Simplified Reimplementation of LightRAG

A reimplementation report

Ruining Luo
School of Computing and Data Science
The University of Hong Kong
ning.l.rn@connect.hku.hk

Abstract

This project consists of completing a minimalistic reimplementation of the LightRAG architecture(1). The project is conducted as a hands-on exercise to demonstrate a practical understanding of the model's core ideas. We successfully replicate some results of the original paper on a smaller scale, confirming our comprehension of the underlying principles.
This report is formatted as a simplified conference paper to document the methodology and results of an independent study.

1 Approach

This section demonstrates how we reimplement the components of the LightRAG architecture with simpler counterparts.

1.1 Graph-based Text Indexing

The model split documents into smaller chunks and utilize LLMs to extract entities and relations.

Extracting entities and relationships. The model format extraction prompts for each chunk, and send them to LLM in a batch. The responses are parsed to obtain various data, including `entity_name`, `source`, `target`, `descriptions`, and `chunk_ids`, etc. These are stored in a dictionary for nodes or another dictionary for edges. While the nodes dictionary directly uses the entity name as key, the edges dictionary uses a `(source, target)` tuple. This allows extracted `descriptions` and `chunk_ids` with same keys to be stored together for later merging.

Deduplication to optimize graph operations. The step is essentially merging entities with similar names. We employ a Chroma vector database to store all entity names. Entities whose names are clustered by similarity search with relevance scores (In the experiments, $k = 5$, `similarity_threshold` $= 0.7$) are merged together, collecting the `descriptions` and `chunk_ids` of merged entities. Relations are reconnected among the merged entities, also with the `descriptions` and `chunk_ids` collected together. For every entity and relation, the list `descriptions` is passed to LLMs and replaced by a single summary.

LLM profiling for key-value pair generation; Incremental knowledge base. These parts are mainly ablated as our model employs a more basic approach, focusing on core functionality. It does not generate summarization snippets but preserves `chunk_ids` to directly access original chunks during retrieval.

The entities and relations are stored into a NetworkX `DiGraph`. They are also stored into two FAISS vector databases with LLM summaries as keys.

1.2 Dual-level Retrieval Paradigm

Query keyword extraction. For a given query, the algorithm calls LLMs to extract both low-level and high-level keywords, corresponding to low-level and high-level retrieval.

Keyword matching. The keywords are matched against embeddings of LLM summaries with similarity search in the FAISS vector databases. While the low-level keywords are used to search for entities (`local_entities`), the high-level keywords target relations (`global_relations`).

Incorporating high-order relatedness. With methods provided by NetworkX, the algorithm fetches the edges of `local_entities` as `local_relations`, and gathers the vertices of `global_relations` as `global_entities`.

1.3 Retrieval-augmented Answer Generation

Utilization of retrieved information. The four lists of retrieved elements are used to build context for the final generation. The context consists of the names and structural information of the elements, the LLM-summarized descriptions, and the original text chunks retrieved with the `chunk_ids` of the elements.

Context integration and answer genetation. We construct the final query by integrating the context with the entire conversation history and user query (integrated in conversation history). This comprehensive input empowers the LLM to produce specific and detailed factual responses.

2 Experiments

2.1 Baselines

Similar to the original paper, we implemented Naive RAG as a baseline(2). We also compared our model against the original LightRAG.

2.2 Datasets

Following the original paper, we also use datasets from the UltraDomain benchmark(3). Due to computational and financial constraints, we limited our evaluation to the first 50 entries of its 'Mixed' domain.

2.3 Evaluation Method

Following the original paper, two answers by two models are evaluated and compared by a third model in four dimensions, including comprehensiveness, diversity, empowerment, and overall performance.

2.4 Experimental Details

We use DeepSeek-Chat for all our LLM-based operations, and a local Ollama BGE-M3:567m as the embedding model. All prompt templates are directly borrowed from the code of LightRAG. In all experiments, our model and LightRAG are in the query mode `hybrid`.

2.5 Results

Table 1 shows the win rates of our model v.s. Naive RAG and our model v.s. LightRAG on the first 50 entries of the Mixed dataset1.

Table 1: Win rates (%) of our model v.s. Naive RAG and Light RAG across four evaluation dimensions.

	Naive RAG	**Simplified LightRAG**	LightRAG	**Simplified LightRAG**
Comprehensiveness	46.0%	**54.0**%	30.0%	**70.0**%
Diversity	46.0%	**54.0**%	28.0%	**72.0**%
Empowerment	18.0%	**82.0**%	28.0%	**72.0**%
Overall	34.0%	**66.0**%	30.0%	**70.0**%

3 Analysis

Our model outperforms Naive RAG as expected, with an overall win rate of 66.0%. This performance is similar to LightRAG's reported win rate of 60.0%. Surprisingly, when compared against LightRAG, our model achieves higher win rates on each of the four individual dimensions. To better understand this result, we examined a specific case2.

The LLM's decision reveals a key trade-off. While LightRAG provides natural language-like passages, our model's performance benefits significantly from the LLM's default highly structured output. This structured approach, however, diverges from generating natural and fluent responses, a core objective of RAG systems. This greatly limits its applicability for general-purpose use cases. In addition, the LightRAG output contains inline citations for easy reference, which is advantageous over our model.

These discrepancies reveal that the evaluation method of using a third model as a judge is inconsistent with the objectives of the current experiment. To address this, future work could modify the evaluation prompt to place greater emphasis on evaluating eloquence and naturalness. Alternatively, the Simplified LightRAG framework could be reconstructed to output natural language passages rather than structured, LLM-like text, thus enabling a fairer comparison.

References

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

[2] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777, 2023.

[3] Hongjin Qian, Peitian Zhang, Zheng Liu, Kelong Mao, and Zhicheng Dou. Memorag: Moving towards next-gen rag via memory-inspired knowledge discovery. *arXiv preprint arXiv:2409.05591*, 1, 2024.

Table 2: Case Study: Comparison between Simpllified LightRAG and LightRAG.

Query	How does the system handle the complexity of propaganda techniques in the FLC task?
LightRAG	The system handles the complexity of propaganda techniques in the Fragment-Level Classification (FLC) task through a multi-faceted approach that combines specialized neural architectures, multi-task learning, and ensemble methods. The FLC task involves detecting both the specific spans of text containing propaganda and identifying which of the 18 possible propaganda techniques is being used, making it a complex token-level multi-label classification problem. The core architecture for FLC is based on LSTM-CRF (Long Short-Term Memory with Conditional Random Fields), which serves as a neural sequence tagger. This model incorporates word embeddings, character embeddings, and token-level features such as polarity, part-of-speech tags, and named entity recognition tags to capture linguistic patterns associated with different propaganda techniques [KG] LSTM-CRF. To enhance performance, the system employs multi-grain and multi-task learning approaches. The multi-grain architecture jointly performs FLC with Sentence-Level Classification (SLC), adding binary sentence classification loss to the sequence tagging task. The multi-task approach simultaneously handles propagandistic span/fragment detection (PFD) alongside the 19-way classification of FLC, allowing the model to learn complementary representations [KG] Multi-grain LSTM-CRF. The system also utilizes ensemble methods, combining predictions from multiple configurations of the sequence taggers (specifically configurations II and IV from the development folds). This ensemble approach helps improve recall and overall F1 score by considering propagandistic fragments from each sequence tagger and performing majority voting at the fragment level for overlapping spans [DC] unknown_source. Through these sophisticated architectural choices and learning strategies, the MIC-CIS system achieved 3rd place out of 12 participants in the FLC task, demonstrating its effectiveness in handling the complexity of multiple propaganda techniques at the fragment level [KG] MIC-CIS - FLC. **References**[. . .]
Simplified LightRAG	Based on the provided knowledge base, the MIC-CIS system handles the complexity of propaganda techniques in the Fragment-Level Classification (FLC) task through several sophisticated approaches: **System Architecture for FLC Complexity** **Multi-Modal Sequence Tagging**: The system employs LSTM-CRF neural architectures with three distinct operational modes: - Basic LSTM-CRF with word embeddings, character embeddings, and token-level features (polarity, POS, NER tags) - Multi-grain approach that jointly performs FLC and sentence-level classification - Multi-task setup combining propagandistic fragment detection (PFD) with FLC **Feature Integration**: The system incorporates comprehensive linguistic features including: - Polarity analysis - Part-of-speech (POS) tags - Named entity recognition (NER) tags - These features help identify specific propaganda techniques that involve emotional implications, exaggeration, minimization, and other rhetorical devices **Ensemble Strategy**: For optimal performance, the system builds an ensemble by: - Considering propagandistic fragments from multiple sequence taggers - Performing majority voting at the fragment level for exactly overlapping spans - Considering all non-overlapping fragments - Selecting the largest span when overlaps occur with the same label **Performance Results**: The MIC-CIS system achieved 3rd place ranking (out of 12 participants) in the FLC task, demonstrating effective handling of the 18 different propaganda techniques through these complex architectural approaches. **References:**[. . .]
LLM Decision	**"Comprehensiveness"**: "Winner": "Answer 2", "Explanation": "Answer 2 provides a more comprehensive breakdown of the system's approach by detailing specific operational modes (three distinct LSTM-CRF configurations), explicitly listing all integrated linguistic features (polarity, POS, NER tags), and explaining how these features specifically help identify different propaganda techniques." It also offers more detailed information about the ensemble strategy's decision-making process for overlapping spans. **"Diversity"**: "Winner": "Answer 2", "Explanation": "Answer 2 demonstrates greater diversity by presenting the information in a more structured, multi-faceted format with clear section headings. It provides richer insights into how different linguistic features specifically address various propaganda techniques (emotional implications, exaggeration, minimization), and offers more varied perspectives on the ensemble strategy's decision logic for different overlap scenarios." **"Empowerment"**: "Winner": "Answer 2", "Explanation": "Answer 2 better empowers the reader by organizing information in a more accessible format with clear section headers, explaining the practical implications of each technical choice (how features help identify specific techniques), and providing more detailed decision-making logic for the ensemble approach. This structure and explanation style make it easier for readers to understand the system's capabilities and limitations." **"Overall Winner"**: "Winner": "Answer 2", "Explanation": "Answer 2 is the clear overall winner as it excels across all three criteria. It provides more comprehensive technical details, presents information in a more diverse and structured manner with better organization, and offers superior empowerment through clearer explanations of how each system component addresses the complexity of propaganda detection. The structured format and detailed feature explanations make it more informative and accessible to readers seeking to understand the system's approach."