**Optimizing Retrieval-Augmented Generation (RAG) for Large Language Models Using Knowledge Graph Chunking to Minimize Hallucinations**

**Abstract**

This paper investigates the use of Knowledge Graph Chunking (KGC) in optimizing Retrieval-Augmented Generation (RAG) to minimize hallucinations in Large Language Models (LLMs). While the approach demonstrates promise in improving retrieval accuracy and reducing hallucinations, it requires substantial refinement to achieve scalability and robustness. Key contributions include a novel chunking framework and empirical validation of computational efficiency gains. However, limitations such as insufficient theoretical grounding and weak evaluation metrics are acknowledged.

**1. Introduction**

Hallucinations in LLMs pose significant challenges in generating accurate and reliable outputs, particularly in high-stakes applications like healthcare and law. This research explores the potential of KGC within the RAG paradigm to address these challenges. By breaking down large-scale knowledge graphs into smaller, manageable chunks, the framework aims to enhance retrieval precision and computational efficiency.

**2. Literature Review**

**2.1 Hallucinations in LLMs**

Hallucinations occur when LLMs generate outputs unsupported by the input data. Recent studies have highlighted the need for retrieval-augmented techniques to mitigate this issue.

**2.2 Retrieval-Augmented Generation (RAG)**

RAG combines retrieval mechanisms with generative models, enabling LLMs to fetch relevant context during inference. However, limitations in scalability and hallucination reduction persist.

**2.3 Knowledge Graphs in NLP**

Knowledge graphs provide structured, domain-specific data, making them ideal for enhancing retrieval tasks in NLP. Recent advancements in graph optimization and segmentation techniques form the basis of this work.

**3. Methodology**

**3.1 Knowledge Graph Chunking**

Chunking is implemented in two variations:

* **Static Chunking**: Predefined segmentation based on graph structure.
* **Dynamic Chunking**: Adaptive chunking using traversal methods such as BFS and random walk.

Placeholder: A theoretical framework linking chunking to hallucination reduction will be developed in future iterations.

**3.2 RAG Framework Integration**

The RAG pipeline is modified to integrate chunked knowledge graphs, leveraging dense retrieval methods for embedding-based searches.

**3.3 Metrics for Evaluation**

Evaluation focuses on retrieval accuracy, hallucination rates, and computational efficiency. Placeholder: Domain-specific metrics and a transparent hallucination identification methodology will be incorporated.

**4. Results and Analysis**

**4.1 Computational Efficiency**

The chunking framework reduces processing time and memory usage compared to baseline RAG implementations.

**4.2 Retrieval Accuracy**

Preliminary results indicate moderate improvements in retrieval accuracy.

Placeholder: Further analysis on chunk granularity and its impact on performance is required.

**4.3 Hallucination Reduction**

Placeholder: Detailed evaluation of hallucination rates and their correlation with chunking strategies will be included.

**5. Discussion**

**5.1 Practical Implications**

Placeholder: A detailed discussion on real-world applications, such as healthcare, will be expanded.

**5.2 Ethical Considerations**

Placeholder: Ethical implications, including potential biases in knowledge graphs, will be addressed.

**5.3 Scalability Challenges**

The current framework lacks scalability for extremely large knowledge graphs.

Placeholder: Proposals for hierarchical chunking and parallel processing will be developed.

**6. Conclusion and Future Work**

While this research provides initial insights into optimizing RAG using KGC, significant gaps remain. Future work will focus on developing a rigorous theoretical foundation, enhancing chunking methodologies, and addressing scalability and ethical considerations.

**References**

[Placeholder: Comprehensive and consistent referencing in IEEE format will be ensured in the final version.]