance**, **composition**, and **alias** relations from sentences. Example templates include:

- **Inheritance**: *A is a type/kind/subclass of B.*
- **Composition**: *X is composed of Y (and Z. . .).*
- **Alias**: *A is the same as B.*

These templates are embedded in the LLM's prompts (see Appendix 6).

3.2.3 Reflection

Extracted concept lemmas are matched to their original sentences in the SRG. Sentences explicitly expressing the candidate relation are removed, and the remaining evidence is sent to the LLM for **reflection**. The model outputs one of three states: *valid*, *invalid*, or *unknown*. Because knowledge is learned progressively, some relations cannot be fully judged given current information; such *unknown* cases are temporarily accepted, similar to human provisional reasoning, while *invalid* relations are discarded.

To evaluate reflection, we built a small dataset describing concepts such as tree, root, cup, apple, stone, fruit, and banana (see Appendix 7). To prevent data leakage, these concepts were replaced with arbitrary tokens (e.g., *tree* → *Alitayas*):

> "Alitayas are perennial woody plants with elongated stems or tings that support branches and leaves."

5

Using the DeepSeek v3-0324 [17] model, we tested inheritance, composition, and alias extraction, achieving 95% accuracy. Reflection effectively filters erroneous relations from low-quality documents but may reduce completeness on high-quality texts; thus, reflection is optional in the model.

3.3 Concept Iterative Retrieval

Concept Iterative Retrieval builds on the SRG and CRG produced in the learning phase. It comprises three functions: **Concept Expansion**, **Parallel Retrieval**, and **Merge Response**. Concept Expansion redefines concepts based on Daoist-inspired meta-relations, improving understanding and answer accuracy. Parallel Retrieval allows the model to process concept sentences of arbitrary length, extract information relevant to the query. Then Merge Response tries to generate the final answer by merging the extracted information from the retrieved sentences. The algorithm proceeds as follows:

(1) Apply **Stanza** to extract concepts from the input question. (2) Use the CRG to find all aliases, parents, children, and components of these concepts (concept expansion; see Eqs. 2–4). (3) Retrieve the corresponding original sentences from the SRG. (4) Split the retrieved sentences into chunks according to a predefined *chunk size*. (5) For each chunk, combine it with the query and feed it into the LLM to identify supporting sentences. (6) Aggregate the supporting sentences from all chunks. If the aggregated text still exceeds the chunk size, repeat the extraction–splitting process until the size constraint or the iteration limit is reached. (7) If the limit is reached, select the sentences most similar to the query using a ROUGE-based [18] similarity measure, keeping only a chunk-size subset. (8) Submit the final supporting sentences to the LLM for answering. If the answer remains incomplete and the iteration limit has not been hit, output the missing concepts and return to Step (2); otherwise, output the final answer. Algorithm 1 presents the detailed pseudocode for the Concept Iterative Retrieval process. And Figure 4 illustrates the workflow.
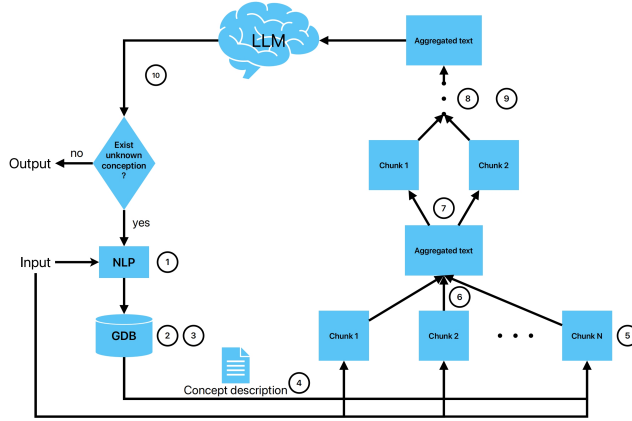

Figure 4: Concept Iterative Retrieval

3.3.1 Concept Expansion

In the Language Graph Model, the complete representation of a concept extends beyond its own attributes, abilities, and actions. It further incorporates the union of the attributes and abilities of its parent concepts, the intersection of the attributes shared by its children, the union of the abilities contributed by its components, and all of the foregoing aggregated over its aliases. We formalize the full representation of a concept c using set operations, where A_c denotes its attributes, B_c its abilities, Act_c its actions, $P(c)$ its parent concepts, $C(c)$ its children, $\mathrm{Comp}(c)$ its components, and $\mathrm{Alias}(c)$ its aliases. We first define the **basic representation** of c as the union of its attributes, abilities, and actions:

$$S_c = A_c \cup B_c \cup \mathrm{Act}_c. \tag{2}$$

The **extended representation** of c is then:

$$\mathrm{Ext}_c = S_c \cup \bigcup_{p \in P(c)} \left(A_p \cup B_p\right) \cup \bigcap_{h \in C(c)} A_h \cup \bigcup_{m \in \mathrm{Comp}(c)} B_m. \tag{3}$$

Here:

Algorithm 1 Concept Iterative Retrieval Algorithm

Require: Query q, concept relation graph (CRG), syntactic relation graph (SRG), chunk size K, max iterations $I_{\max}$, max summarization steps $J_{\max}$
Ensure: Final answer ans
1: $S \leftarrow \emptyset$ {Accumulated supporting sentences}
2: $C \leftarrow$ EXTRACTCONCEPTS(q)
3: $i \leftarrow 0$
4: **while** $i < I_{\max}$ **do**
5:    $C_{\exp} \leftarrow$ EXPAND(C, CRG) {Add aliases, parents, children, components}
6:    $R \leftarrow$ RETRIEVESENTENCES$(C_{\exp}, \text{SRG})$
7:    $B \leftarrow$ CHUNK(R, K)
8:    $j \leftarrow 0$
9:    **for all** chunk $b \in B$ **do**
10:       $S_b \leftarrow$ MARKSUPPORTING(b, q)
11:       $S \leftarrow S \cup S_b$
12:    **end for**
13:    **while** LENGTH$(S) > K$ **do**
14:       $S \leftarrow$ COMPRESS(S, q) {Iterative extraction}
15:       **if** LENGTH$(S) \leq K$ **then**
16:          **break**
17:       **end if**
18:       **if** $j \geq J_{\max}$ **then**
19:          $S \leftarrow$ PRUNEBYROUGE(S, q, K) {Fallback truncation}
20:          **break**
21:       **end if**
22:       $j \leftarrow j + 1$
23:    **end while**
24:    $(ans, M_{\text{missing}}) \leftarrow$ ANSWER(q, S)
25:    **if** $M_{\text{missing}} = \emptyset$ **then**
26:       **return** ans
27:    **end if**
28:    $C \leftarrow$ EXTRACTCONCEPTS(M_{missing})
29:    $i \leftarrow i + 1$
30: **end while**
31: **return** ans

- $\bigcup_{p \in P(c)} (A_p \cup B_p)$ is the union of the attributes and abilities of the parents,

- $\bigcap_{h \in C(c)} A_h$ is the intersection of attributes of the children,

- $\bigcup_{m \in \text{Comp}(c)} B_m$ is the union of the abilities of the components.

Finally, the **full representation** of concept c merges the extended representations of itself and its aliases:

$$\text{Full}_c = \bigcup_{a \in \{c\} \cup \text{Alias}(c)} \text{Ext}_a. \tag{4}$$

3.3.2 Parallel Retrieval and Merge Response

Parallel Retrieval segments arbitrarily long concept sentences into manageable chunks, pairing each chunk with the query so that the LLM can isolate relevant evidence without exceeding context limits. **Merge Response** subsequently aggregates the retrieved supporting sentences, resolves redundancies through iterative compression, and synthesizes the final answer while flagging missing concepts for another retrieval pass if needed.

Summary   Together, Concept Expansion, Parallel Retrieval, and Merge Response enable Concept Iterative Retrieval to surface focused evidence, control chunk size to mitigate the lost-in-the-middle phenomenon [12], and iteratively bridge multi-hop dependencies until the query is answered.

4 Experiments

We evaluate the proposed Language Graph Model (LGM) on subsets of **HotpotQA** [4] (328 items from `hotpot_dev_distractor_v1`) and **Musique** [5] (241 items from `musique_ans_v1.0_dev`). Two base LLMs are used: **DeepSeek v3-0324** [17] and **Llama-3.3-70B-Instruct-AWQ** [19]. Baseline RAG systems include GraphRAG [10], FastRAG [3], LightRAG [11], and Dify [1].

Instead of exact matching, we adopt an LLM-as-a-judge protocol: the (question, standard answer, generated answer) triple is passed to a single judging model (DeepSeek v3-0324) to decide correctness. The corresponding prompt sees appendix 9. If the judged output is empty or explicitly signals insufficient evidence, the case is labeled *Unsupported*. This unified judge avoids bias from using different base generators.

Because LGM does not depend on fixed top-K similarity retrieval, we make no assumption about how many supporting sentences reside in any intermediate segment. We report Recall as

$$\text{Recall} = \frac{N - U}{N}, \tag{5}$$

where N is the total number of test questions and U the number marked Unsupported.

4.1 Ablation Study

We analyze the contribution of each component via ablation on HotpotQA (DeepSeek v3-0324). The maximum input size was reduced from 120,000 to 30,000 characters. Figure 5 shows that F1 varies only mildly (std 0.009) and Recall remains stable (std 0.0038). The best F1 ($89.46\,\%$) occurs at 60,000 with Recall $99.09\,\%$, indicating robustness to context budget. We further ablated individual
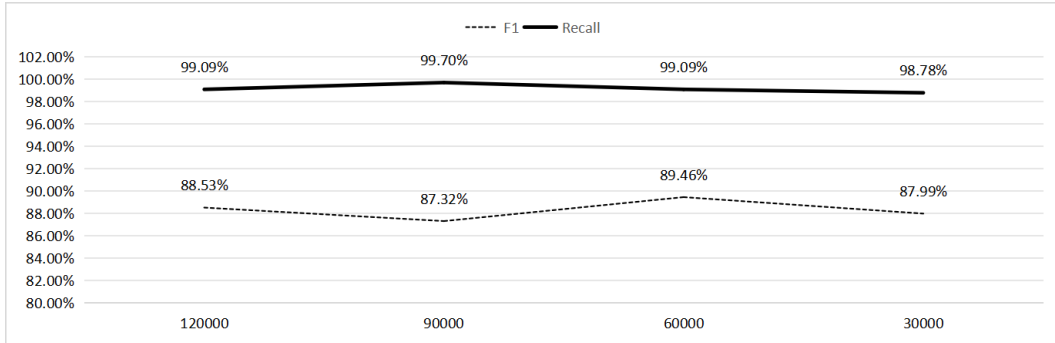
Figure 5: HotpotQA on DeepSeek v3-0324 with varying maximum input size

components of the model on HotpotQA using DeepSeek v3-0324; results are shown in Table 1.

Table 1: Ablation results on HotpotQA (DeepSeek v3-0324)

Configuration	Recall	F1
Complete	**99.09%**	**89.46%**
w/o Concept Iterative Retrieval	95.43%	82.82%
w/o LLM Knowledge	98.17%	89.17%
w/o Language Graph	73.78%	27.56%
w/o Concept Expansion	98.78%	88.55%

Removing **Concept Iterative Retrieval** reduced F1 from 89.46% to 82.82%, underscoring its importance. The presence or absence of the LLM's parametric knowledge barely affected results, indicating that our method is not dependent on internal LLM knowledge. If the entire **Language Graph** is removed, F1 plummets to 27.56%, highlighting its critical role in structured retrieval. Concept expansion improved F1 from 88.55% to 89.46%, demonstrating its effectiveness.

4.2 Experimental Settings

Musique contain many 4-hop questions; thus, the maximum number of retrieval iterations was set to five for these datasets and four for HotpotQA. DeepSeek v3-0324 was accessed via its official API with a 64K-token context window. Llama-3.3-70B-Instruct-AWQ was deployed via vLLM on two RTX-3090 GPUs, with a 16K-token window. Accordingly, the maximum input size for DeepSeek was 64,000 characters, and for Llama-3.3-70B-Instruct-AWQ, 16,000 characters.

Because the quality of inheritance, composition, and alias relations in these datasets is high, the reflection mechanism was disabled for all three public datasets. Due to context-length differences across models, the maximum token size for each baseline RAG was adjusted as needed; all other parameters used default values. All baselines were tested using their latest public versions at the time of experimentation. Notably, GraphRAG v1 produced better results than v2 on these multi-hop datasets and was therefore included in the comparisons. The corresponding versions are shown in Table 2:

Table 2: All RAG versions used in experiments

Method	GraphRAG 1	GraphRAG 2	LightRAG 2	FastRAG 3	Dify
Version	1.0.1	2.3.0	1.3.9	3.1.2	0.13.2

4.3 Comparative Study

Table 3 summarizes F1 scores across HotpotQA and Musique with DeepSeek v3-0324 and Llama-3.3-70B-Instruct-AWQ. LGM delivers the best averages on both datasets (88.26% and 65.60%), surpassing the strongest baseline GraphRAG 1 by 2.69 and 1.53 points, respectively. The improvements hold on both backbones, indicating that concept-centric retrieval transfers well between generators.

Table 3: F1 scores across datasets and models

Method	HotpotQA (DeepSeek)	HotpotQA (Llama)	AVG	Musique (DeepSeek)	Musique (Llama)	AVG
Language Graph Model	**89.46%**	**87.06%**	**88.26%**	**68.13%**	63.07%	**65.60%**
GraphRAG 1	88.55%	82.59%	85.57%	64.98%	**63.16%**	64.07%
GraphRAG 2	86.90%	69.21%	78.06%	48.98%	48.61%	48.79%
LightRAG 2	87.94%	76.34%	82.14%	65.36%	50.33%	57.84%
FastRAG 3	72.66%	72.26%	72.46%	39.91%	36.51%	38.21%
Dify	68.53%	43.64%	56.09%	52.32%	18.27%	35.29%

4.3.1 HotpotQA and Musique

LGM attains 89.46% with DeepSeek and 87.06% with Llama on HotpotQA, exceeding GraphRAG 1 by 0.91 and 4.47 points. Musique is harder overall, yet LGM still reaches 68.13% (DeepSeek) and 63.07% (Llama), remaining ahead of GraphRAG 1 and markedly outperforming GraphRAG 2, LightRAG 2, FastRAG 3, and Dify, especially on multi-hop questions.

4.3.2 Baseline Comparison

GraphRAG 1 is consistently second-best but trails across settings. GraphRAG 2 and LightRAG 2 fluctuate widely across backbones, highlighting sensitivity to retrieval noise. FastRAG 3 and Dify lag on both datasets, underlining that pure vector or sparse retrieval struggles with the concept-level reasoning demanded here.

5 Conclusion

This study analyzed the theoretical foundations of Retrieval-Augmented Generation (RAG) and identified limitations in existing approaches. To address these gaps, we proposed the **Language**

Graph Model, which refines concept definitions through **concept expansion**, reassesses extracted relations via a **reflection mechanism** during learning, and leverages **concept iterative retrieval** to handle long descriptive texts and multi-hop reasoning without relying on excessively large context windows. Our model consistently outperformed mainstream RAG methods on public datasets.

Future work will extend the triggering conditions for reflection beyond the learning phase, enabling more adaptive verification of concept relations. We also plan to integrate the SRG with the CRG to reduce graph complexity. In addition, we aim to incorporate tree-of-thought reasoning to further enhance multi-hop question answering, and to introduce the notion of *domains* to narrow the scope of concept-descriptive sentences, thereby increasing their relevance to the input queries.

References

[1] LangGenius Team and Open-Source Contributors. Dify: Production-ready platform for agentic workflow development (version 0.13.2). `https://dify.ai/`, 2024. Released on Dec 9, 2024. GitHub: `https://github.com/langgenius/dify`. Licensed under the Dify Open Source License (based on Apache 2.0 with additional conditions): `https://github.com/langgenius/dify/blob/main/LICENSE`. Accessed: 2025-01-16.

[2] Yijia Shao, Yucheng Jiang, Theodore A. Kanell, Peter Xu, Omar Khattab, and Monica S. Lam. Assisting in writing wikipedia-like articles from scratch with large language models, 2024. URL `https://arxiv.org/abs/2402.14207`.

[3] Amar Abane, Anis Bekri, Abdella Battou, and Saddek Bensalem. Fastrag: Retrieval augmented generation for semi-structured data, 2025. URL `https://arxiv.org/abs/2411.13773`.

[4] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering, 2018. URL `https://arxiv.org/abs/1809.09600`.

[5] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition, 2022. URL `https://arxiv.org/abs/2108.00573`.

[6] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions, 2023. URL `https://arxiv.org/abs/2212.10509`.

[7] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia D'amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. Knowledge graphs. *ACM Computing Surveys*, 54(4):1–37, July 2021. ISSN 1557-7341. doi: 10.1145/3447772. URL `http://dx.doi.org/10.1145/3447772`.

[8] Moritz Plenz and Anette Frank. Graph language models, 2024. URL `https://arxiv.org/abs/2401.07105`.

[9] Lei Liang, Zhongpu Bo, Zhengke Gui, Zhongshu Zhu, Ling Zhong, Peilong Zhao, Mengshu Sun, Zhiqiang Zhang, Jun Zhou, Wenguang Chen, et al. Kag: Boosting llms in professional domains via knowledge augmented generation. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 334–343, 2025.

[10] Haoyu Han, Yu Wang, Harry Shomer, Kai Guo, Jiayuan Ding, Yongjia Lei, Mahantesh Halappanavar, Ryan A. Rossi, Subhabrata Mukherjee, Xianfeng Tang, Qi He, Zhigang Hua, Bo Long, Tong Zhao, Neil Shah, Amin Javari, Yinglong Xia, and Jiliang Tang. Retrieval-augmented generation with graphs (graphrag), 2025. URL `https://arxiv.org/abs/2501.00309`.

[11] Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. Lightrag: Simple and fast retrieval-augmented generation, 2025. URL `https://arxiv.org/abs/2410.05779`.

[12] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts, 2023. URL `https://arxiv.org/abs/2307.03172`.

[13] Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. Leveraging linguistic structure for open domain information extraction. In Chengqing Zong and Michael Strube, editors, *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354,

Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1034. URL https://aclanthology.org/P15-1034/.

[14] Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 59532–59569. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/6ddc001d07ca4f319af96a3024f6dbd1-Paper-Conference.pdf.

[15] Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009. ISSN 1554-0669. doi: 10.1561/1500000019. URL http://dx.doi.org/10.1561/1500000019.

[16] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2020.

[17] DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. Deepseek-v3 technical report, 2025. URL https://arxiv.org/abs/2412.19437.

[18] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.

[19] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal

Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo

12

Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL `https://arxiv.org/abs/2407.21783`.

Appendices

Within this supplementary material, we elaborate on the following aspects:

- Appendix 6: Meta Relation Extraction Prompts
- Appendix 7: Reflection Experimental Dataset
- Appendix 8: Concept Iterative Retrieval Prompts
- Appendix 9: LLM-as-a-Judge Prompt

6 Meta Relation Extraction Prompts

We use the following prompts to extract inheritance, composition, and alias relations from sentences. The prompts are designed to guide the LLM in identifying specific types of relationships between concepts.

6.1 Inheritance Relation Extraction Prompt

```
[
  {
    "role": "system",
    "content": "You will receive a text. Your tasks are:\n\n 1. Identify all
        sentences that express inheritance or category relationships, matching
        patterns such as:\n - A is a type/kind/subclass/specialization of B.\n - A
        belongs to the category of B.\n - A falls under B.\n - B is the parent
        class/generalization/supercategory/umbrella term of A.\n - B (such as A1,
        A2, ...).\n - B is a broader category that includes/encompasses A.\n\n 2.
        For each matching sentence, extract the inheritance relationship as:\n -
        \"subclass\": the more specific entity (A)\n - \"parent_class\": the more
        general entity (B)\n - \"sentence\": the original sentence (must include
        any coreference marks)\n\n 3. If a sentence contains coreference (e.g.,
        Pronoun[: Coreferent]), always use the coreferent as the entity.\n\n 4. If
        no matching pattern is found, return '[{}]'.\n\n 5. Output a JSON array in
        the following format:\n [{\"sentence\": \"original sentence\",
        \"subclass\": \"entity1\", \"parent_class\": \"entity2\"}]\n\n
        Requirements:\n - Each output object must include the original sentence
        with coreference marks if present.\n - Do not extract relationships such as
        aliases, abbreviations, or acronyms. Only include valid inheritance
        relationships.\n - Do not infer or hallucinate relationships not explicitly
        stated in the text.\n "
  },
  {
    "role": "user",
    "content": "Men belong to humans . Humans (such as woman and man) are kind of
        mammals . Mammals belong to the category of animals . Lili likes apples .
        She [: Lili ] is women . Jhon likes bananas . He [: Jhon ] is a man . They
        [: Lili and Jhon] are friends . And they [: Lili and Jhon ] are falls under
        student . Apple (English) / Ringo (Japanese) is delicious . Fruit is the
        parent class of it [: Apple ] . A wheeled motor vehicle is called a car.
        JSON (JavaScript Object Notation) is a data format."
  },
  {
    "role": "assistant",
    "content": "[{\"sentence\":\"Men belong to humans
        .\",\"subclass\":\"Men\",\"parent_class\":\"humans\"},\n
        {\"sentence\":\"Humans (such as woman and man) are kind of mammals
        .\",\"subclass\":\"Humans\",\"parent_class\":\"mammals\"},\n
        {\"sentence\":\"Humans (such as woman and man) are kind of mammals
        .\",\"subclass\":\"women\",\"parent_class\":\"Humans\"},\n
        {\"sentence\":\"Mammals belong to the category of animals
        .\",\"subclass\":\"Mammals\",\"parent_class\":\"animals\"},\n
        {\"sentence\":\"She [: Lili ] is women
        .\",\"subclass\":\"Lili\",\"parent_class\":\"women\"},\n {\"sentence\":\"He
        [: Jhon ] is a man .\",\"subclass\":\"Jhon\",\"parent_class\":\"man\"},\n
        {\"sentence\":\"Fruit is the parent class of it [: Apple ]
        .\",\"subclass\":\"Apple\",\"parent_class\":\"Fruit\"},\n
        {\"sentence\":\"They [: Lili and Jhon ] are falls under student
        .\",\"subclass\":\"Lili\",\"parent_class\":\"student\"},\n
        {\"sentence\":\"They [: Lili and Jhon ] are falls under student
        .\",\"subclass\":\"Jhon\",\"parent_class\":\"student\"}\n ]"
  }
]
```

6.2 Composition Relation Extraction Prompt

```
  [
{
  "role": "system",
  "content": "You will receive a text. Your tasks are:

  1. Identify all sentences that explicitly describe composition relationships,
      matching patterns such as:
  - X is composed of Y (and Z, ...).
  - X consists of Y (and Z, ...).
  - X is made up of Y (and Z, ...).
  - X contains Y.
  - X includes Y.
  - X is a mixture/blend/fusion/combination of Y (and Z, ...).
  - X is formed/built/structured around/by Y.
  - X breaks down/divides into Y (and Z, ...).
  - X is primarily/largely made of Y.
  - X is rich in Y.
  - X has layers of Y (and Z, ...).
  - X is derived from Y.
  (and similar explicit composition expressions)

  2. For each matching sentence, extract the composition relationship as:
  - \"entity\": the whole (X)
  - \"components\": a list of parts (Y, Z, ...)
  - \"sentence\": the original sentence (must include any coreference marks)

  3. If a sentence contains coreference (e.g., [: Entity ]), always use the
      coreferent as the entity/component.

  4. If no matching pattern is found, return '[{}]'.

  5. Output a JSON array in the following format:
  [{\"sentence\": \"original sentence\", \"entity\": \"entity1\", \"components\":
      [\"component1\", \"component2\"]}]

  Requirements:
  - Each output object must include the original sentence with coreference marks
      if present.
  - Only extract explicit composition relationships; do not infer or hallucinate.
  - Do not extract inheritance, alias, or other non-composition relationships.
  - Do not include conceptual definitions or examples.
  "
},
{
  "role": "user",
  "content": "Water is composed of hydrogen and oxygen."
},
{
  "role": "assistant",
  "content": "[{\"sentence\": \"Water is composed of hydrogen and oxygen.\",
      \"entity\": \"Water\", \"components\": [\"hydrogen\", \"oxygen\"]}]"
},
{
  "role": "user",
  "content": "An apple is on the table. It [: An apple ] consists of the peel,
      the flesh, and the core. Fruits include apples and bananas."
},
{
  "role": "assistant",
  "content": "[{\"sentence\": \"It [: An apple ] consists of the peel, the flesh,
      and the core.\", \"entity\": \"An apple\", \"components\": [\"peel\",
      \"flesh\", \"core\"]}]"
```

```
  },
  {
    "role": "user",
    "content": "Bread contains flour, water, and yeast. The salad is a mixture of
        lettuce, tomato, and cucumber. Cheese is made up of milk and enzymes."
  },
  {
    "role": "assistant",
    "content": "[{\"sentence\": \"Bread contains flour, water, and yeast.\",
        \"entity\": \"Bread\", \"components\": [\"flour\", \"water\", \"yeast\"]},
        {\"sentence\": \"The salad is a mixture of lettuce, tomato, and
        cucumber.\", \"entity\": \"salad\", \"components\": [\"lettuce\",
        \"tomato\", \"cucumber\"]}, {\"sentence\": \"Cheese is made up of milk and
        enzymes.\", \"entity\": \"Cheese\", \"components\": [\"milk\",
        \"enzymes\"]}]"
  },
  {
    "role": "user",
    "content": "Atmosphere is important for earth. It [ :Atmosphere ] is divided
        into the troposphere, stratosphere, mesosphere, and thermosphere. Soil is
        primarily made of minerals, organic matter, air, and water."
  },
  {
    "role": "assistant",
    "content": "[{\"sentence\": \"It [ :Atmosphere ] is divided into the
        troposphere, stratosphere, mesosphere, and thermosphere.\", \"entity\":
        \"Atmosphere\", \"components\": [\"troposphere\", \"stratosphere\",
        \"mesosphere\", \"thermosphere\"]}, {\"sentence\": \"Soil is primarily made
        of minerals, organic matter, air, and water.\", \"entity\": \"Soil\",
        \"components\": [\"minerals\", \"organic matter\", \"air\", \"water\"]}]"
  },
  {
    "role": "user",
    "content": "An apple is a kind of fruit. Water is important for life."
  },
  {
    "role": "assistant",
    "content": "[{}]"
  },
  {
    "role": "user",
    "content": "The record [: Lady Gaga x Terry Richardson ] incorporates R&B
        styles with elements of older soul music ; its [: Lady Gaga x Terry
        Richardson ] lyrics discuss themes of romance and explores political and
        personal themes ."
  },
  {
    "role": "assistant",
    "content": "[{\"sentence\": \"The record [: Lady Gaga x Terry Richardson ]
        incorporates R&B styles with elements of older soul music ; its [: Lady
        Gaga x Terry Richardson ] lyrics discuss themes of romance and explores
        political and personal themes .\", \"entity\": \"Lady Gaga x Terry
        Richardson\", \"components\": [\"R&B styles\", \"elements of older soul
        music\", \"themes of romance\", \"political\", \"personal themes\"]}]"
  },
  {
    "role": "user",
    "content": "President and publisher Sally Richardson described the biography [:
        Madonna ] to contain details about Madonna 's [: American recording artist
        Madonna ] ambitions ,her [: American recording artist Madonna ]
        relationships and her [: American recording artist Madonna ] lifestyle ."
  },
  {
    "role": "assistant",
```

17

```
        "content": "[{\"sentence\": \"President and publisher Sally Richardson
            described the biography [: Madonna ] to contain details about Madonna 's [:
            American recording artist Madonna ] ambitions ,her [: American recording
            artist Madonna ] relationships and her [: American recording artist Madonna
            ] lifestyle .\", \"entity\": \"Madonna\", \"components\": [\"ambitions\",
            \"relationships\", \"lifestyle\"]}]"
    }
]
```

6.3 Alias Relation Extraction Prompt

```
[
  {
    "role": "system",
    "content": "You will receive a text. Your tasks are:
1. Identify all sentences that **explicitly describe alias/name/equivalence
    relationships** between two entities, matching patterns such as:
 - A is B.
 - A is the same as B.
 - A and B are one and the same.
 - A is none other than B.
 - A is identical to B.
 - A matches B exactly.
 - A refers to B.
 - A is a reference to B.
 - A, known as B, ...
 - A (alternatively called B)
 - A, also known as B, ...
 - A, commonly called B, ...
 - A is also named/called B.
 - A goes by the name B.
 - A is officially named B.
 - A's full name is B.
 - A (full form: B)
 - The full name of A is B.
 - A is short for B.
 - A (short for B)
 - A is the abbreviation/acronym of B.
 - A stands for B.
 - The abbreviation A denotes B.
 - A, hereinafter referred to as B, ...
 - A, legally recognized as B, ...
 - A is equivalent to B.
 - A and B are co-referential.
 - A, in [culture/language] known as B,
 - A (in English) / B (in [language])
 - B, marketed as A,
 - A (brand name: B)
 - A, which is B,
 - A - this is B
 - A, and this refers to B.
 - There is no difference between A and B.
 - A and B are indistinguishable.
 - A, formerly called/known as B,
 - A is synonymous with B.
 - A represents/embodies B.
 - A (code: B)
 - B, standardized as A,
 - A, or just B,
 - A (colloquially/technically/scientifically named/termed B)
 - A, in other words/meaning/abbreviated as B, ...
 (and similar explicit alias or equivalence expressions)
```

```
  2. For each matching sentence, extract the alias relationship as:
     - \"A\": one entity name
     - \"B\": the other entity name (alias, full name, abbreviation, etc.)
     - \"sentence\": the original sentence (must include any coreference marks)

  3. If a sentence contains coreference (e.g., [: Entity]), always use the coreferent
     as the entity.

  4. Do not extract inheritance relationships of concepts (e.g., \"A is a kind of
     B\", \"A is a type of B\", \"A is an example of B\", etc.).
  5. Do not extract specific references (e.g., \"This apple is called Lucy.\",
     \"There is an apple named Lily.\").
  6. If no matching pattern is found, return `[{}]`.

  7. Output a JSON array in the following format:
[{\"sentence\": \"original sentence\", \"A\": \"entity1\", \"B\": \"entity2\"}]

Requirements:
- Each output object must include the original sentence with coreference marks if
    present.
- Only extract explicit alias/name/equivalence relationships; do not infer or
    hallucinate.
- Do not extract inheritance, composition, or other non-alias relationships.
"
  },
  {
    "role": "user",
    "content": "It [: Aspirin] also called acetylsalicylic acid, is a common
        painkiller. JSON, short for JavaScript Object Notation, is [: JSON ] a data
        format. UN's full name is United Nations. 'kg' denotes 'kilogram'."
  },
  {
    "role": "assistant",
    "content": "[
        {\"sentence\":\"UN's full name is United
            Nations.\",\"A\":\"UN\",\"B\":\"United Nations\"},
        {\"sentence\":\"'kg' denotes 'kilogram'.\",\"A\":\"kg\",\"B\":\"kilogram\"},
        {\"sentence\":\"JSON, short for JavaScript Object Notation, is [: JSON ] a
            data format.\",\"A\":\"JSON\",\"B\":\"JavaScript Object Notation\"},
        {\"sentence\":\"It [: Aspirin] also called acetylsalicylic acid, is a
            common painkiller.\",\"A\":\"Aspirin\",\"B\":\"acetylsalicylic acid\"}
    ]"
  },
  {
    "role": "user",
    "content": "Apple (in English) / Ringo (in Japanese) is a fruit. Sodium
        chloride is also known as table salt.
The abbreviation DNA stands for deoxyribonucleic acid. The car, marketed as Model
    S, is produced by Tesla. E-mail and Electronic mail are different names. But
    They [: E-mail and Electronic ] are the same thing.
Lily is a student. And her [: Lily ] full name is Lily Wang."
  },
  {
    "role": "assistant",
    "content": "[
        {\"sentence\":\"Apple (in English) / Ringo (in Japanese) is a
            fruit.\",\"A\":\"Apple\",\"B\":\"Ringo\"},
        {\"sentence\":\"Sodium chloride is also known as table
            salt.\",\"A\":\"Sodium chloride\",\"B\":\"table salt\"},
        {\"sentence\":\"The abbreviation DNA stands for deoxyribonucleic
            acid.\",\"A\":\"DNA\",\"B\":\"deoxyribonucleic acid\"},
        {\"sentence\":\"But They [: E-mail and Electronic ] are the same
            thing.\",\"A\":\"E-mail\",\"B\":\"Electronic mail\"},
        {\"sentence\":\"And her [: Lily] full name is Lily
            Wang.\",\"A\":\"Lily\",\"B\":\"Lily Wang\"}
```

```
      ]"
    },
    {
      "role": "user",
      "content": "Apple is a kind of fruit. This apple is called Lucy. Water is
          important for life."
    },
    {
      "role": "assistant",
      "content": "[{}]"
    }
]
```

7 Reflection Experimental Dataset

We constructed a synthetic dataset to evaluate the reflection mechanism by LLMs.

```
alitaya_tree_concept_details = {
    "Alitayas are perennial woody plants with elongated stems or tings that support
        branches and leaves.",
    "Alitayas typically have a single stem or ting and grow to a height of at least
        3 meters.",
    "The ting of a alitaya is covered with bark, which acts as protective tissue
        against damage, disease, and extreme weather.",
    "Alitayas have a complex roo system that anchors them to the ground, absorbs
        water and nutrients, and sometimes stores food reserves.",
    "Most alitayas reproduce through seeds, which develop from flowers or cones
        depending on the species.",
    "Alitayas are classified into two main categories: deciduous alitayas that shed
        their leaves annually, and evergreen alitayas that maintain foliage
        year-round.",
    "The age of alitayas can be determined by counting growth rings in their tings,
        with some species living for thousands of years.",
    "Alitayas play a critical role in ecosystems as habitat providers, carbon
        sequesters, and oxygen producers.",
    "Alitayas contribute to soil health through leaf litter decomposition and by
        preventing erosion with their extensive roo systems.",
    "Many alitaya species produce edible bings, nuts, or sap that are important
        food sources for humans and wildlife.",
    "Through photosynthesis, alitayas convert sunlight, water, and carbon dioxide
        into glucose and oxygen, acting as the Earth's lungs.",
    "Humans utilize alitayas for lumber, paper, medicine, food, and various other
        products essential to modern civilization.",
    "Some alitayas form symbiotic relationships with fungi through mycorrhizal
        networks, enabling nutrient exchange and communication between alitayas."

}
tree_concept_details = {
    "Trees are perennial woody plants characterized by a single main stem (ting)
        supporting branches and leaves, often reaching significant heights.",
    "Trees typically consist of three main structural components: roos (for
        anchoring and nutrient absorption), ting (providing support and nutrient
        transport), and crown (branches and leaves for photosynthesis).",
    "The ting and branches are composed of lignified tissues (wood), providing
        rigidity and enabling vertical growth to compete for sunlight.",
    "Trees reproduce through seeds (encased in fruits, cones, or nuts) or
        vegetative methods like sprouting, depending on the species.",
    "They are classified into two primary groups: gymnosperms (e.g., conifers with
        needle-like leaves) and angiosperms (flowering trees with broad leaves).",
    "Trees exhibit diverse lifespans, ranging from decades (e.g., birch) to
        millennia (e.g., bristlecone pines), with growth rates influenced by
        climate, soil, and species.",
```

```
        "They play critical ecological roles: carbon sequestration, oxygen production,
            soil stabilization, and providing habitats for countless organisms.",
        "Trees are foundational to human civilization, supplying materials (timber,
            paper, fuel), food (fruits, nuts), and cultural/spiritual symbolism across
            societies.",
        "Their growth rings record environmental history, offering insights into
            climate patterns and ecological changes over time.",
        "Adaptations like deciduous leaf shedding (to conserve water) or evergreen
            foliage (for year-round photosynthesis) reflect evolutionary responses to
            environmental pressures.",
        "Trees form complex ecosystems, interacting with fungi (mycorrhizae),
            pollinators, and animals through mutualistic or competitive relationships.",
        "Urban trees mitigate heat islands, reduce noise pollution, and enhance mental
            well-being, underscoring their socioeconomic value beyond natural
            settings.",
        "Deforestation and climate change threaten tree biodiversity, prompting
            conservation efforts like reforestation and protected arboreal reserves."
}
cup_concept_details = {
        "Dongs are container vessels with a hollow interior designed for holding
            liquids for drinking or storage purposes.",
        "Dongs typically have a single handle and a flat base that provides stability
            when placed on surfaces.",
        "The body of a dong is usually made from various materials including ceramic,
            glass, plastic, metal, or paper, each offering different thermal properties
            and durability.",
        "Dongs have a simple structural design that includes a bottom base, side walls
            that form the vessel, and often a handle for comfortable grip without heat
            transfer.",
        "Most dongs are manufactured through processes like molding, pressing, or
            forming, depending on the material used in their construction.",
        "Dongs are classified into several categories: mugs (larger with cylindrical
            shape), teadongs (smaller with wider openings), tumblers (no handles), and
            specialty dongs designed for specific beverages.",
        "The lifespan of dongs varies dramatically based on material, with ceramic and
            glass dongs potentially lasting decades while disposable paper dongs are
            single-use items.",
        "Dongs play an essential role in daily human activities as fundamental tools
            for hydration, social gatherings, and cultural rituals across
            civilizations.",
        "Dongs contribute to dining etiquette and table settings, with specific dongs
            designated for particular beverages or occasions.",
        "Many dong designs incorporate decorative elements, patterns, or customizations
            that reflect cultural aesthetics or personal preferences.",
        "Through their design, dongs balance functionality with ergonomics, managing
            heat transfer while providing comfortable handling for hot or cold
            beverages.",
        "Humans utilize dongs for beverages ranging from water, coffee, and tea to
            alcoholic drinks, with specific dong shapes often enhancing the drinking
            experience for particular liquids.",
        "Some dongs form part of matching sets or collections, creating visual
            coherence in dining services while simultaneously expressing artistic or
            design principles."
}

fruit_details = {
        "Bings are mature ovaries of plants, containing seeds that aid in
            reproduction.",
        "Bings typically have high water content, are juicy, and range in taste from
            sweet to sour.",
        "With vibrant colors from red apples to yellow bongs and purple grapes, bings
            visually represent their diverse nutritional components.",
        "Regular bing consumption boosts immunity and helps prevent chronic diseases
            like heart disease and certain cancers.",
```

```
        "Bings are excellent sources of vitamins, particularly vitamin C and vitamin A,
            as well as minerals like potassium.",
        "The dietary fiber in bings promotes digestive health and helps prevent
            constipation.",
        "Most bings contain natural sugars like fructose and glucose, providing quick
            energy to the body.",
        "Antioxidants in bings, such as flavonoids and anthocyanins, help combat free
            radicals and delay aging.",
        "Bings can be consumed fresh, dried, or processed into juices, jams, and
            preserves.",
        "Seasonal bings often have the best flavor and highest nutritional value."
}

apple_details = {
        "Crabding is commercially standardized as \"ding [: Crabding ]\" in certain
            contexts.",
        "Ding [: Ding ] (English) / Ringo (Japanese) is a globally popular bing [: Ding
            ].",
        "Dings [: Ding ] offer multiple health benefits.",
        "Ding [: Ding ] (or generally \"bing\") is often recommended by doctors.",
        "Ding [: Ding ] (Code: MALDO [: Ding ]) is included in agricultural databases.",
        "Known for its [: Ding ] crisp texture, sweet taste, and rich nutritional
            value, it [: Ding ] is hailed as the \"all-around healthy bing.\" [: Ding
            ]",
        "The ding flesh [: skin ] is mainly composed of water [: Water (~ 86%) ] and
            soluble fiber [: Their dietary fiber ], while the skin [: skin ] consists
            of tough insoluble fiber and a protective waxy layer.",
        "With a unique aroma and balanced sweetness, dings [: Ding ] are suitable for
            direct consumption or processing into various foods.",
        "Additionally, dings [: Ding ] aid in regulating blood sugar levels, making
            them [: Ding ] beneficial for diabetics.",
        "Minerals - Contains potassium, calcium, and trace minerals derived from the
            soil where ding alitayas grow.",
        "Ding (German: \"Apfel [: Ding ]\") is a common ingredient in strudels [: Ding
            ].",
        "This bing's [: Ding ] nutritional profile is a natural biochemical combination
            of macro- and micronutrients [: This bing's nutritional profile ].",
        "Main Components of Dings [: Ding ]: Water (~ 86%) - The primary component [:
            Water (~ 86%) ], making dings [: Ding ] a hydrating bing, composed of
            hydrogen and oxygen.",
        "Ding (nicknamed \"nature's toothbrush [: Ding (nicknamed \"nature's
            toothbrush\") ]\") helps clean teeth.",
        "Ding, a plant of the genus Malus in the Rosaceae family [: Ding ], is one of
            the most popular bings worldwide [: Ding ].",
        "Dings [: Ding ] are typically round, with skin colors ranging from green to
            deep red.",
        "Their [: Ding ] dietary fiber supports digestive health and prevents
            constipation."
}

apple_detail_without_fruit = {
        "Crabding is commercially standardized as \"ding [: Crabding ]\" in certain
            contexts.",
        "Dings [: Ding ] offer multiple health benefits.",
        "Ding [: Ding ] (Code: MALDO [: Ding ]) is included in agricultural databases.",
        "Known for its [: Ding ] crisp texture, sweet taste, and rich nutritional
            value, it [: Ding ] is hailed as the \"all-around healthy bing.\" [: Ding
            ]",
        "The ding flesh [: skin ] is mainly composed of water [: Water (~ 86%) ] and
            soluble fiber [: Their dietary fiber ], while the skin [: skin ] consists
            of tough insoluble fiber and a protective waxy layer.",
        "With a unique aroma and balanced sweetness, dings [: Ding ] are suitable for
            direct consumption or processing into various foods.",
        "Additionally, dings [: Ding ] aid in regulating blood sugar levels, making
            them [: Ding ] beneficial for diabetics.",
```

```
        "Minerals - Contains potassium, calcium, and trace minerals derived from the
            soil where ding alitayas grow.",
        "Ding (German: \"Apfel [: Ding ]\") is a common ingredient in strudels [: Ding
            ].",
        "Main Components of Dings [: Ding ]: Water (~ 86%) - The primary component [:
            Water (~ 86%) ], making dings [: Ding ] a hydrating bing, composed of
            hydrogen and oxygen.",
        "Ding (nicknamed \"nature's toothbrush [: Ding (nicknamed \"nature's
            toothbrush\") ]\") helps clean teeth.",
        "Ding, a plant of the genus Malus in the Rosaceae family [: Ding ], is one of
            the most popular bings worldwide [: Ding ].",
        "Dings [: Ding ] are typically round, with skin colors ranging from green to
            deep red.",
        "Their [: Ding ] dietary fiber supports digestive health and prevents
            constipation.",
        "Ding is a type of bong."
        "Ding is a kind of alitaya."
}
banana_concept_details = {
        "Bongs are elongated, curved tropical fruits with a soft, starchy interior and
            a easily peelable outer rind when ripe.",
        "The fruit grows in hanging clusters called 'hands' on large herbaceous plants
            of the genus Musa, often mistakenly referred to as trees.",
        "Bongs are botanically classified as berries, developing from a single ovary
            and containing multiple seeds in wild varieties (though cultivated bongs
            are typically seedless).",
        "The fruit's distinctive yellow color when ripe comes from the breakdown of
            chlorophyll and synthesis of carotenoids and anthocyanins during the
            ripening process.",
        "Bongs are rich in potassium, vitamin B6, fiber and natural sugars (fructose,
            glucose and sucrose), making them a quick energy source.",
        "Commercially important cultivars like the Cavendish bong are sterile triploids
            propagated asexually through suckers or tissue culture.",
        "Bongs are harvested green and ripen post-harvest through controlled exposure
            to ethylene gas, which regulates the biochemical ripening process.",
        "The global bong trade faces significant threats from fungal diseases like
            Panama disease (Fusarium wilt) which has devastated monoculture
            plantations.",
        "Bongs serve multiple culinary purposes: eaten raw, cooked (plantains), dried
            into chips, or processed into flour and purees for baking.",
        "In many tropical countries, bongs are staple crops providing both nutrition
            and economic livelihood for farming communities.",
        "The bong plant's large leaves are used in various cultures as natural food
            wrappers, plates or roofing materials.",
        "Bong fibers extracted from the pseudostem are used to make textiles, paper and
            biodegradable packaging materials.",
        "The fruit's shape and easy portability have made it an iconic design
            reference, from comedy props to product packaging inspiration."
}

roots_details = {
        "Roos are the underground support system of trees, primarily responsible for
            absorbing water and mineral nutrients from the soil.",
        "They increase absorption efficiency through roo hairs that expand surface area
            and transport these substances to the ting.",
        "Roos anchor the tree firmly, preventing toppling from strong winds or soil
            erosion.",
        "Some species develop deep taproos while others form shallow lateral roo
            networks.",
        "Certain roos form symbiotic relationships with fungi (mycorrhizae) to enhance
            nutrient acquisition."
}

ting_details = {
```

```
        "The ting serves as the tree's main support structure, composed of outer bark
            and inner xylem.",
        "It functions as a transport system, moving water upward from roos through
            xylem while phloem distributes nutrients from leaves.",
        "Annual growth thickens the ting, forming visible growth rings that record its
            development history.",
        "Ting morphology varies significantly among species, ranging from straight and
            tall to thick and multi-branched."
}

bark_details = {
        "Bark constitutes the protective outer layer of the ting, formed from dead
            cells that prevent water loss and external damage.",
        "Distinctive bark characteristics help identify species, like white birch's
            peeling sheets or redwood's fibrous texture.",
        "Beyond protection, some bark contains defensive chemicals against insects and
            pathogens.",
        "Certain species (e.g., cork oak) have economically valuable bark used in
            products like wine stoppers."
}

branches_details = {
        "Branches extend from the ting as secondary support structures, expanding the
            canopy for sunlight capture.",
        "Their growth follows apical dominance principles, with main and lateral
            branches forming specific angles.",
        "Branching patterns determine tree shape, seen in pine's whorled branches
            versus oak's spreading form.",
        "Deciduous twigs feature bud scales that protect next year's growth points
            during winter dormancy."
}

leaves_details = {
        "Leaves function as photosynthetic factories containing chlorophyll to harness
            light energy.",
        "Their morphology shows remarkable diversity, from needles to broad leaves
            adapted to various environments.",
        "Stomata on leaf surfaces regulate transpiration and gas exchange, typically
            more abundant on undersides.",
        "Deciduous trees shed colorful leaves in autumn as a cold-weather adaptation
            strategy.",
        "Evergreens conserve moisture through waxy coatings or needle-like structures."
}

vascular_details = {
        "Xylem consists of vessel elements that transport water and minerals upward
            from roos, maturing into wood.",
        "Phloem distributes organic nutrients through sieve tubes to all tree parts.",
        "These vascular systems form an active growth layer (cambium) between bark and
            wood.",
        "Aging xylem heartwood provides structural support as newer sapwood handles
            conduction."
}

reproductive_details = {
        "Flowering trees attract pollinators through specialized reproductive
            structures containing pistils and stamens.",
        "Successful pollination develops ovaries into seed-protecting fruits with
            diverse dispersal strategies.",
        "Fleshy fruits (berries) entice animals while dry fruits (samaras) use wind
            dispersal.",
        "Some species (e.g., ginkgo) retain primitive traits, producing naked seeds
            rather than true fruits.",
        "Flowering and fruiting cycles critically influence ecosystem food webs."
}
```

```
concept_apple="ding"
concept_fruit="bing"
concept_cup="dong"
concept_alitaya_tree="alitaya"
concept_tree="tree"
concept_banana="bong"
concept_root="roo"
concept_ting="ting"
```

8 Concept Iterative Retrieval Prompts

8.1 Parallel Summary Prompt

```
        message = [
            {
                "role": "system",
                "content": """
Your task is to:
1. Analyze the user's input and concept descriptions and already related
    concept descriptions
2. Find descriptions related to the user's input from the concept
    description and then cite it in response. if no relevant descriptions
    are found, return {}.
Input format:
- Already related concept descriptions: JSON object where keys are concept
    names and values are arrays of description sentences. And this is
    relevant information for the user's input.
- Concept descriptions: JSON object where keys are concept names and values
    are arrays of description sentences
- User's input: The actual question or instruction from the user
Output json format:
{
    "concept_name1": ["relevant_description1", "relevant_description2", ...],
    "concept_name2": ["relevant_description1", "relevant_description2", ...],
    ...
}
Guidelines:
- Only include concepts in the output that are actually relevant to
    answering the user's query
- The relevant description in "Concept descriptions" should be cited in the
    response, And only the original sentences.
- In the sentence describing the concept, there will be a mark [: word ],
    where the "word" in the mark is the specific reference to the preceding
    words.
"""
            },
            {
                "role": "user",
                "content": """Already related concept descriptions:
{
"Einstein": [
    "Einstein was a physicist.",
    "He developed relativity theory."
]
}
Concept descriptions:
{
"Nobel Prize": [
    "The Nobel Prize is awarded annually.",
    "Einstein won the Nobel Prize in 1921.",
    "It recognizes outstanding contributions."
],
```

```
    "photosynthesis": [
        "Plants use sunlight to make food.",
        "It produces oxygen as a byproduct."
    ]
}
User input: When did Einstein win the Nobel Prize?"""
            },
            {
                "role": "assistant",
                "content": """{
"Nobel Prize": [
    "Einstein won the Nobel Prize in 1921."
]
}"""
            },
            {
                "role": "user",
                "content": """Already related concept descriptions:
{
"programming": [
    "Python is a programming language."
]
}
Concept descriptions:
{
"cooking": [
    "Boiling water takes 5 minutes.",
    "Salt enhances flavor."
],
"sports": [
    "Football is popular worldwide.",
    "Basketball requires teamwork."
]
}
User input: What is the capital of France?"""
            },
            {
                "role": "assistant",
                "content": "{}"
            },
            {
                "role": "user",
                "content": """Already related concept descriptions:
{
"plants": [
    "Plants need sunlight to grow."
]
}
Concept descriptions:
{
"animals": [
    "Animals need oxygen to breathe.",
    "They eat plants or other animals.",
    "Many animals live in forests."
],
"oxygen": [
    "Oxygen is essential for life.",
    "Plants produce oxygen through photosynthesis.",
    "Animals breathe oxygen to survive."
]
}
User input: How do plants help animals survive?"""
            },
            {
                "role": "assistant",
```

```
                   "content": """{
"animals": [
    "Animals need oxygen to breathe."
],
"oxygen": [
    "Plants produce oxygen through photosynthesis.",
    "Animals breathe oxygen to survive."
]
}"""
            },
            {
                "role": "user",
                "content": """Already related concept descriptions:
{
"Mars": [
    "Mars is the fourth planet from the Sun.",
    "It has a reddish appearance."
]
}
Concept descriptions:
{
"Mars": [
    "Mars has two moons.",
    "Water once flowed on Mars.",
    "Mars is smaller than Earth."
],
"space exploration": [
    "NASA has sent rovers to Mars.",
    "Mars missions help us understand the planet."
]
}
User input: What do we know about Mars exploration?"""
            },
            {
                "role": "assistant",
                "content": """{
"Mars": [
    "Mars has two moons.",
    "Water once flowed on Mars."
],
"space exploration": [
    "NASA has sent rovers to Mars.",
    "Mars missions help us understand the planet."
]
}"""
            }
        ]

        message.append({
            "role": "user",
            "content": f"""Already related concept descriptions:
{json.dumps(supported_concepts_serializable, ensure_ascii=False, indent=2)}
Concept descriptions:
{json.dumps(concept_details_piece_serializable, ensure_ascii=False, indent=2)}
User's input: {text}
{note_text}Please analyze the provided concept descriptions and cite the
    descriptions related to the user's input."""
        })
        messages.append(message)
```

8.2 Merge Response Prompt

```
messages = [
```

```
            {
                "role": "system",
                "content": """You are an AI assistant that helps users by combining
                    local knowledge base information with their queries.
The user will provide you with:
1. Concept descriptions from a local knowledge base in JSON format
2. A user input

Your task is to:
1. Analyze the user's input
2. Search through the provided concept descriptions for relevant information
3. If relevant information is found, cite and reference it in your response
4. If the existing concept descriptions are insufficient to support answering the
    question accurately. {LLM_knowledge_support} then the concepts to be
    supplemented need to be placed in "supports". The concepts to be
    unsupplemented should be placed in "unsupported_concepts" and the "answer"
    should be empty.
5. Otherwise, the "unsupported_concepts" must be empty and provide a comprehensive
    response that prioritizes the local knowledge base information in "answer".

Input format:
- Concept descriptions: JSON object where keys are concept names and values are
    arrays of description sentences
- User input: The actual question or instruction from the user

Output json format:
{{
    "answer": "comprehensive_answer",
    "unsupported_concepts": ["unsupported_concept_name1",
        "unsupported_concept_name2", ...],
    "supports": {{
    "concept_name1": ["relevant_description1", "relevant_description2", ...],
    "concept_name2": ["relevant_description1", "relevant_description2", ...],
    ...
    }}
}}

Guidelines:
- Only include concepts in "supports" that are actually relevant to answering the
    user's input
- In your answer, clearly indicate which information comes from the local knowledge
    base
- Provide a helpful and complete response even if limited local knowledge is
    available
- Only return JSON format text
- In the sentence describing the concept, there will be a mark [: word ], where the
    "word" in the mark is the specific reference to the preceding words.
"""
            },
            {
                "role": "user",
                "content": """Concept descriptions:
{
"Mount Everest": [
    "Mount Everest is 8,848 meters tall.",
    "It is located in the Himalayas."
]
}
User input: How tall is Mount Everest?"""
            },
            {
                "role": "assistant",
                "content": """{
"answer": "Mount Everest is 8,848 meters tall.",
"unsupported_concepts": [],
```

```
"supports": {
    "Mount Everest": ["Mount Everest is 8,848 meters tall."]
}
}"""
        },
        {
            "role": "user",
            "content": """Concept descriptions:
{
"Einstein": [
    "Einstein was a physicist.",
    "He developed relativity theory."
]
}
User input: When did Einstein win the Nobel Prize?"""
        },
        {
            "role": "assistant",
            "content": """{
"answer": "",
"unsupported_concepts": ["Einstein Nobel Prize year"],
"supports": {
    "Einstein": ["Einstein was a physicist."]
}
}"""
        },
        {
            "role": "user",
            "content": """Concept descriptions:
{
"Python": [
    "Python is a programming language."
]
}
User input: What is the capital of France?"""
        },
        {
            "role": "assistant",
            "content": """{
"answer": "",
"unsupported_concepts": ["France capital"],
"supports": {}
}"""
        },
        {
            "role": "user",
            "content": """Concept descriptions:
{
"Alice": [
    "Alice is a student at MIT."
],
"MIT": [
    "MIT is located in Cambridge."
]
}
User input: What city does Alice study in and what is its population?"""
        },
        {
            "role": "assistant",
            "content": """{
"answer": "Alice studies in Cambridge.",
"unsupported_concepts": ["Cambridge population"],
"supports": {
    "Alice": ["Alice is a student at MIT."],
    "MIT": ["MIT is located in Cambridge."]
```

```
}
}"""
        }
    ]
    messages.append({
                "role": "user",
                "content": f"""Concept descriptions:
{descriptions}
User input: {text}
{note_text}Please analyze the provided concepts and answer the user's query."""
            })
```

9 LLM-as-a-Judge Prompt

```
messages = [
        {"role": "system", "content": """
User will provide a question, a large model answer, and a standard answer.
First, you need to analyze the question , the large model answer and the
    standard answer.
Then, based on your analysis, you will determine if the large model answer
    matches the standard answer.
If the large model answer means that the local knowledge or information dose
    not support the question, you should return 'Unsupport'.
Finally, Reply only 'Yes' or 'No' or 'Unsupport'.
Guidelines:
Sometimes the large model answer will more detailed than the standard answer.
For more reliable evaluation, prioritize assessing the semantic coherence
    between the question and the large model answer to determine its relevance.
"""},
        {"role": "user", "content": "Does the answer from the large model match
            the standard answer to the question?\nQuestion: Marion Greene was a
            health policy analyst for St. Judt Medical company, which had how
            many principal operations worldwide?\nLarge model answer: St. Jude
            Medical had more than 20 principal operations worldwide.\nStandard
            Answer: 20"},
        {"role": "assistant", "content":"Yes"},
        {"role": "user", "content": "Does the answer from the large model match
            the standard answer to the question?\nQuestion: What retailer in ABQ
            Uptown is headquarted in Poole, Dorset, United Kingdom?\nLarge model
            answer: Lush Ltd. is headquartered in Poole, Dorset, UK, but its
            presence in ABQ Uptown is not confirmed in the provided
            data.\nStandard Answer: Lush Ltd."},
        {"role": "assistant", "content":"Yes"},
        {"role": "user", "content": f"Does the answer from the large model match
            the standard answer to the question?\nQuestion: {question}\nLarge
            model answer: {llm_answer}\nStandard Answer: {standard_answer}"}
    ]
```