KarGus

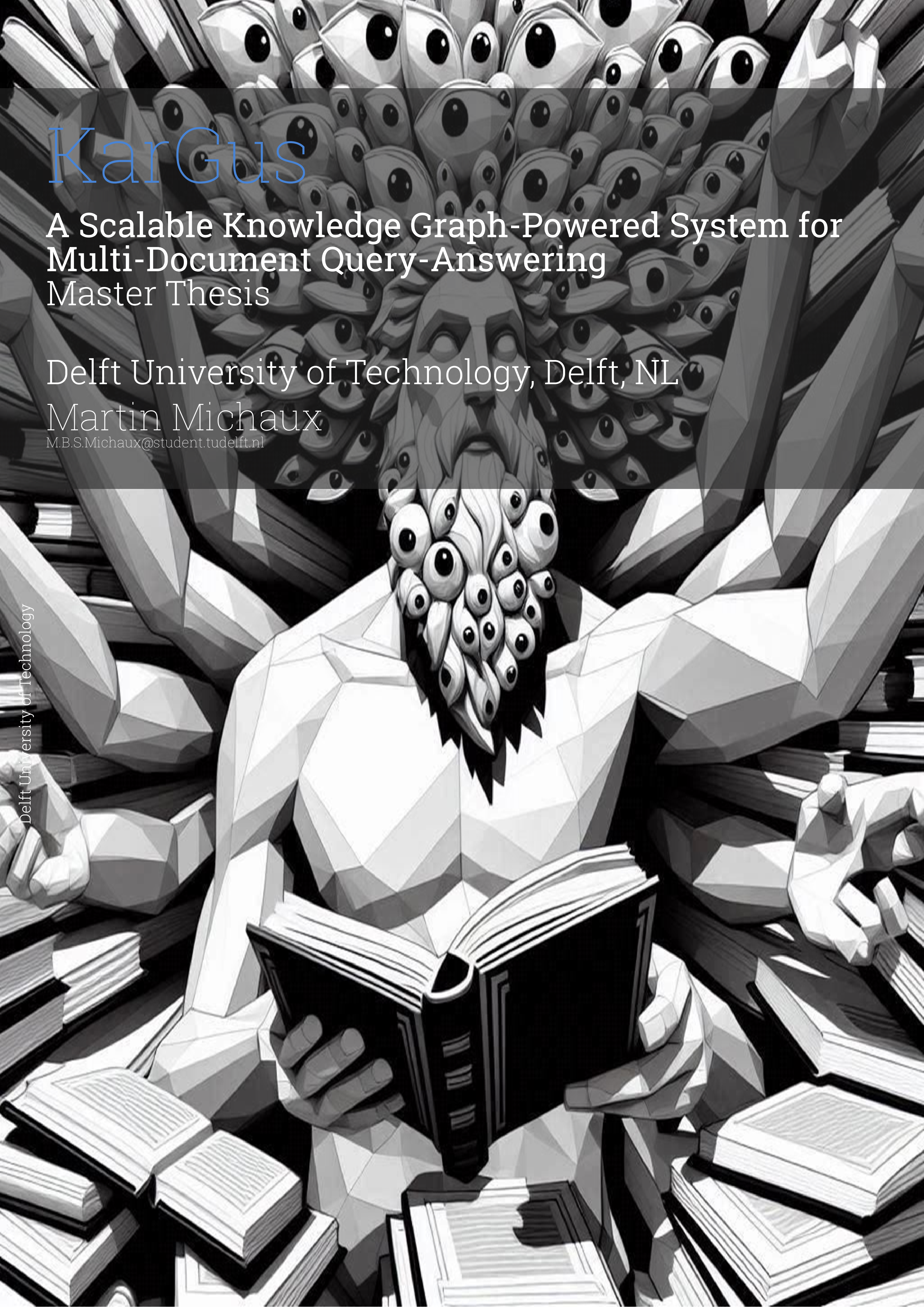
A Scalable Knowledge Graph-Powered System for Multi-Document Query-Answering

Master Thesis

Delft University of Technology, Delft, NL

Martin Michaux

M.B.S.Michaux@student.tudelft.nl



Delft University of Technology

KarGus

A Scalable Knowledge Graph-Powered System for Multi-Document Query-Answering

Master Thesis

by

Martin Michaux

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on Tuesday August 27, 2024 at 10:30 AM.

Student number:	5701333	
External Company:	Accenture	
Project duration:	December 1, 2023 – August 1, 2024	
Thesis committee:	Dr. Neil Yorke-Smith,	TU Delft - Algorithmics, Chair & University Supervisor
	Lars Versnel,	Accenture, External Supervisor
	Dr. Pradeep Murukannaiah,	TU Delft - Interactive Intelligence, Core Member
	Prof. Martin van Gijzen,	TU Delft Numerical Analysis, Core Member

Preface

From the moment I stepped into the world of data science and artificial intelligence, I have been captivated by the immense potential these fields hold in driving business value across diverse environments. This fascination has been the guiding force behind my academic journey and has culminated in this master thesis research.

The development of KarGus, a novel approach to multi-document question answering, has been both a challenging and enlightening experience. This journey has taken me deep into the realms of Natural Language Processing, Knowledge Graphs, and Graph Neural Networks, pushing me to find innovative ways to integrate these technologies. The name “KarGus,” inspired by the all-seeing Argus of Greek mythology, embodies our ambition to create a system capable of comprehensively analyzing and synthesizing information across multiple documents, much like the hundred eyes of Argus surveying all around him.

Working on this thesis in the dynamic world of consulting at Accenture, under the guidance of Lars Versnel, has been a transformative experience. It has exposed me to a different way of thinking, bridging the gap between academic research and real-world business applications. This experience has not only enriched my technical skills but also broadened my perspective on how artificial intelligence can be leveraged to solve complex business challenges.

I am particularly grateful to my university supervisor, Neil Yorke-Smith, whose support has been instrumental in the success of this thesis. His guidance in keeping me on track with milestones, and his meticulous tips have been crucial in navigating the complexities of this research. Neil’s expertise and encouragement have been a constant source of motivation, pushing me to explore new ideas and refine my approach.

This research would not have been possible without the support and resources provided by the Faculty of Electrical Engineering, Mathematics and Computer Science at Delft University of Technology. The academic environment fostered by the university has been conducive to pushing the boundaries of what’s possible in information retrieval and artificial intelligence.

As we continue to grapple with the challenges of information overload and the need for efficient knowledge extraction, particularly in corporate intelligence, I hope that the insights and methodologies presented in this thesis will contribute to ongoing efforts in making information more accessible and actionable. While KarGus represents a significant step forward, it also opens up new avenues for future research and development in the field of intelligent information retrieval.

This thesis is not just a culmination of my academic journey, but also a reflection of my passion for leveraging artificial intelligence to create tangible business value. It stands as a testament to the power of combining academic rigour with real-world applicability, and I am excited to see how this work might contribute to future advancements in the field.

Martin Michaux

M.B.S.Michaux@student.tudelft.nl
Delft, August 2024

Abstract

This study introduces KarGus, a novel system for multi-document question answering (MD-QA) designed for diverse domains. KarGus integrates advanced Natural Language Processing techniques with Knowledge Graph (KG) construction and Graph Neural Networks (GNNs) to enhance retrieval performance across various specialized fields. We explore the efficacy of combining semantic similarity, TF-IDF, and Named Entity Recognition features in KG construction and information retrieval. Experimental evaluation on a corpus of 30 documents (1810 pages, 10,853 text chunks) from corporate intelligence demonstrates that KarGus outperforms traditional embedding-based methods, achieving a Recall@5 of 0.850 compared to the baseline's 0.823 ($p < 0.05$). The optimal configuration emphasized semantic similarity (weight 0.75), keyword relevance (0.2), and entity information (0.05). Analysis of the KG structure revealed moderately well-defined community structures and efficient information traversal properties. While GNN models showed promising training results, they underperformed in the retrieval task, highlighting challenges in GNN application to MD-QA. This research contributes to the field of information retrieval by demonstrating the efficacy of integrating NLP techniques with graph-based approaches in MD-QA. The adaptable nature of KarGus suggests potential applications across various specialized domains. Future work will focus on validating cross-domain performance and refining GNN implementations for diverse retrieval tasks.

Keywords: Multi-Document Question Answering (MD-QA), Knowledge Graphs (KG), Natural Language Processing (NLP), Graph Neural Networks (GNN), Information Retrieval (IR), Corporate Intelligence, Retrieval-Augmented Generation (RAG)

Contents

1 Introduction

The field of natural language processing (NLP) has witnessed remarkable advancements in recent years, driven by the development of large language models (LLMs) [Brown et al., 2020]. These models have revolutionized various NLP tasks, including question-answering (QA) systems, enabling more human-like communication and opening up new possibilities for intelligent systems. However, despite their impressive capabilities, LLMs often struggle with factual consistency and completeness [Kamalloo et al., 2023], especially when dealing with rapidly evolving information or specialized domains.

This limitation is particularly evident in multi-document question answering (MD-QA) systems, which often struggle with complex, domain-specific information across various fields. Current approaches typically rely on simple embedding techniques or keyword matching, which fail to capture nuanced relationships and contextual information crucial in specialized domains. Moreover, these systems often lack the ability to dynamically adapt to rapidly evolving knowledge, leading to outdated or incomplete responses.

To address these challenges, researchers have proposed the Retrieval-Augmented Generation (RAG) approach [Lewis et al., 2020b]. RAG combines the generative power of LLMs with a retrieval component that selects relevant context from external knowledge sources. This integration allows RAG models to leverage the strengths of LLMs while grounding their responses in factual, up-to-date information [Lewis et al., 2020a].

However, the effectiveness of RAG models heavily depends on the quality and relevance of the retrieved information. Traditional retrieval methods often fall short in capturing the complex relationships and contextual nuances present in specialized domains. This limitation has motivated the exploration of more sophisticated approaches to enhance the retrieval component of RAG models.

Knowledge Graphs (KGs) have emerged as a promising solution to these retrieval challenges [Ji et al., 2022]. KGs offer several advantages over traditional methods: they capture semantic relationships and interdependencies between concepts, provide a structured representation of information, and enable more nuanced reasoning over complex data. By constructing a KG from a document corpus, the retrieval component can leverage rich semantic information to identify and retrieve relevant information with greater accuracy and contextual awareness [NASTASE et al., 2015].

The construction and traversal of KGs for Information Retrieval (IR) can be further enhanced through advanced NLP techniques [Ye et al., 2023]. Methods such as named entity recognition (NER), term frequency-inverse document frequency (TF-IDF), and text embeddings enable the extraction of meaningful entities, relationships, and semantic information from unstructured text.

Incorporating these NLP-driven insights into the KG construction and traversal process creates a more robust and scalable approach for MD-QA, effectively synthesizing and retrieving information from complex, large-scale datasets.

Additionally, the emergence of Graph Neural Networks (GNNs) [Wu et al., 2019b] has opened up new possibilities for processing and reasoning over KGs. GNNs are deep learning models specifically designed to operate on graph-structured data, allowing them to capture complex relationships and dependencies within KGs. By leveraging GNNs, we can enhance the retrieval process by learning rich representations of entities and relationships that incorporate both textual and structural information. This approach enables more sophisticated reasoning over the KG, potentially uncovering relevant information that may be several hops away from the initial query nodes.

In this research, we introduce KarGus, a novel system that addresses the limitations of existing MD-QA systems by uniquely integrating cutting-edge technologies – LLMs, RAG, KGs, advanced NLP techniques, and GNNs. KarGus innovates by:

1. Combining multiple NLP features (TF-IDF, NER, and semantic embeddings) to create a richer, more nuanced KG representation of document content.
2. Employing a GNN for graph traversal, enabling more effective navigation of complex document structures.
3. Focusing on snippet-level retrieval for more precise answers in specialized contexts.
4. Balancing computational efficiency with retrieval accuracy, making it suitable for real-time applications in various domains.

By leveraging the strengths of each component, KarGus aims to provide a scalable, adaptable, and accurate solution for IR and question-answering in complex information environments. This approach addresses the key challenges in MD-QA, offering a more comprehensive and context-aware system capable of handling the intricacies of specialized domains and rapidly evolving knowledge landscapes.

1.1. Background

1.1.1. Large Language Models

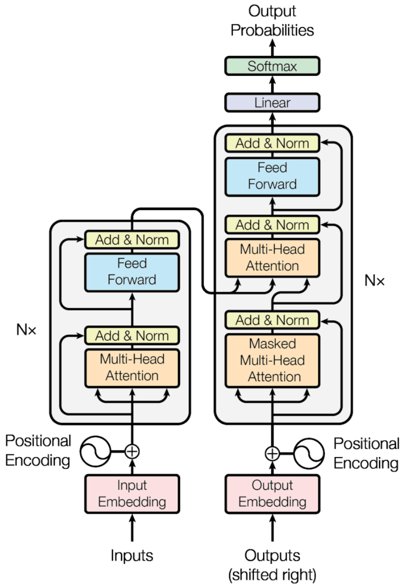


Figure 1.1: Transformer architecture (Source)

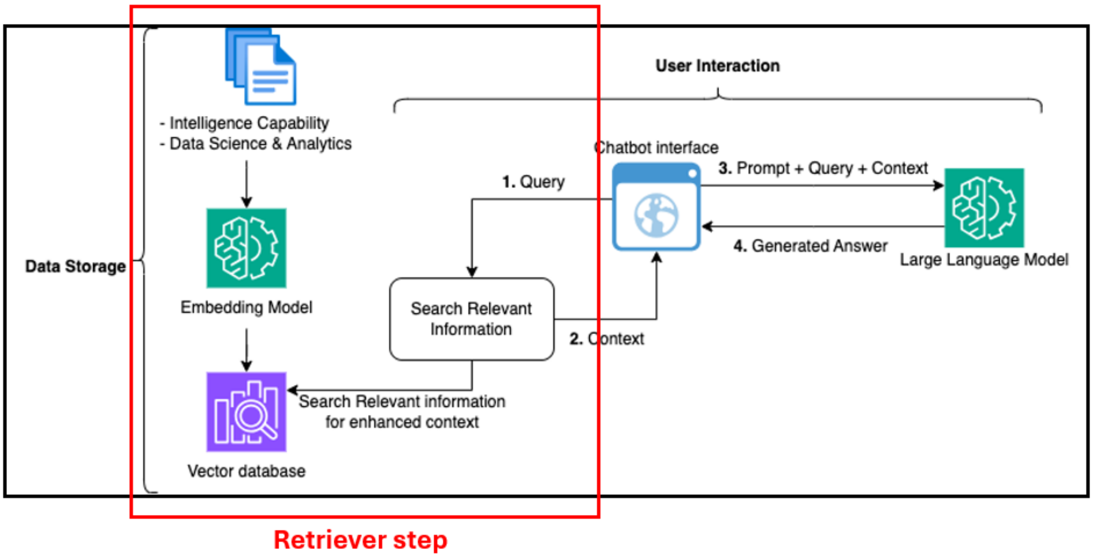
LLMs are transformer-based neural networks that have been pre-trained on massive amounts of textual data, enabling them to acquire a broad understanding of natural language [Brown et al., 2020]. Models like GPT-4, developed by OpenAI, and Gemini, introduced by Google, have demonstrated remarkable capabilities in generating human-like text, answering questions, and even exhibiting some level of reasoning and common-sense understanding [Brown et al., 2020, Team et al., 2024]. An example of the architecture can be seen in Figure 1.1.

LLMs leverage the transformer architecture, which uses self-attention mechanisms to process input sequences in parallel, capturing long-range dependencies more effectively than traditional recurrent neural networks [Vaswani et al., 2017]. The training process involves exposing the model to diverse text corpora, allowing it to learn patterns, relationships, and contextual information from the data. This pre-training phase is typically followed by fine-tuning on specific tasks, enabling the model to adapt its knowledge to particular applications such as question answering [Raffel et al., 2019].

LLMs are trained using self-supervised learning techniques, such as masked language modelling and next-sentence prediction, which allow them to learn the underlying patterns and relationships within the training data [Devlin et al., 2018]. This approach enables the models to develop a rich understanding of language without relying on explicit, labelled data.

Despite their impressive performance, LLMs have limitations [Kaddour et al., 2023]. One significant challenge is their tendency to ”hallucinate” or generate factually incorrect information, especially when dealing with topics or domains not well-represented in their training data [Maynez et al., 2020]. Additionally, LLMs lack direct access to up-to-date, specialized knowledge sources, which can lead to outdated or incomplete responses [Lazaridou et al., 2022].

1.1.2. Retrieval-Augmented Generation



User Interaction
- Intelligence Capability
- Data Science & Analytics
Chatbot interface
3. Prompt + Query + Context
1. Query
4. Generated Answer
Large Language Model
Data Storage
Embedding Model
Search Relevant Information
2. Context
Search Relevant information for enhanced context
Vector database
Retriever step

Figure 1.2: RAG architecture

To address the limitations of LLMs and enhance their performance in QA tasks, researchers have proposed the RAG approach [Lewis et al., 2020b]. RAG models combine the generative power of LLMs with a retrieval component that selects relevant context from external knowledge sources, such as document corpora or knowledge bases [Lewis et al., 2020a].

RAG models consist of two main components: a retriever and a generator. The retriever is responsible for identifying and retrieving relevant information from the knowledge source based on the input query. This component often employs techniques such as dense vector retrieval or semantic search to efficiently find relevant documents or passages [Karpukhin et al., 2020]. The generator, typically an LLM, generates the final output by conditioning on the retrieved context and the query. This two-stage process allows the model to leverage both the broad knowledge captured in the LLM and the specific, up-to-date information from the external knowledge source [Guu et al., 2020].

An example of the architecture can be seen in Figure 1.2 on two corpora of documents of corporate intelligence.

The retrieval component plays a crucial role in the RAG model's performance, as it determines the relevance and completeness of the information provided to the generator. Effective retrieval can mitigate the LLM's tendency to hallucinate and ensure that the generated responses are grounded in factual, up-to-date knowledge [Borgeaud et al., 2021].

1.1.3. Knowledge Graphs for Information Retrieval

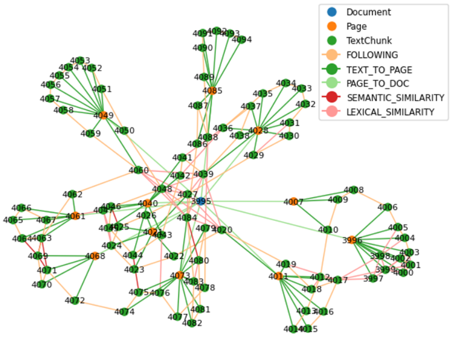


Figure 1.3: Knowledge Graph Example

One promising approach for the retrieval component in RAG models is the use of KGs. As shown in the example of Figure 1.3, a KG is a structured representation of information, where entities (nodes) are connected by relationships (edges), capturing the semantic relationships and interdependencies between concepts [Ji et al., 2022]. By constructing a KG from a document corpus, the retriever can leverage the rich semantic information encoded in the graph to identify and retrieve relevant information more accurately [Wang et al., 2022]. This methodology can replace the retriever step seen in red in the Figure 1.2.

KGs offer several advantages in IR tasks. They provide a structured and interpretable representation of knowledge, allowing for more complex reasoning and inference [Wang et al., 2017]. KGs can capture hierarchical relationships, transitive properties, and domain-specific rules, enabling more nuanced and context-aware retrieval. Additionally, graph-based algorithms can be applied to traverse the KG efficiently, identifying relevant information through multi-hop reasoning or by analyzing the graph's topology [Saxena et al., 2020].

They have been widely used in various domains, such as search engines [Wu et al., 2019a], recommendation systems [Guo et al., 2020], and question-answering systems [Diefenbach et al., 2018], due to their ability to represent and reason over complex relationships between entities. However, constructing a KG from unstructured text data poses several challenges, as traditional methods often rely on rule-based or pattern-matching techniques, which can be brittle and require significant manual effort [Noy et al., 2019].

1.1.4. NLP Techniques for Knowledge Graph Construction

To overcome the limitations of traditional KG construction methods, recent approaches have focused on leveraging advanced NLP techniques to automatically extract entities, relationships, and semantic information from text [Noy et al., 2019]. In this research, we explore the use of the following NLP techniques to enhance the construction and traversal of Knowledge Graphs for IR in RAG models:

Term Frequency-Inverse Document Frequency

TF-IDF [Sammut and Webb, 2010] is a widely used technique in IR and text mining that evaluates the importance of a word or phrase within a document corpus. It combines the term frequency (TF), which measures how frequently a term appears in a document, with the inverse document frequency (IDF), which quantifies how rare the term is across the entire corpus [Sparck Jones, 1988].

TF is defined as:

$$\text{TF}(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$$

where $f_{t,d}$ is the frequency of term t in document d, and the denominator is the sum of the frequencies of all terms in document d.

IDF is defined as:

$$\text{IDF}(t,D) = \log\left(\frac{N}{|\{d \in D : t \in d\}|}\right)$$

where N is the total number of documents in the corpus D, and $|\{d \in D : t \in d\}|$ is the number of documents in which the term t appears.

The TF-IDF score is then calculated as:

$$\text{TF-IDF}(t,d,D) = \text{TF}(t,d) \times \text{IDF}(t,D)$$

TF-IDF provides a numerical measure of a term's relevance to a document, taking into account both its local importance (within a single document) and its global importance (across the entire corpus). This technique helps identify key terms that are both frequent in a specific document and distinctive across the corpus, making it valuable for extracting important concepts and relationships for KG construction [Zhang and Ge, 2019].

Named Entity Recognition

NER is a fundamental NLP task that involves identifying and classifying named entities, such as people, organizations, locations, and other proper nouns, within text [Nadeau and Sekine, 2007]. NER is crucial for extracting and understanding the real-world entities mentioned in documents, enabling the construction of a more comprehensive and semantically rich KG [Lample et al., 2016].

It typically involves training machine learning models (such as conditional random fields or neural networks) on labelled datasets to recognize and classify entities based on their context and linguistic features. Modern NER systems often leverage pre-trained language models and fine-tuning techniques to achieve high accuracy across different domains [Devlin et al., 2018].

Fine-tuning adapts pre-trained models like BERT to domain-specific tasks by updating their parameters with domain-specific data. This process optimizes the model's ability to recognize and classify entities accurately within specialized domains, enhancing its practical utility in applications such as corporate intelligence and information extraction [Kulkarni et al., 2024].

Text Embeddings

Text embeddings are dense vector representations of text that capture semantic and contextual information. These embeddings are typically learned using neural network-based techniques, such as Word2Vec [Mikolov et al., 2013] or BERT [Devlin et al., 2018], which are trained on large text corpora to capture the distributional properties of words and their relationships.

These encoders transform discrete textual data into continuous vector spaces, where semantic similarities can be measured using distance metrics such as cosine similarity. This allows for efficient similarity comparisons, clustering, and other operations that are challenging with raw text data. In the context of KG construction, embeddings can be used to identify semantically related terms, measure the strength of relationships between entities, and even discover latent connections not explicitly mentioned in the text [Wang et al., 2022].

1.1.5. Graph Neural Networks

GNNs are a class of deep learning models designed to operate on graph-structured data. They have gained significant attention in recent years due to their ability to learn and reason over complex relational structures, making them particularly well-suited for tasks involving KGs.

GNNs provide a powerful framework for processing and analyzing graph data. These models work by iteratively updating node representations based on the features of their neighbouring nodes and edges. This process allows GNNs to capture both local and global structural information, enabling them to learn rich representations that can be used for various downstream tasks such as node classification, link prediction, and graph classification [Wu et al., 2019b].

GCN

Graph Convolutional Networks, introduced by Kipf and Welling [Kipf and Welling, 2016], are foundational architecture in the field of GNNs. GCNs generalize the operation of convolution from grid-like data (e.g., images) to graph-structured data. The key idea behind GCNs is to define a convolution operator on graph nodes that aggregates feature information from a node's local neighbourhood.

The basic GCN layer can be described as:

$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)})$$

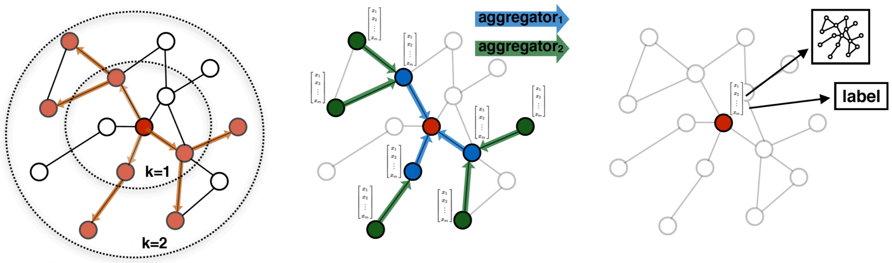
Where $\tilde{A} = A + I_N$ is the adjacency matrix with added self-connections, $\tilde{D}$ is the degree matrix of $\tilde{A}$, $H^{(l)}$ is the matrix of node features at layer l, $W^{(l)}$ is a layer-specific trainable weight matrix, and σ is a non-linear activation function.

GCNs offer several advantages:

- **Spectral-based approach:** GCNs leverage the spectral theory of graphs, allowing them to efficiently capture global graph properties.
- **Computational efficiency:** The localized first-order approximation of spectral graph convolutions allows for efficient computation, even on large graphs.
- **Inductive capability:** GCNs can generalize to unseen nodes, making them suitable for evolving graph structures.

GraphSAGE

Graph Sample and Aggregate is a popular and effective GNN architecture introduced by [Hamilton et al., 2017b]. It is designed to generate embeddings for nodes in large-scale graphs by leveraging local neighbourhood information.



1. Sample neighborhood

2. Aggregate feature information from neighbors

3. Predict graph context and label using aggregated information

Figure 1.4: GraphSAGE architecture (Source)

As shown on 1.4, it processes graphs iteratively, sampling neighbors for each node and aggregating their information. The aggregation function, a key component, can vary based on the specific implementation. Common choices include mean, max, or sum pooling of neighbor features. More advanced versions may use learnable neural network layers or attention mechanisms. This function combines neighbor data into a fixed-size vector, regardless of the number of neighbors. The algorithm then merges this aggregated information with the node's own features, often using concatenation followed by a non-linear transformation. This process repeats for several iterations, expanding each node's effective neighbourhood. The result is a set of node representations capturing both local and broader graph structure, useful for various downstream tasks. GraphSAGE's flexibility in choosing the aggregation function allows it to adapt to different graph structures and tasks.

This technique offers several key advantages:

- **Inductive learning:** Unlike many other GNN models, GraphSAGE can generate embeddings for unseen nodes, making it suitable for dynamic or evolving graphs [Hamilton et al., 2017b].
- **Scalability:** GraphSAGE uses a sampling strategy to aggregate information from a node's neighbourhood, allowing it to scale to large graphs where considering all neighbors would be computationally infeasible [Ying et al., 2018].
- **Flexibility:** The model can incorporate various aggregation functions (e.g., mean, max, LSTM) and can be easily integrated with different types of node features [Xu et al., 2018].

By leveraging GraphSAGE in the context of KG-based IR, we can learn rich representations of entities and relationships that capture both textual and structural information. This can lead to more accurate and contextually relevant retrieval results, enhancing the overall performance of the QA system [Wang et al., 2022].

1.2. Literature Review

Recent years have seen significant advancements in addressing the challenges of MD-QA, paving the way for more sophisticated approaches to IR and synthesis. This section explores key contributions in this field, highlighting their approaches, strengths, and limitations in comparison to our proposed KarGus system.

One notable advancement is the development of retrieval-augmented language models. Izacard et al. [Izacard et al., 2022] proposed a few-shot learning approach with retrieval-augmented language models, demonstrating improved performance on complex QA tasks without extensive fine-tuning. This work highlights the potential of combining retrieval mechanisms with powerful language models. However, their approach struggles with domain-specific knowledge that requires a more nuanced understanding.

In the realm of knowledge integration, Liu et al. [Yang et al., 2023] introduced a novel few-shot learning method leveraging knowledge graph-based self-supervision. Their approach shows promise in enhancing model performance on domain-specific tasks with limited training data, a crucial consideration for specialized applications. While effective, this method does not fully capture the complex relationships between documents in a large corpus as it is only LLM-prompted-based.

Wang et al. [Wang et al., 2023] introduced KG Prompting for MD-QA, utilizing a KG to generate prompts for LLMs. While effective, this approach requires multiple LLM interactions per query, potentially increasing computational demands and response times. In contrast, our KarGus system employs a more efficient one-time KG traversal using a trained model.

Lee and Kang [Min et al., 2019] presented a method leveraging a graph structure to improve text retrieval for open-domain question answering. Their approach combines structured knowledge bases with unstructured text. However, their heavy reliance on TF-IDF for initial text representation does not fully capture semantic nuances. KarGus addresses this limitation by incorporating more advanced NLP techniques, including semantic embeddings and named entity recognition.

Seonwoo et al. [Seonwoo et al., 2022] proposed a Virtual Knowledge Graph for zero-shot domain-specific document retrieval. While innovative, their focus on document-level retrieval differs from KarGus's snippet-focused approach, which allows for more granular and precise answers.

The challenge of processing and reasoning over large-scale document collections has been addressed by Khattab et al. [Khattab et al., 2023], who proposed a "Demonstrate-Search-Predict" framework composing retrieval and language models for knowledge-intensive NLP tasks. This approach demonstrates potential for scalable and adaptable QA systems but does not fully leverage the structural information present in complex document sets.

Lu et al. [Lu et al., 2019] introduced QUEST, a system for answering complex questions using multi-document evidence. While effective for certain types of complex questions, their triple-based KG construction strategy is limited in capturing more nuanced document relationships. Our approach uses a broader range of NLP features and graph structures, potentially allowing for more flexible and comprehensive document representation.

Other advancements in multi-modal learning, as explored by Zhang et al. [Zhang et al., 2024a], have opened new avenues for MD-QA by integrating diverse data types. This work points towards future directions where textual, visual, and potentially other modalities of information can be seamlessly integrated for more comprehensive question answering.

The explainability of QA systems, crucial for their adoption in various settings, has seen progress with work like that of Wiegreffe and Marasović [Wiegreffe and Marasovic, 2021]. Their review of datasets for explainable NLP provides insights into developing more transparent and interpretable QA systems, an aspect that KarGus aims to address through its graph-based approach.

While these advancements have significantly pushed the boundaries of MD-QA, challenges remain in developing systems that can effectively handle the complexities of varied environments. These include the need for domain adaptability, handling of highly specialized information, and the ability to reason over large, diverse document sets. By addressing these aspects, KarGus aims to address these persistent challenges in the context of different domains for MD-QA, offering a more comprehensive and adaptable solution to the challenges posed by complex, multi-document information retrieval and question answering.

1.3. Motivation

The field of MD-QA [Rajpurkar et al., 2016] faces significant challenges that necessitate the development of more effective and scalable approaches. Existing methods, particularly those relying on traditional embedding-based indexing, struggle to capture the intricate semantic relationships and interdependencies among entities and concepts. This limitation often results in inaccurate retrievals and suboptimal responses, especially in domains requiring deep contextual understanding, such as corporate intelligence, legal research, and academic inquiries.

Embedding-based methods, while useful for retrieval tasks, struggle with complex questions like the question 1.5. They often focus on matching keywords in the question with keywords in documents, potentially overlooking the deeper meaning and relationships between concepts [Arseniev-Koehler, 2021, Bojanowski et al., 2017]. For instance, in our example, an embedding method might ”recognize” documents containing ”goals” and ”researchers” but miss the crucial connection to ”quantum computers” and the specific intent of finding researchers’ goals in that domain.

Furthermore, as the volume and complexity of information continue to grow exponentially, these shortcomings become increasingly pronounced. The vast and intricate nature of modern datasets demands methods that can effectively synthesize and interpret information from multiple sources, which traditional systems fail to accomplish. Additionally, these systems often lack the ability to dynamically adapt to rapidly evolving knowledge, leading to outdated or incomplete responses.

The integration of domain-specific knowledge [Gururangan et al., 2020] with general language understanding remains a critical challenge. While LLMs have made significant strides in NLP, they often lack the capacity to ground their responses in specific, up-to-date domain knowledge. This gap can result in inconsistencies, outdated information, or even fabricated responses when dealing with specialized or rapidly changing fields.

Scalability is another pressing concern [Khattab et al., 2023]. The growing volume of information necessitates approaches that can efficiently process and analyze large-scale datasets without compromising quality or relevance. This challenge is particularly acute in enterprise environments, where quick and accurate information retrieval from vast document repositories can significantly impact decision-making processes.

These limitations underscore the need for a novel approach that can address the multifaceted challenges of MD-QA. Such an approach should be capable of capturing complex semantic relationships, adapting to diverse and evolving datasets, integrating domain-specific knowledge, and scaling efficiently to handle large volumes of information. By addressing these challenges, a new system has the potential to transform information retrieval and question answering across a broad spectrum of fields, enabling more reliable, insightful, and contextually relevant interactions with complex information spaces.

What are the two major goals that researchers are currently working on in order to improve quantum computers?

Figure 1.5: Example of a complex query

1.4. Proposed Approach

To address the limitations of existing MD-QA systems, we introduce KarGus, a novel system that integrates cutting-edge technologies – LLMs, RAG, KGs, advanced NLP techniques, and GNNs [Izacard et al., 2022, Opdahl and Nunavath, 2020]. Unlike previous KG-based approaches that rely solely on entity relationships, KarGus incorporates semantic similarity, TF-IDF, and named entity recognition to create a more comprehensive and nuanced representation of document content.

The name ”KarGus” is inspired by Argus Panoptes, the all-seeing giant in Greek mythology known for his hundred eyes, reflecting our system’s aim to ”see” and process all relevant information within a document corpus. Figure 1.6 illustrates how KarGus captures and interconnects multiple aspects of information.

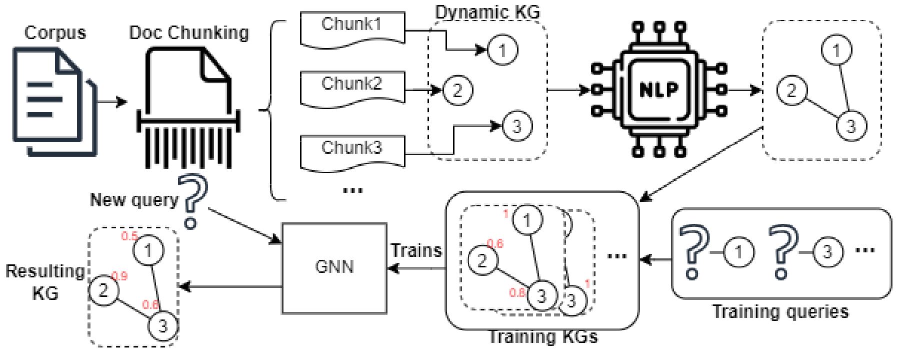


Figure 1.6: KarGus Architecture

1.4.1. Advanced NLP Integration

The first core objective of our research focuses on how we can leverage the strengths of advanced NLP techniques to enhance MD-QA performance. This leads to our first research question:

Research Question 1: How can the unique combination of advanced NLP techniques in KarGus enhance MD-QA performance compared to traditional single-feature approaches?

To address this question, we conduct a comprehensive analysis of NLP features and their heuristics. Our approach includes:

- An ablation study to examine individual feature contributions
- A weight study to analyze collective feature effects on the KG's construction and scoring
- An impact study of different relationship types in the KG's traversal

This analysis aims to identify the most effective combination of NLP features and heuristic weights for constructing and traversing the KG, optimizing its ability to capture relevant information and relationships.

1.4.2. Heuristic Optimization

While integrating multiple NLP features can potentially improve performance, it's crucial to understand their individual and collective impacts on the system's retrieval capabilities. This leads to our second research question:

Research Question 2: What is the impact of different NLP feature combinations, heuristic weights, and relationship types on the system's retrieval performance?

To address this question, we focus on:

- Fine-tuning the system's performance by adjusting the influence of each NLP feature
- Analyzing the impact of different relationship types on information retrieval within the graph
- Optimizing the balance between feature complexity and retrieval accuracy

This approach allows us to develop a more nuanced understanding of how different aspects of our NLP-driven KG construction and traversal affect the overall performance of KarGus.

1.4.3. GNN Integration and Performance Comparison

The integration of GNNs represents a significant advancement in our approach. By learning from the rich structural information encoded in the KG, along with the NLP-derived features, the GNN can discover complex patterns and relationships that may not be apparent through traditional retrieval methods. This leads to our third research question:

Research Question 3: How does KarGus's retrieval performance, leveraging its innovative KG and GNN approach, compare to traditional embedding-based methods in diverse, specialized domains? Furthermore, how do different GNN architectures, specifically GCN and GraphSAGE, perform relative to each other in this context?

To address this question, we:

- Develop, train, and test both GCN and GraphSAGE GNNs within the KarGus system
- Use NLP techniques and their heuristic scores as node features, incorporating the graph structure
- Compare the performance of GCN and GraphSAGE against each other and against traditional baseline methods in information retrieval
- Analyze the strengths and weaknesses of each GNN architecture in the context of our MD-QA task

This comprehensive comparison not only positions KarGus against traditional methods but also provides insights into the relative merits of different GNN architectures for MD-QA tasks. By evaluating both GCN and GraphSAGE, we aim to understand which architecture is better suited for capturing the complex relationships in our knowledge graphs and translating that understanding into improved retrieval performance.

KarGus combines NLP-driven KG construction, heuristic-based traversal, and GNN-powered reasoning to transform information retrieval across diverse fields. Unlike traditional embedding methods that rely on keyword matching, KarGus analyzes conceptual connections and captures long-range dependencies within queries. This approach enables deeper contextual understanding, making it particularly effective for complex, multi-document queries in domains such as corporate intelligence, legal research, and academic literature review.

Referring back to the example in Figure 1.5, KarGus can discern the relationship between ”goals” and ”researchers improving quantum computers,” ensuring relevant answers to nuanced inquiries. By addressing our research questions, KarGus aims to provide an accurate, context-aware solution for information retrieval and question answering in complex environments. This has the potential to significantly enhance how organizations extract insights from large, diverse document sets, leading to more informed decision-making and knowledge discovery.

2 Article

2.1. Introduction

2.1.1. Background

The progress of Large Language Models (LLMs) has transformed question-answering systems [Chen and Zeng, 2013], enabling sophisticated query comprehension and coherent answer generation [Lewis et al., 2020a]. Models like GPT-4 and Gemini [Chowdhery et al., 2022] showcase remarkable capabilities in human-like text generation and reasoning [Kamalloo et al., 2023]. However, applying LLMs to multi-document question answering (MD-QA) across diverse domains presents significant challenges, particularly in ensuring factual accuracy and avoiding hallucinations [Kamalloo et al., 2023].

To address these limitations, researchers have developed the Retrieval-Augmented Generation (RAG) approach [Lewis et al., 2020b, Izacard et al., 2022], which combines LLMs with external knowledge retrieval. While RAG models improve factual grounding, current methods often rely on embedding vector databases [Reimers and Gurevych, 2019], which struggle to capture complex semantic relationships in large-scale information spaces [Arseniev-Koehler, 2021, Bojanowski et al., 2017].

Existing MD-QA systems, typically based on simple embedding techniques or keyword matching, often fail to capture the nuanced relationships and contextual information crucial in specialized domains. This limitation underscores the need for more advanced approaches that can effectively handle complex, domain-specific queries across various fields.

Recent advancements in QA systems have highlighted persistent challenges in scalability and integration, particularly when dealing with complex, multi-document scenarios [Zhang et al., 2024b]. Existing research struggles to efficiently integrate advanced NLP techniques within retrieval systems that can adapt to vast and varied datasets across diverse domains such as legal research, academic inquiries, and specialized industries [Gupta, 2011, Bowman et al., 2015, Liu et al., 2018, Zhang et al., 2019, Liu et al., 2020, Arbaaeen and Shah, 2021].

Knowledge Graphs (KGs) offer a promising solution to these challenges [NASTASE et al., 2015]. By capturing semantic relationships and interdependencies between concepts, KGs enable more accurate and context-aware information retrieval [Lin et al., 2021, Probierz et al., 2023]. The construction and traversal of KGs can be further enhanced through advanced NLP techniques such as named entity recognition (NER), term frequency-inverse document frequency (TF-IDF), and text embeddings [Marwan Omar, 2022, Schneider et al., 2022]. These techniques allow for the extraction of meaningful entities, relationships, and semantic information from unstructured text, creating a more robust and scalable approach for MD-QA.

The emergence of Graph Neural Networks (GNNs) has opened up new possibilities for processing and reasoning over KGs [Hamilton et al., 2017b, Wu et al., 2019b]. GNNs can capture complex relationships and dependencies within KGs, potentially uncovering relevant information that may be several hops away from the initial query nodes. These graph-based approaches have shown promise in improving performance in IR and question-answering tasks [Lin et al., 2021, NASTASE et al., 2015, Probierz et al., 2023, Schneider et al., 2022].

Despite these advancements, there remains a need for a unified approach that effectively combines these technologies to address the multifaceted challenges of MD-QA across diverse domains. This research aims to fill this gap by developing a novel system that integrates advanced NLP techniques, KG construction, and GNN-based reasoning to provide more accurate, context-aware, and scalable solutions for complex information retrieval tasks.

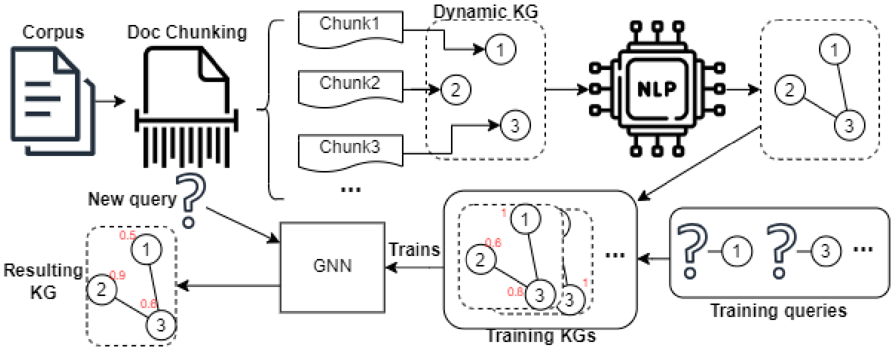


Figure 2.1: KarGus Architecture - First processes document corpora through chunking and dynamic KG node construction. NLP techniques, including semantic similarity, NER, and TF-IDF, connect these nodes based on similarity using KNN. Then GNN learns node features (NLP scores), edge types (semantic/lexical similarities), and graph structure. Finally, for new queries, GNN leverages these learned features to traverse the KG, producing a refined KG for accurate IR.

2.1.2. Proposed Approach

To address the limitations of existing MD-QA methods in handling complex, domain-specific queries and capturing nuanced relationships within large document sets, we propose KarGus, a novel system integrating advanced NLP techniques, dynamic KG construction, and GNNs for enhanced MD-QA across diverse domains [Liu et al., 2022].

KarGus's architecture consists of two primary modules: Graph Construction and Graph Traversal. The Graph Construction module dynamically builds a KG from multiple documents using advanced NLP techniques, capturing the interconnected nature of information across documents as seen in Figure 3.5. The Graph Traversal module employs a GNN to navigate this KG and retrieve relevant information based on complex queries, enabling the system to uncover information several hops away from initial query nodes.

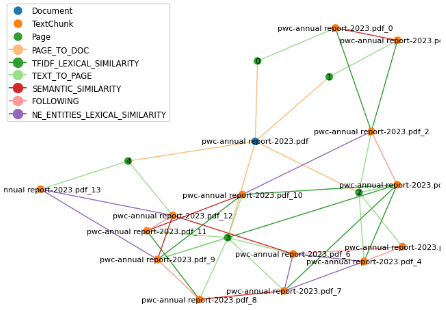

Document
TextChunk
Page
PAGE_TO_DOC
TFIDF_LEXICAL_SIMILARITY
TEXT_TO_PAGE
SEMANTIC_SIMILARITY
FOLLOWING
NE_ENTITIES_LEXICAL_SIMILARITY
pwc-annual report-2023.pdf_0
pwc-annual report-2023.pc
0
1
pwc-annual report-2023.pdf_2
pwc-annual report-2023.pdf
4
nnual report-2023.pdf_13
pwc-annual report-2023.pdf_10
pwc-annual report-2023.pc
2
pwc-annual report-2023.pdf_12
pwc-annual report-2023.pdf_11
3
pwc-annual report-2023.pdf_6
pwc-annual report-2023.p
pwc-annual report-2023.pdf_9
pwc-annual report-2023.pdf_4
pwc-annual report-2023.pdf_7
pwc-annual report-2023.pdf_8

Figure 2.2: Subgraph of the KarGus Knowledge Graph

KarGus contributes to the field of MD-QA through several key innovations:

- A multi-faceted NLP approach combining semantic similarity, TF-IDF, and named entity recognition for comprehensive document representation.
- Dynamic KG construction that captures both semantic and lexical relationships across multiple documents.
- GNN-based traversal for effective navigation of complex information structures.
- A scalable and adaptable architecture balancing computational efficiency with retrieval accuracy.

This approach allows KarGus to handle queries requiring deep contextual understanding, going beyond simple keyword matching. By leveraging the rich structure of the knowledge graph, the system improves the relevance and semantic connectivity of retrieved information, addressing a key challenge in complex MD-QA tasks.

Our experimental results demonstrate that KarGus outperforms traditional embedding-based methods [Arseniev-Koehler, 2021, Bojanowski et al., 2017], particularly in capturing both broad contextual information and specific entity relationships. This makes it well-suited for domains requiring nuanced understanding of complex document sets, such as corporate intelligence.

While this article's example is focused on corporate intelligence, KarGus's design allows for potential adaptation to other domains like legal research and academic literature review. Future work includes exploring the integration of KarGus within a RAG system, potentially enabling even more powerful and context-aware information retrieval and generation capabilities.

2.1.3. Related work

Recent research has made significant strides in integrating KGs into MD-QA systems. Wang et al. [Wang et al., 2023] introduced KG Prompting, which uses KGs to generate prompts for LLMs. While effective, this approach requires multiple LLM interactions per query, potentially increasing computational demands. In contrast, using a trained agent, KarGus employs a more efficient one-time KG traversal.

Lee and Kang [Min et al., 2019] combined structured knowledge bases with unstructured text for open-domain question answering. However, their heavy reliance on TF-IDF for initial text representation may not fully capture semantic nuances. KarGus addresses this limitation by incorporating more advanced NLP techniques, including semantic embeddings and named entity recognition.

Seonwoo et al. [Seonwoo et al., 2022] proposed a Virtual Knowledge Graph (VKG) for zero-shot domain-specific document retrieval. While innovative, their focus on document-level retrieval differs from KarGus's segment-focused approach, which allows for more granular and precise answers.

Other notable works include Xu and Wallace's [Xu and Lapata, 2020] KG-based approach for query-focused multi-document summarization, Lu et al.'s [Lu et al., 2019] QUEST system for complex question answering, and Kang and Kim's [Kang et al., 2023] method for maintaining knowledge consistency in open-domain dialogues. While these approaches have advanced the field, they often focus on specific subtasks or domains.

Table 2.1 provides a comparative overview of KarGus and other the recent approaches in MD-QA mentioned in this section. KarGus builds upon prior works by uniquely integrating multiple NLP features in graph construction, utilizing a GNN for traversal, and focusing on segment-level retrieval. This approach allows for more precise and contextually relevant IR. By balancing broad context with specificity, KarGus can handle complex, information-dense documents and provide more accurate answers across various domains. Its domain adaptability further sets it apart from existing solutions, pushing the boundaries of MD-QA beyond previous methods.

2.2. Methodology

This section details the methodological approach employed in the KarGus system, focusing on integrating advanced NLP techniques with KG construction and traversal methods. Figure 2.1 illustrates the overall architecture of the KarGus system, comprising three main components: NLP Feature Extraction, Knowledge Graph Construction, and GNN-based Retrieval. We begin by describing the NLP heuristics utilized to capture semantic and lexical relationships within the document corpus. Subsequently, we explain the process of KG construction and the GNN-based retrieval mechanism.

2.2.1. NLP Heuristics

To effectively capture semantic and lexical relationships within and between documents, we employ three complementary NLP heuristics: Text Embedding, TF-IDF, and NER scores. These heuristics were selected based on their proven effectiveness in various NLP tasks [Marwan Omar, 2022] and their ability to capture different aspects of textual similarity. The Figure 3.1 in the Appendix section illustrates the integration of these features in our system.

For both chunk-chunk (when constructing the dynamic KG) and query-chunk (when receiving a query) comparisons, we utilize a unified set of comparison functions:

Text Embedding Comparison

$$\text{embed\_compare}(x, y) = \cos(\text{embed}(x), \text{embed}(y)) \tag{2.1}$$

We employ the OpenAI ”Ada 002” Text Encoder model to generate dense vector representations for both document chunks and queries. This model was chosen for its state-of-the-art performance in capturing nuanced contextual information [Reimers and Gurevych, 2019]. The embedding process transforms text into a 1536-dimensional vector space, where seman-

Approach	Multi-NLP Features	Dynamic KG Construction	GNN Integration	Segment-level Retrieval	Domain Adaptability
KarGus (Ours)	✓	✓	✓	✓	High
[Wang et al., 2023]		✓			Medium
[Min et al., 2019]					Low
[Seonwoo et al., 2022]	✓	✓			High
[Xu and Lapata, 2020]		✓			Medium
[Lu et al., 2019]		✓		✓	Low

Table 2.1: Comparison of KarGus with Related Works

tic similarities can be efficiently computed using cosine similarity.

This approach allows us to identify contextually similar content that may not share exact lexical matches, enabling the system to capture nuanced relationships between different parts of a document or between a query and document chunks, enabled by the Equation 2.1. The use of pre-trained embeddings also provides a level of transfer learning, allowing our system to leverage semantic knowledge gained from large-scale language modeling.

TF-IDF Comparison

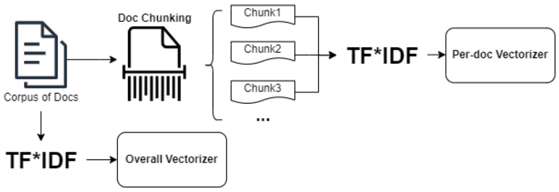


Figure 2.3: TF-IDF pipeline

Our TF-IDF implementation, illustrated in Figure 2.3 and enabled by Equation 2.4, balances the local importance of a term within a document (TF) with its global importance across the corpus (IDF) [Sammut and Webb, 2010].

$$\text{tfidf}_1(x, y) = \alpha \cdot \frac{|K_x \cap K_y|}{\max(|K_x|, |K_y|)} \tag{2.2}$$

$$\text{tfidf}_2(x, y) = (1 - \alpha) \cdot \cos(\text{tfidf}(x), \text{tfidf}(y)) \tag{2.3}$$

$$\text{tfidf\_compare}(x, y) = \text{tfidf}_1(x, y) + \text{tfidf}_2(x, y) \tag{2.4}$$

The process involves two stages of vectorization:

- For each document, we compute an ”overall vectorizer” based on the full corpus. This vectorizer captures the global importance of terms across all documents.
- We create a ”per-doc vectorizer” for each chunk within the document. This local vectorizer considers the top 20 keywords per chunk, optimizing computational efficiency while maintaining comprehensive representation.

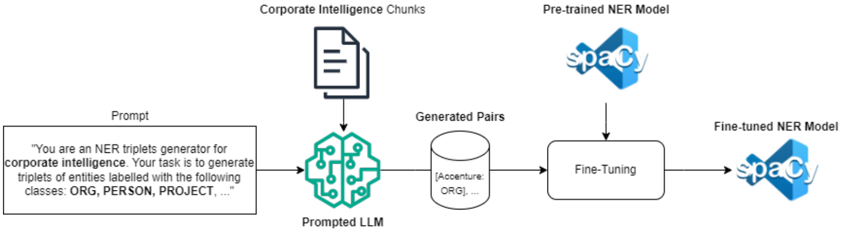
NER Comparison

NER is employed to identify and leverage key entities within the text. To enhance performance for our specific task, we fine-tuned a general SpaCy NER model. This fine-tuning process, showed in Figure 2.4, involved generating NER pairs (entitY:label) using prompt engineering with the OpenAI GPT-3.5 Turbo model, followed by training the existing SpaCy open-source model on these generated pairs. This fine-tuning was necessary to adapt the model to the specific entity types and naming conventions present in our corpus, particularly for domain-specific terms that may not be well-represented in general-purpose NER models.

We utilize this fine-tuned SpaCy NER model and compare to other chunks or queries using Equation 2.9. We compute the named entities and generate Word2Vec (W2V) vectors for each identified entity. The choice of W2V for entity representation was based on its ability to capture semantic relationships between entities in a computationally efficient manner [Mikolov et al., 2013].

$$\text{ner}_1(x, y) = \beta \cdot \frac{|E_x \cap E_y|}{\max(|E_x|, |E_y|)} \tag{2.5}$$

$$\text{sim}(e_x, E_y) = \max_{e_y \in E_y} \cos(\text{W2V}(e_x), \text{W2V}(e_y)) \tag{2.6}$$



"You are an NER triplets generator for **corporate intelligence**. Your task is to generate triplets of entities labelled with the following classes: **ORG, PERSON, PROJECT**, ..."

Figure 2.4: NER Model Fine-tuning Process

$$\text{ner}_{2a}(x, y) = \frac{1}{|E_x|} \sum_{e_x \in E_x} \text{sim}(e_x, E_y) \tag{2.7}$$

$$\text{ner}_{2b}(x, y) = (1 - \beta) \cdot \text{ner}_{2a}(x, y) \tag{2.8}$$

$$\text{ner\_compare}(x, y) = \text{ner}_1(x, y) + \text{ner}_{2b}(x, y) \tag{2.9}$$

Overall

In these comparison equations, x and y represent either two chunks or a query and a chunk, as clarified in Table 2.2. K_x and K_y are the keyword sets, E_x and E_y are the entity sets, and α and β are the weighting factors (empirically set to initially 0.5). This unified approach allows for consistent comparison across different textual units while capturing multiple dimensions of similarity. Examples of comparing two chunks in each NLP feature are displayed in Figures 3.2, 3.3 and 3.4 in the Appendix.

The distinction in TF-IDF vectorizers, observed in Figure 2.3 between chunk-chunk and query-chunk comparisons is crucial for both scalability and representation balance. For chunk-chunk comparisons within a document, we use a document-specific vectorizer, allowing for efficient updates when new documents are added to the corpus without requiring recomputation of existing comparisons. In contrast, query-chunk comparisons utilize an overall corpus vectorizer, providing a more comprehensive lexical representation across the entire document set. This approach strikes a balance between computational efficiency in dynamic KG construction and the need for broader semantic/lexical representation when processing queries, ensuring that our system remains scalable while maintaining robust retrieval capabilities.

This multi-faceted process enables KarGus to capture both broad contextual information and specific entity relationships, crucial for handling complex queries in specialized domains like corporate intelligence. By combining semantic understanding with lexical matching and entity recognition, our system can effectively process a wide range of query types, from broad topical inquiries to specific entity-focused questions, while maintaining consistency between document analysis and query processing methodologies.

2.2.2. Knowledge Graph Construction

The construction of the KG in the KarGus system leverages the NLP heuristics described above to create a rich, interconnected representation of the document corpus [Ye et al., 2023]. The process involves two main steps: document processing and graph connectivity.

Documents Processing:
We employ the Langchain tool for text chunking, configured to segment documents into chunks of 512 tokens with an overlap of 64 tokens, using a RecursiveCharacterTextSplitter.

The chunk size of 512 tokens aligns with the typical input size of many transformer-based models, facilitating potential future integrations. The overlap of 64 tokens helps maintain context continuity between chunks, reducing the risk of splitting important semantic units. Each resulting node, representing a document chunk, is enriched with the NLP features described earlier: TF-IDF scores, NER entities and their embeddings, and text embedding similarity scores.

Graph Connectivity:
To establish connections between nodes, we utilize the K-nearest neighbors (KNN) algorithm [Peterson, 2009], leveraging the computed NLP features. The connectivity process focuses on linking nodes within the same document, employing a refined set of criteria based on our NLP heuristics.

The choice of KNN was motivated by its ability to create a sparse yet meaningful graph structure, balancing connectivity with computational efficiency. Alternative methods such as threshold-based connectivity or fully-connected graphs were considered but found to be either too restrictive or computationally prohibitive for large document sets.

Lexical features are evaluated using:

- The length of common TF-IDF keywords
- Cosine similarity of TF-IDF vectors
- Commonality of NER entities
- Average cosine similarities of their Word2Vec matrices

Semantic features are assessed through the cosine similarity of text embedding vectors, enabling the connection of contextually similar nodes that may differ lexically.

The choice of $k = 5$ for the number of nearest neighbors was determined empirically, offering an optimal balance between connectivity and sparsity. As illustrated in our experiments in Figure 3.6 of the Appendix, this configuration maintains essential semantic links without excessive density, which could otherwise impede performance with irrelevant connections [Lin et al., 2021].

2.2.3. Features Analysis

To optimize the retrieval accuracy of the KarGus system, a comprehensive feature analysis is conducted. This analysis is essential to understand the individual and collective impact of various NLP features and heuristic weights on the system's performance. Our methodology consists of several stages, described below:

Evaluation of NLP Features: Initially, we focus on evaluating the performance of different combinations of the NLP features. This step involves testing each feature individually as well as in all possible combinations. To maintain consistency across tests, equal heuristic weights are applied to each feature during this phase. The aim is to identify which features contribute most significantly to accurate retrieval outcomes and how they interact when combined.

Assessment of Heuristic Weights: Upon determining the effective combinations of NLP features, we proceed to test varying heuristic weights for these combinations. The objective here is to ascertain the most impactful weights, which will inform the configuration of our GNN. Different sets of weights are applied to evaluate how they influence the retrieval efficacy of the system. This process involves a meticulous analysis to determine the optimal weight distribution that maximizes the relevance and accuracy of the retrieved documents.

Analysis of Pathways to Ground Truth In instances where the ground truth node does not appear among the initial set of retrieved nodes, we compute the shortest path from the starting nodes to the ground truth node. During this traversal, we track the types of edges encountered—categorized into semantic, tf-idf lexical, and NER lexical similarity edges. This tracking helps us to identify which types of connections are most frequently utilized in reaching the correct answers. The analysis of these pathways provides critical insights into the relative importance of different edge types in the graph structure, guiding further refinements in the graph's construction and the strategic application of heuristic weights.

To implement this analysis, we employ a series of experiments where each configuration of features and weights is tested against a benchmark dataset. The performance metrics from these experiments are collected and analyzed to derive statistical significance and practical relevance. The findings from this comprehensive feature analysis will directly inform the enhancement strategies for the KarGus system, ensuring that our system not only performs effi-

Score Type	**Chunk-Chunk Comparison**	**Query-Chunk Comparison**
Semantic	embed_compare(c_1, c_2)	embed_compare(q, c)
TF-IDF	tfidf_compare$_{\text{doc}}(c_1, c_2)$*	tfidf_compare$_{\text{corpus}}(q, c)$**
NER	ner_compare(c_1, c_2)	ner_compare(q, c)

Table 2.2: Comparison Functions for Chunk-Chunk and Query-Chunk, where c_1 and c_2 are document chunks, q is the query, c is a document chunk, * means using per-document vectorizer, and ** means using overall corpus vectorizer.

ciently but also scales effectively across diverse datasets. It also contributes substantially to the refinement of our retrieval system, setting a robust foundation for the subsequent integration and optimization of the GNN within the KarGus framework.

2.2.4. Graph Neural Network

To enhance the retrieval performance of our system, we implemented two GNN architectures: Graph SAmple and aggreGatE (GraphSAGE) and Graph Convolutional Network (GCN). These approaches leverage both the node and edge features derived from our NLP heuristics and the structural information encoded in our KG.

GraphSAGE: Graph SAmple and aggreGatE is our primary GNN model [Hamilton et al., 2017b]. It is an inductive framework for node embedding that aggregates feature information from a node's local neighborhood. We chose GraphSAGE for several key reasons:

- Inductive learning capability, crucial for our dynamic KG where new document chunks may be added over time.
- Scalability to large graphs, supporting our goal of handling extensive document corpora efficiently.
- Flexibility in aggregation functions, allowing us to effectively incorporate our diverse NLP features (TF-IDF, NER, and text embeddings).
- Neighborhood aggregation approach, which aligns well with our graph construction method based on semantic and lexical similarities.

GCN: As a baseline comparison, we also implement a Graph Convolutional Network [Kipf and Welling, 2017]. GCN is a fundamental GNN architecture that allows us to assess the tradeoffs between the more advanced GraphSAGE model and a simpler GNN architecture in the context of our MD-QA task.

Model Implementation and Training

In implementing our GNN models, we prioritized computational efficiency and scalability. We used NLP feature scores (TF-IDF, NER, and semantic similarity) as node features, avoiding high-dimensional text embedding vectors to reduce processing time and memory requirements. This decision aligns with research by [Xu et al., 2018], who demonstrated that carefully selected low-dimensional features can yield comparable performance to full embeddings in graph-based NLP tasks.

Using the parameters described in 3.7.4, our training process for both GNN models involves several steps:

1. **Dataset Preparation:** We utilized a dataset of 200 question-answer pairs, with 150 pairs designated for training and 50 for testing.
2. **Feature Computation:** Heuristic scores comparing each question to all KG nodes formed the initial feature set, capturing relevance across multiple dimensions.
3. **Score Assignment:** We employed a novel scoring strategy:
   - Ground truth nodes received a high score (1.0), establishing clear targets for the model.
   - Other nodes were scored based on heuristic scores and weights from our feature analysis.
   - Neighborhood nodes had enhanced scores, decreasing with graph distance from the ground truth. This approach creates a relevance gradient that helps the model learn both content and structural importance.
4. **Model Training:** GraphSAGE and GCN models were trained using node features (heuristic scores), edge connections (the structure of the KG), edge features[1] (type and weight), and target importance scores. This comprehensive input allows the models to learn complex patterns in the graph structure and content.
5. **Evaluation:** We used the test set to assess model performance, focusing on ground truth node ranking and the relevance of top-k retrieved nodes. This evaluation approach is consistent with standard practices in information retrieval research.

[1] The edge features include the type (between TFIDF_LEXICAL_SIMILARITY, NE_ENTITIES_LEXICAL_SIMILARITY and SEMANTIC_SIMILARITY) and the weight (between 0 and 1 representing how similar the two nodes are).

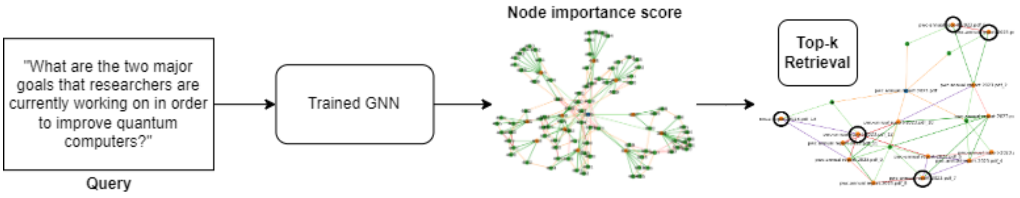

"What are the two major goals that researchers are currently working on in order to improve quantum computers?"
Query
Trained GNN
Node importance score
Top-k Retrieval

Figure 2.5: GNN Retriever

By integrating GNN models with our NLP-driven KG, we aimed to achieve more nuanced, context-aware retrieval, potentially uncovering relevant information several hops away from initial query nodes.

2.3. Evaluation

The evaluation of the KarGus system focuses on assessing the performance of our MD-QA retriever [Kamalloo et al., 2023]. To achieve this, we utilize a robust methodology that involves generating question-ground truth node pairs using a pre-trained LLM. This setup allows us to objectively measure the effectiveness of our retriever models in returning contextually relevant documents.

2.3.1. Generation of Evaluation Data:

Using the capabilities of the OpenAI GPT-3.5 Turbo LLM coupled with LlamaIndex, we generate diverse pairs of questions and their corresponding ground truth nodes. These pairs serve as a benchmark to test the performance of our retrieval methods. The generation process ensures that each question is associated with the most contextually appropriate segment within our corpus of documents, hence establishing a clear target for evaluation purposes.

2.3.2. Metrics for Comparison:

During the NLP Feature Analysis experimentation 2.4.3, to compare the performance of various retrievers, we employ the MLflow library, which provides a suite of evaluation metrics. Each retriever is tested against the generated dataset, and the following metrics are calculated for comprehensive assessment:

- **Precision at k (P@k):** Measures the proportion of retrieved documents among the top k that is the correct ground truth node, indicating the accuracy of the retriever in retrieving relevant documents.
- **Recall at k (R@k):** Indicates whether the ground truth node appears within the top k retrieved documents, essentially checking the retriever's success in finding the relevant document.
- **Normalized Discounted Cumulative Gain at k (NDCG@k):** Evaluates the ranking quality by giving higher scores to cases where the ground truth node appears higher in the top k results, rewarding effective ranking.

These metrics offer a comprehensive evaluation of retriever performance, encompassing both the accuracy and ranking quality of the retrieved results.

2.3.3. Comparative Analysis:

For the Performance Comparison tests 2.4.5, we selected recall as our primary evaluation metric due to the nature of our retrieval task: each query has exactly one relevant document. Recall at k (where k is 5 or 10) directly measures our system's ability to find this single relevant document within the top k results, providing a clear and interpretable measure of retrieval effectiveness.

Our analysis encompasses three key dimensions:

- **NLP Feature Impact:** We tested the individual and collective impact of different NLP features and heuristic weights on retrieval effectiveness. We compared our NLP-driven retrievers to a baseline embedding index method using Facebook AI Similarity Search (FAISS) with OpenAI's Text Encoder ”text-embedding-ada-002” encoder.
- **Heuristic Performance:** We evaluated the performance of the top-ranked nodes, as determined by our initial heuristic scores, against the baseline.

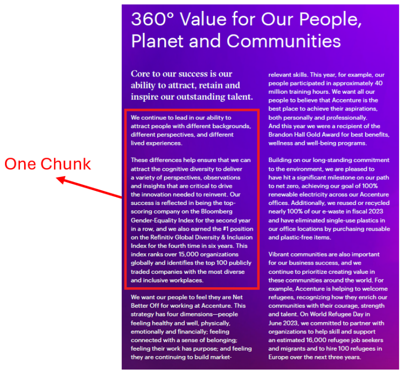
- **GNN Performance:** We assessed the performance of our trained Graph Neural Network models (GraphSAGE and GCN) against both the baseline and our heuristic-based approach, providing insights into the effectiveness of graph-based learning in our retrieval task.

This multi-faceted approach allows us to comprehensively evaluate our system's performance, comparing different components of our method against established baselines and each other. It provides a thorough assessment of the strengths and limitations of our NLP-driven and graph-based retrieval methods in the context of our specific MD-QA task.

2.4. Results

This section describes the experimental setup used to validate the effectiveness of the KarGus system in the corporate intelligence domain. It encompasses the data used, the analysis of NLP features, and performance comparisons of different system components.

2.4.1. Data Collection



One Chunk

360° Value for Our People, Planet and Communities

Core to our success is our ability to attract, retain and inspire our outstanding talent.

We continue to lead in our ability to attract people with different backgrounds, different perspectives, and different lived experiences.

These differences help ensure that we can attract the cognitive diversity to deliver a variety of perspectives, observations and insights that are critical to drive the innovation needed to reinvent. Our success is reflected in being the top-scoring company on the Bloomberg Gender-Equality Index for the second year in a row, and we also earned the #1 position on the Refinitiv Global Diversity & Inclusion Index for the fourth time in six years. This index ranks over 15,000 organizations globally and identifies the top 100 publicly traded companies with the most diverse and inclusive workplaces.

We want our people to feel they are Net Better Off for working at Accenture. This strategy has four dimensions—people feeling healthy and well, physically, emotionally and financially; feeling connected with a sense of belonging; feeling their work has purpose; and feeling they are continuing to build market-relevant skills. This year, for example, our people participated in approximately 40 million training hours. We want all our people to believe that Accenture is the best place to achieve their aspirations, both personally and professionally. And this year we were a recipient of the Brandon Hall Gold Award for best benefits, wellness and well-being programs.

Building on our long-standing commitment to the environment, we are pleased to have hit a significant milestone on our path to net zero, achieving our goal of 100% renewable electricity across our Accenture offices. Additionally, we reused or recycled nearly 100% of our e-waste in fiscal 2023 and have eliminated single-use plastics in our office locations by purchasing reusable and plastic-free items.

Vibrant communities are also important for our business success, and we continue to prioritize creating value in these communities around the world. For example, Accenture is helping to welcome refugees, recognizing how they enrich our communities with their courage, strength and talent. On World Refugee Day in June 2023, we committed to partner with organizations to help skill and support an estimated 16,000 refugee job seekers and migrants and to hire 100 refugees in Europe over the next three years.

Figure 2.6: Example of a chunk

The dataset comprises a selection of corporate documents, including annual reports and project reports from Accenture and its competitors. The dataset totals 30 documents and 1810 pages, segmented into 10853 text chunks[2] using the Langchain tool with a chunk size of 512 tokens and an overlap of 64 tokens. An example of a chunk split is shown in the Figure 2.6. For the construction of the KG, we chose a k-value of 5 for the KNN algorithm as described in section 2.2.2.

Each experiment was conducted using 200 questions, with each question's answer residing in a randomly selected chunk from the entire corpus. This approach ensures a diverse and comprehensive evaluation of the system's retrieval capabilities.

2.4.2. Knowledge Graph Analysis

The analysis of the 30 subgraphs revealed a structure balancing local clustering with global connectivity. The KG demonstrated an average modularity score of 0.4372, indicating moderately well-defined community structures. Most documents exhibited a moderate number of edges and nodes, with two notable outliers (Figures 3.7a, 3.7b). The community size distribution skewed towards small to moderate-sized clusters, with modularity values ranging widely across subgraphs (Figures 3.8a, 3.8b).

The graph's connectivity is characterized by an average clustering coefficient of 0.28 and a mean transitivity of 0.23 (Figures 3.9a, 3.10a). These metrics, combined with relatively short average path lengths, suggest a network structure conducive to efficient information traversal. A slight positive assortativity (mean 0.07, Figure 3.10b) indicates a tendency for nodes to connect with others of similar degree, potentially enhancing network resilience.

The degree distribution approximated a normal pattern with a peak around 20, notably featuring some highly connected hub nodes (Figure 3.11a). Analysis of centrality measures – degree, PageRank, and betweenness – further highlighted the presence of influential nodes within the graph structure (Figures 3.12a, 3.12b, 3.12c). For instance, the degree centrality distribution revealed a subset of nodes with significantly higher connectivity, crucial for information flow. These characteristics collectively describe a graph structure that balances local clustering with global connectivity, potentially supporting efficient information retrieval operations.

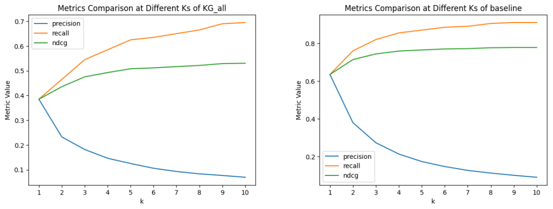
2.4.3. NLP Features Analysis

The impact of different NLP feature combinations (TF-IDF keywords, NER entities, and semantic similarity) and heuristic weights on retrieval performance were analyzed. This analysis involved the following steps:

[2] 1 token ≈ 0.75 words

Feature Combination Impact

We evaluated the performance of individual NLP features and their combinations, comparing them against a baseline using balanced heuristic weights (1/3 each for semantic, keyword, and entity features). Figure 2.8a illustrates the full KG performance, while Figure 2.8b shows the baseline performance.



(a) Full KG Performance

(b) Baseline Performance

Figure 2.7: Comparison of Full KG Balanced Heuristic Weights and Baseline Performance

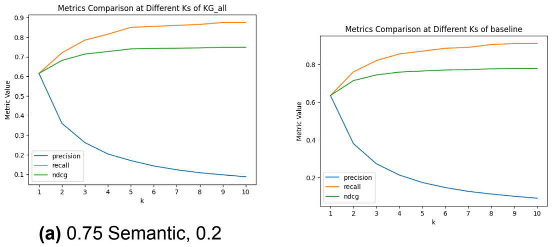
The key findings of this experiments include:

- The baseline outperformed our KG's initial starting nodes, as expected.
- Semantic similarity emerged as the most crucial component, with its performance matching the baseline when used alone. Higher k values improved recall and NDCG but decreased precision.
- Entity-based retrieval performed significantly lower, likely due to the specificity of entity information. However, it showed potential when combined with other features.
- Keyword-based retrieval, while slightly lower performing than semantic similarity, demonstrated good potential. The combination of semantic and keyword features notably improved retrieval performance.
- As k increased, recall and NDCG improved for keyword-based retrieval, highlighting the importance of keyword relevance in capturing contextual appropriateness.

The balanced heuristic weights revealed that while semantic similarity was the most important feature, keywords showed good potential, and entities added value in combination with others. These insights guided our subsequent phase of optimizing heuristic weights to further improve retrieval performance.

Heuristic Weights

We systematically tested various heuristic weight configurations to optimize the performance of our KG retrieval system. Figure 2.8a shows the performance of the optimal configuration, while Figure 2.8b presents the baseline performance for comparison.



(a) 0.75 Semantic, 0.2 Keyword, 0.05 Entity

(b) Baseline Performance

Figure 2.8: Comparison KG Optimal Heuristic Weights and Baseline Performance

Here follow the major outcomes of this analysis:

- Increasing the weight of the semantic heuristic markedly improved performance, capturing crucial contextual and conceptual relationships between nodes.
- Keyword relevance proved complementary to semantic features. Configurations with keyword weights of 0.2-0.3 improved both precision and recall.
- Entity-based retrieval consistently showed lower performance. A small entity weight (0.05) was found to be optimal, likely due to the specific and sparse nature of entity information.
- The optimal configuration (0.75 semantic, 0.2 keyword, 0.05 entity) allowed our KG's starting nodes to surpass the baseline in top-k evaluations.

This analysis demonstrates the effectiveness of our weighted approach in balancing different features for improved retrieval performance. Future work could explore more advanced optimization techniques, such as genetic algorithms, to fine-tune these weights further across various datasets and queries.

Following this, we conducted a pathway analysis to examine graph relationships and their impact on identifying ground truth nodes not among the starting nodes, providing additional insights into our KG's structure and retrieval capabilities.

Pathway Analysis

The pathway analysis provides crucial insights into our system's retrieval mechanisms, particularly when the ground truth node is not among the initial results. Figure 2.9 illustrates the distribution of edge types in successful retrievals.

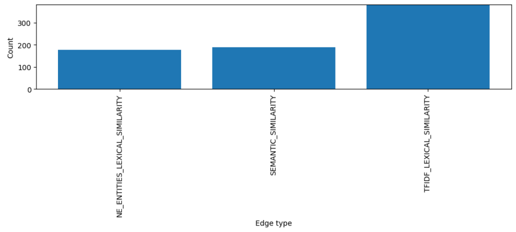


Figure 2.9: Edge Types

The main results of this experimentation are:

- TF-IDF lexical similarity edges were most frequent, followed by the embedding semantic similarity and entities lexical similarity, highlighting the importance of lexical features in retrieval.
- The average shortest path to the ground truth node was 2.04, demonstrating efficient graph traversal.
- Ground truth nodes were present in the starting nodes 87% of the time (Figure 3.14a), indicating high initial retrieval accuracy.
- In 97% of cases, correct starting nodes were in the right document (Figure 3.14b), significantly contributing to system effectiveness.

These results underscore the robustness of our approach, showing that the system excels at identifying relevant nodes either directly or through efficient graph traversal. The combination of semantic and lexical features provides a comprehensive retrieval strategy, with TF-IDF lexical similarities playing a particularly strong role. This analysis offers valuable insights for refining graph construction, traversal heuristics, and feature weighting, demonstrating the system's capability in handling various retrieval scenarios effectively.

2.4.4. Graph Neural Network

This experiment evaluates the performance of two GNN models: GCN and GraphSAGE. The models were tested across various dataset sizes, with and without edge features.

Training

Their performances were evaluated across dataset sizes ranging from 100 to 2,170,600 nodes, with data split into 60% training, 20% validation, and 20% testing. Figure 2.10 illustrates the training results for both models.

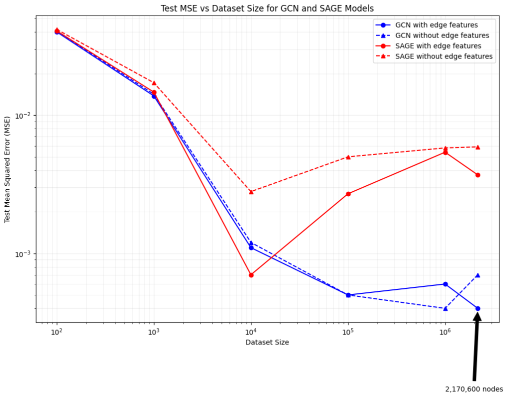


Figure 2.10: SAGE and GCN Training results

Both GCN and GraphSAGE demonstrated improved performance with increasing dataset size, as evidenced by decreasing Mean Squared Error (MSE) values. Notably, GCN's Test MSE decreased from 0.0401 to 0.0004 as the dataset grew, indicating enhanced learning and generalization capabilities with larger datasets.

For datasets below 10,000 nodes, GraphSAGE showed comparable or slightly superior performance to GCN, likely due to its inductive learning capability and flexible aggregation functions. However, as the dataset size exceeded 10,000 nodes, GCN consistently outperformed GraphSAGE. This performance inversion suggests that GCN's spectral graph convolution operation becomes more effective at capturing global graph properties with increased data availability.

The impact of edge features varied between models. GraphSAGE consistently benefited from edge feature inclusion across all dataset sizes, while GCN showed minimal impact on smaller datasets but slight advantages on larger ones. This difference highlights the models' distinct approaches to leveraging structural information in the graph.

These findings underscore the importance of considering dataset size and structural characteristics when choosing between GCN and SAGE for graph-based learning tasks in information retrieval contexts.

Testing

For the testing phase, we used the full dataset with 60% (120 questions) used for training and tested on the remaining 40% (remaining 80 questions). The GNN models were evaluated using recall at k=5 and k=10, which directly measures the system's ability to find the single relevant document within the top k results. The results are summarized in Table 2.3.

Model	Edge F?	R@5	R@10
GraphSAGE	False	**0.022**	**0.037**
GraphSAGE	True	0.004	0.003
GCN	False	0.002	0.001
GCN	True	0.002	0.001

Table 2.3: GNN Model Test Performance (Recall@k), where Edge F? means w/ or w/o the edge features

They indicate that the GNN models, despite showing promising trends during training, struggle to achieve high recall scores in the testing phase. The GraphSAGE model without edge features performed the best among the GNN configurations, aligning with our initial hypothesis that GraphSAGE would be more effective for our task. However, the overall performance is significantly lower than expected, suggesting that further optimization and refinement of the GNN approach is necessary for effective retrieval in our MD-QA context.

2.4.5. Performance Comparison

Table 2.4 presents a comprehensive comparison of different retriever configurations, including the baseline, optimized KG-based retrievers, and GNN models. The results reveal several key insights into the performance of our MD-QA system in corporate intelligence settings.

The baseline and semantic-only KG approaches demonstrated identical performance (Recall@5 = 0.823, Recall@10 = 0.865), validating our KG-based semantic similarity implementation. However, NER-only retrieval performed poorly (Recall@5 = 0.105), indicating its insufficiency as a standalone feature for accurate retrieval in this context.

Combinations of features in the KG-based approach generally outperformed individual features, highlighting the benefits of our multi-feature methodology. The optimized KG-based configuration (0.75 Semantic, 0.2 TFIDF, 0.05 NER) achieved the highest Recall@5 (0.850), surpassing the baseline. This result underscores the importance of nuanced feature integration and weight optimization in complex retrieval tasks.

Notably, the GNN models (GraphSAGE and GCN) significantly underperformed compared to both the baseline and the optimized KG-based approaches, with the best GNN configuration (GraphSAGE without edge features) achieving only a Recall@5 of 0.022. This substantial performance gap suggests that while GNNs show potential, they require considerable further development to be competitive in this specific MD-QA task.

KarGus demonstrates superior performance in handling complex queries that require synthesizing information across multiple documents, a common challenge in corporate intelligence. The system's ability to outperform the baseline embedding method (Recall@5 of 0.850 vs 0.823) showcases its effectiveness in capturing long-range dependencies and contextual relationships often missed by traditional approaches.

These findings collectively highlight the effectiveness of our optimized KG-based approach and underscore the challenges in applying GNNs to this specific retrieval task. The results point to promising areas for future research, including further optimization of KG-based methods and investigation into specialized GNN architectures for corporate intelligence applications.

2.5. Discussion

The development and evaluation of KarGus have revealed significant insights into the challenges and opportunities in MD-QA for corporate intelligence. Our analysis, including the recent GNN experiments, has uncovered several key areas that warrant further discussion.

The success of KarGus in outperforming traditional methods, particularly in handling complex corporate queries, represents a significant step forward in MD-QA for specialized domains.

Retriever	Recall@5	Recall@10
Baseline	0.823	0.865
Full KG*	0.635	0.692
KG Semantic only*	0.823	0.865
KG NER only*	0.105	0.132
KG TFIDF only*	0.586	0.675
KG NER and TFIDF*	0.453	0.592
KG NER and Semantic*	0.456	0.540
KG TFIDF and Semantic*	0.725	0.775
0.75 Semantic, 0.2 TFIDF, 0.05 NER	**0.850**	0.870
0.6 Semantic, 0.3 TFIDF, 0.1 NER	0.783	0.835
0.8 Semantic, 0.15 TFIDF, 0.05 NER	0.835	**0.885**
SAGE (without edge features)	0.022	0.037
SAGE (with edge features)	0.004	0.003
GCN (without edge features)	0.002	0.001
GCN (with edge features)	0.002	0.001

Table 2.4: Comparison of Retriever Configurations with * as balanced weights

Our approach demonstrates the potential of integrating advanced NLP techniques with graph-based representations to create more intelligent and context-aware information retrieval systems.

However, our error analysis revealed several critical challenges. The system's performance in entity recognition, particularly for complex organizational structures and novel industry terms, indicates a need for more robust NER in specialized domains. Graph traversal efficiency also emerged as a significant challenge, with an average shortest path of 2.04 edges to reach ground truth nodes not in the initial set. This suggests that while our current approach is reasonably effective, more sophisticated graph construction and navigation techniques could substantially enhance retrieval performance.

The sensitivity of system performance to heuristic weights suggests that a static weighting approach may be insufficient. Dynamic weight adjustment mechanisms could significantly enhance the system's adaptability and robustness across different document types within the corporate domain.

Notably, the performance of our GNN models, particularly GraphSAGE and GCN, fell significantly short of expectations. This outcome provides valuable insights into the challenges of applying graph-based deep learning to MD-QA tasks in corporate intelligence settings and underscores the need for further research in adapting GNN techniques to this specific domain.

2.6. Conclusion

In this paper, we presented KarGus, a novel system for MD-QA in corporate intelligence settings. By integrating advanced NLP techniques with KG construction and GNNs, KarGus demonstrates improvements over traditional embedding-based methods in handling complex, domain-specific queries.

Our experiments show that KarGus achieves a Recall@5 of 0.850, outperforming the baseline (0.823) and showcasing its ability to capture long-range dependencies and contextual relationships. The system's multi-faceted approach, combining semantic similarity, TF-IDF, and named entity recognition, proves effective in creating a comprehensive and nuanced representation of document content.

While KarGus shows promise, our analysis also reveals important areas for future work, including testing cross-domain adaptability, improving graph traversal efficiency, and refining GNN implementations for this specific task. These findings lay the groundwork for future research in advanced MD-QA systems for specialized domains.

Overall, KarGus represents a significant step forward in addressing the challenges of complex information retrieval, demonstrating the potential of graph-based approaches in enhancing the accuracy and contextual relevance of MD-QA systems.

3 Conclusion

3.1. Research Summary

This thesis introduced KarGus, a novel approach to multi-document question answering (MD-QA) designed to address critical challenges in various applications, with a particular focus on corporate intelligence. KarGus represents a significant advancement in information retrieval (IR) and synthesis from complex document sets, combining advanced Natural Language Processing (NLP) techniques, Knowledge Graph (KG) construction, and Graph Neural Networks (GNNs).

The development of KarGus was motivated by the limitations of existing MD-QA methods, particularly their struggle with effectively synthesizing information across multiple documents and handling domain-specific terminology. These challenges are especially critical in corporate intelligence settings, tested in our experimentations, where the ability to quickly and accurately extract relevant information from vast document repositories can significantly impact decision-making processes.

Our research methodology encompassed several key components:

1. **Advanced NLP Integration:** We developed a multi-faceted approach combining semantic similarity, TF-IDF, and Named Entity Recognition (NER) for comprehensive document representation. This integration allowed for a richer understanding of document content and relationships.
2. **Dynamic Knowledge Graph Construction:** We implemented a novel method for dynamically constructing knowledge graphs from document sets. This approach captures both semantic and lexical relationships across documents, creating a rich, interconnected representation of the information space.
3. **Graph Neural Network Implementation:** We explored the use of two GNN architectures - Graph Sample and Aggregate (GraphSAGE) and Graph Convolutional Networks (GCN) - for graph-based reasoning. These models were trained to learn from the structural information encoded in our KGs, along with NLP-derived features.
4. **Comparative Performance Analysis:** We conducted extensive experiments to compare KarGus against traditional embedding-based methods. Our evaluation metrics included Recall@k, with a particular focus on Recall@5 and Recall@10.
5. **Feature and Weight Optimization:** Through rigorous testing, we determined optimal configurations for NLP feature combinations and their respective weights, fine-tuning KarGus for peak performance.
6. **Graph Structure Analysis:** We performed in-depth analysis of the resulting knowledge

graph structures, examining properties such as modularity, clustering coefficient, and transitivity to understand the efficiency of information traversal.

Key findings from our research include:

- KarGus outperformed the baseline embedding method, achieving a Recall@5 of 0.850 compared to the baseline's 0.823.
- The optimal configuration emphasized semantic similarity (weight 0.75), keyword relevance (0.2), and entity information (0.05).
- GNN models showed promising results in training but underperformed in the retrieval task, highlighting the challenges in applying GNNs to MD-QA.
- Analysis of the KG structure revealed moderately well-defined community structures and efficient information traversal properties.

Our approach integrates semantic analysis, entity recognition, and graph-based representation to create a more robust and context-aware retrieval system. By constructing a KG that captures both the semantic and structural relationships within and between documents, KarGus provides a more comprehensive and nuanced understanding of the information landscape, particularly in complex domains like corporate intelligence.

This research contributes significantly to the field of IR and MD-QA, offering new insights into the integration of NLP techniques with graph-based approaches. It lays the groundwork for future advancements in context-aware, multi-document information synthesis and retrieval systems.

3.2. Research Questions

Our research was guided by three primary questions:
RQ1: How can the unique combination of advanced NLP techniques in KarGus enhance MD-QA performance compared to traditional single-feature approaches?
RQ2: What is the impact of different NLP feature combinations, heuristic weights, and relationship types on the system's retrieval performance?
RQ3: How does KarGus's retrieval performance, leveraging its innovative KG and GNN approach, compare to traditional embedding-based methods in diverse, specialized domains?

In addition to these primary research questions, we also explored the potential of GNNs in enhancing our system's performance. However, our experiments with GNN models, specifically GraphSAGE and GCN, yielded suboptimal results in the retrieval task despite showing promise during training. Due to this underperformance, we decided to treat the GNN component as an exploratory study outside the scope of our main research questions. This decision allowed us to maintain focus on the core strengths of KarGus while still providing valuable insights into the challenges of applying GNNs to MD-QA tasks in corporate intelligence settings. The GNN exploration, while not central to our main findings, offers important directions for future research in integrating graph-based deep learning with knowledge graph-powered question answering systems.

3.3. Discussion of the results

3.3.1. Integration of NLP Techniques

Our research demonstrated that the integration of semantic similarity, TF-IDF, and NER features in KG construction enhances the system's ability to capture complex relationships within documents. This multi-faceted approach allowed for a more nuanced representation of document content, enabling more accurate and contextually relevant retrievals. Specifically, our experiments showed that:

- Semantic similarity emerged as the most crucial component, with its performance matching the baseline when used alone. The semantic-only approach achieved a Recall@5 of 0.823 and a Recall@10 of 0.865, identical to the baseline performance.

- TF-IDF based retrieval, while slightly lower performing than semantic similarity, demonstrated good potential with a Recall@5 of 0.586 and a Recall@10 of 0.675.
- Entity-based retrieval performed significantly lower (Recall@5 of 0.105, Recall@10 of 0.132), likely due to the specificity of entity information. However, it showed potential when combined with other features.

The combination of these NLP techniques in graph construction proved particularly effective in capturing both the broad semantic context and specific entity relationships, contributing to a more comprehensive understanding of document content.

3.3.2. Impact of Feature Combinations and Weights

Our experiments revealed that different combinations of NLP features and their associated weights have a substantial impact on retrieval performance. Key findings include:

- The optimal configuration, emphasizing semantic similarity (weight 0.75) while incorporating keyword relevance (weight 0.2) and entity information (weight 0.05), achieved the highest performance with a Recall@5 of 0.850 and a Recall@10 of 0.870.
- This optimal configuration outperformed both the baseline and individual feature approaches, demonstrating the value of our multi-feature methodology.
- The combination of semantic and keyword features notably improved retrieval performance, achieving a Recall@5 of 0.725 and a Recall@10 of 0.775 when weighted equally.

These results underscore the importance of balancing different NLP features to capture various aspects of document content and query relevance. The dominance of semantic similarity in the optimal configuration highlights its crucial role in understanding the contextual meaning of both queries and documents, while the inclusion of keyword and entity information provides important complementary signals.

3.3.3. Comparison with Traditional Methods

The comparison between KarGus and traditional embedding-based methods revealed significant insights:

- **Overall Performance:** KarGus demonstrated superior performance in handling queries from corporate documents, achieving a Recall@5 of 0.850 and Recall@10 of 0.870, outperforming the baseline's 0.823 and 0.865 respectively.
- **Information Synthesis:** The improved recall suggests KarGus is more effective at identifying and synthesizing relevant information across document chunks, indicating potential for handling complex, multi-document queries typical in corporate intelligence settings.
- **Contextual Understanding:** The graph-based approach of KarGus showed a strong ability to capture contextual relationships, leading to more accurate retrievals in our corporate document dataset.
- **Domain Applicability:** While primarily tested on corporate intelligence documents, KarGus's performance suggests promising adaptability to specialized domains where nuanced understanding of terminology and concepts is crucial.
- **Query Complexity Handling:** The marked improvement in recall indicates KarGus's potential for effectively handling multi-concept queries and those requiring deeper information synthesis, areas where traditional methods often face challenges.

These results validate the potential of our KG-based approach in enhancing MD-QA performance, particularly in complex document environments like corporate intelligence. KarGus's ability to outperform traditional methods in identifying relevant information positions it as a powerful tool for advanced IR tasks in specialized domains.

3.3.4. Graph Neural Network Performance

Our experiments with GNNs in the context of MD-QA revealed both promising aspects and significant challenges:

- **Training Performance:** Both GraphSAGE and GCN models showed encouraging trends during training, with decreasing Mean Squared Error as dataset size increased.
- **Model Comparison:** GraphSAGE performed better on smaller datasets, while GCN excelled with larger datasets (>10,000 nodes), suggesting scale-dependent architecture choices.
- **Retrieval Task Challenges:** Despite promising training results, GNN performance in the retrieval task fell short of expectations. The best configuration (GraphSAGE without edge features) achieved only a Recall@5 of 0.022 and a Recall@10 of 0.037.
- **Performance Gap:** The substantial difference between training performance and retrieval task results highlights the complexity of applying GNNs to practical MD-QA applications.

These findings underscore both the potential and challenges of GNNs in MD-QA, pointing to avenues for future research in architecture optimization and better alignment of GNN learning with retrieval tasks in corporate environments.

3.4. Future Work

Our research findings point to several key areas for future development of KarGus. Implementing a Genetic Algorithm for heuristic optimization could enhance our feature weighting system, allowing for dynamic adjustment based on document and query characteristics. Refining the NER model's fine-tuning process and improving entity integration could boost the system's ability to handle domain-specific terminology.

Further investigation into optimal parameter values for NLP integration, such as alpha and beta, could lead to performance improvements. Given the challenges with GNN performance, exploring alternative training approaches, including the use of node embeddings as features, is crucial. Testing the system's performance on tabular data could expand its applicability to diverse data formats.

Assessing KarGus's scalability and adaptability across various corpus sizes and domains is essential for its practical implementation. Integrating KarGus into a RAG environment would provide insights into its real-world performance and potential for enhancing LLM outputs.

Exploring cross-lingual capabilities and incorporating temporal dynamics into the KG could significantly broaden the system's applicability. Lastly, developing explainable AI techniques to elucidate retrieval decisions would enhance trust and usability in corporate settings.

These areas of future work aim to address current limitations and expand KarGus's capabilities, potentially leading to more sophisticated and versatile MD-QA systems for corporate intelligence applications.

3.5. Answering the Research Questions

Based on our findings, we can now provide comprehensive answers to our research questions:

RQ1: Advanced NLP techniques can be effectively integrated into a KG-based retrieval system through a multi-faceted approach that combines semantic similarity, TF-IDF, and NER. This integration allows for a richer representation of document content and relationships, enabling more accurate and contextually relevant retrievals. Our experiments showed that semantic similarity emerged as the most crucial component, matching the baseline performance when used alone (Recall@5 of 0.823, Recall@10 of 0.865). TF-IDF based retrieval demonstrated good potential (Recall@5 of 0.586, Recall@10 of 0.675), while entity-based retrieval, although lower performing individually (Recall@5 of 0.105, Recall@10 of 0.132), showed value when combined with other features. This multi-feature approach proved particularly effective in capturing both broad semantic context and specific entity relationships, contributing to a more comprehensive understanding of document content.

RQ2: Different NLP feature combinations and heuristic weights have a substantial impact on retrieval performance. Our research found that a configuration emphasizing semantic similarity (75%) while incorporating keyword relevance (20%) and entity information (5%) yielded the best results, outperforming both individual features and the baseline approach. This optimal configuration achieved a Recall@5 of 0.850 and a Recall@10 of 0.870, improving upon the baseline and individual feature performances. The combination of semantic and keyword features notably enhanced retrieval performance, achieving a Recall@5 of 0.725 and a Recall@10 of 0.775 when weighted equally. These results underscore the importance of balancing different NLP features to capture various aspects of document content and query relevance, with semantic similarity playing a crucial role in understanding contextual meaning, complemented by keyword and entity information.

RQ3: KarGus demonstrates improved retrieval performance compared to traditional embedding-based methods, particularly in handling complex queries that require synthesizing information from multiple documents. The optimized KG-based approach achieved a Recall@5 of 0.850 and a Recall@10 of 0.870, outperforming the baseline's 0.823 and 0.865 respectively. This improvement is attributed to the system's ability to capture long-range dependencies and contextual relationships through its graph-based approach. KarGus shows great potential in scenarios where relevant information is distributed across multiple documents or requires understanding complex inter-document relationships, a common challenge in corporate intelligence contexts. However, the integration of GNNs, while showing promise during training, did not translate to improved retrieval performance. The best GNN configuration (GraphSAGE without edge features) achieved only a Recall@5 of 0.022, highlighting the need for further research to bridge the gap between GNN learning and effective IR in practical MD-QA applications.

3.6. Final Conclusion

KarGus presents a novel approach to MD-QA for various applications, demonstrating the potential of integrating advanced NLP techniques with graph-based approaches. The system's ability to outperform traditional methods, particularly in handling complex queries and capturing long-range dependencies, showcases the promise of this approach.

While challenges remain, particularly in cross-domain adaptation and the effective application of GNNs to MD-QA tasks, KarGus provides a solid foundation for future research and development in this critical area of IR and knowledge management. The insights gained from this research contribute to the broader field of NLP and IR, offering new perspectives on how to effectively combine semantic analysis, graph-based representations, and neural network techniques for enhanced question answering.

As organizations continue to grapple with increasing volumes of unstructured data, systems like KarGus pave the way for more intelligent, context-aware, and adaptive IR solutions. The continued development and refinement of such systems will play a crucial role in unlocking the full potential of organizational knowledge, enabling more informed decision-making, and driving innovation in data-driven industries.

Moving forward, the focus should be on addressing the challenges identified in this research, particularly in improving GNN performance for MD-QA tasks and further validating KarGus's adaptability across diverse document types and domains. While KarGus has demonstrated strong performance and inherent adaptability in corporate intelligence settings, expanding our experiments to a wider range of specialized domains would provide valuable insights into the system's full potential.

The development of sophisticated MD-QA systems continues to evolve, and this research contributes valuable insights to the advancing field of information management and artificial intelligence. KarGus offers a promising foundation for future explorations, with its demonstrated adaptability and performance contributing to the ongoing progress in this area of study.

Bibliography

[Arbaaeen and Shah, 2021] Arbaaeen, A. and Shah, A. (2021). Ontology-based approach to semantically enhanced question answering for closed domain: A review. *Information*, 12(5):200.

[Arseniev-Koehler, 2021] Arseniev-Koehler, A. (2021). Theoretical foundations and limits of word embeddings: what types of meaning can they capture?

[Bojanowski et al., 2017] Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 1–7.

[Borgeaud et al., 2021] Borgeaud, S., Mensch, A., Hoffmann, J., Cai, T., Rutherford, E., Millican, K., van den Driessche, G., Lespiau, J., Damoc, B., Clark, A., de Las Casas, D., Guy, A., Menick, J., Ring, R., Hennigan, T., Huang, S., Maggiore, L., Jones, C., Cassirer, A., Brock, A., Paganini, M., Irving, G., Vinyals, O., Osindero, S., Simonyan, K., Rae, J. W., Elsen, E., and Sifre, L. (2021). Improving language models by retrieving from trillions of tokens. *CoRR*, abs/2112.04426.

[Bowman et al., 2015] Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642. ACL.

[Brown et al., 2020] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. *CoRR*, abs/2005.14165.

[Chen and Zeng, 2013] Chen, J. and Zeng, D. (2013). A survey on question answering system. *Advances in Artificial Intelligence*, 2013:1–6.

[Chowdhery et al., 2022] Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvyashchenko, S., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N., Prabhakaran, V., Reif, E., Du, N., Hutchinson, B., Pope, R., Bradbury, J., Austin, J., Isard, M., Gur-Ari, G., Yin, P., Duke, T., Levskaya, A., Ghemawat, S., Dev, S., Michalewski, H., Garcia, X., Misra, V., Robinson, K., Fedus, L., Zhou, D., Ippolito, D., Luan, D., Lim, H., Zoph, B., Spiridonov, A., Sepassi, R., Dohan, D., Agrawal, S., Omernick, M., Dai, A. M., Pillai, T. S., Pellat, M., Lewkowycz, A., Moreira, E., Child, R., Polozov, O., Lee, K., Zhou, Z., Wang, X., Saeta, B., Diaz, M., Firat, O., Catasta, M., Wei, J., Meier-Hellstern, K., Eck, D., Dean, J., Petrov, S., and Fiedel, N. (2022). Palm: Scaling language modeling with pathways.

[Devlin et al., 2018] Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

[Diefenbach et al., 2018] Diefenbach, D., Lopez, V., Singh, K., and Maret, P. (2018). Core techniques of question answering systems over knowledge bases: a survey. *Core Techniques of Question Answering Systems over Knowledge Bases: a Survey*, 55.

[Guo et al., 2020] Guo, Q., Zhuang, F., Qin, C., Zhu, H., Xie, X., Xiong, H., and He, Q. (2020). A survey on knowledge graph-based recommender systems. *CoRR*, abs/2003.00911.

[Gupta, 2011] Gupta, A. K. (2011). Question answering system: A survey. *International Journal of Computer Applications*, 34(7):1–5.

[Gururangan et al., 2020] Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., and Smith, N. A. (2020). Don't stop pretraining: Adapt language models to domains and tasks. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

[Guu et al., 2020] Guu, K., Lee, K., Tung, Z., Pasupat, P., and Chang, M. (2020). REALM: retrieval-augmented language model pre-training. *CoRR*, abs/2002.08909.

[Hamilton et al., 2017a] Hamilton, W., Ying, Z., and Leskovec, J. (2017a). Inductive representation learning on large graphs. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

[Hamilton et al., 2017b] Hamilton, W. L., Ying, R., and Leskovec, J. (2017b). Inductive representation learning on large graphs. *CoRR*, abs/1706.02216.

[Izacard et al., 2022] Izacard, G., Lewis, P., Lomeli, M., Hosseini, L., Petroni, F., Schick, T., Dwivedi-Yu, J., Joulin, A., Riedel, S., and Grave, E. (2022). Atlas: Few-shot learning with retrieval augmented language models.

[Ji et al., 2022] Ji, S., Pan, S., Cambria, E., Marttinen, P., and Yu, P. S. (2022). A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2):494–514.

[Kaddour et al., 2023] Kaddour, J., Harris, J., Mozes, M., Bradley, H., Raileanu, R., and McHardy, R. (2023). Challenges and applications of large language models.

[Kamalloo et al., 2023] Kamalloo, E., Clarke, C. L. A., and Rafiei, D. (2023). Limitations of open-domain question answering benchmarks for document-level reasoning. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 2123–2128, New York, NY, USA. Association for Computing Machinery.

[Kang et al., 2023] Kang, M., Kwak, J. M., Baek, J., and Hwang, S. J. (2023). Knowledge-consistent dialogue generation with language models and knowledge graphs.

[Karpukhin et al., 2020] Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and Yih, W.-t. (2020). Dense passage retrieval for open-domain question answering. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

[Khattab et al., 2023] Khattab, O., Santhanam, K., Li, X. L., Hall, D., Liang, P., Potts, C., and Zaharia, M. (2023). Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive nlp.

[Kingma and Ba, 2015] Kingma, D. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA.

[Kipf and Welling, 2016] Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907.

[Kipf and Welling, 2017] Kipf, T. N. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks.

[Kulkarni et al., 2024] Kulkarni, A., Dery, L., Setlur, A., Raghunathan, A., Talwalkar, A., and Neubig, G. (2024). Multitask learning can improve worst-group outcomes.

[Lample et al., 2016] Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named entity recognition. In Knight, K., Nenkova, A., and Rambow, O., editors, *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

[Lazaridou et al., 2022] Lazaridou, A., Gribovskaya, E., Stokowiec, W., and Grigorev, N. (2022). Internet-augmented language models through few-shot prompting for open-domain question answering.

[Lewis et al., 2020a] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020a). Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2005.11401*.

[Lewis et al., 2020b] Lewis, P. S. H., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., and Kiela, D. (2020b). Retrieval-augmented generation for knowledge-intensive NLP tasks. *CoRR*, abs/2005.11401.

[Lin et al., 2021] Lin, Y., Liu, Z., and Sun, M. (2021). Graph-based natural language understanding: A survey. *arXiv preprint arXiv:2103.13113*.

[Liu et al., 2022] Liu, W., Pang, J., Li, N., Yue, F., and Liu, G. (2022). Few-shot short-text classification with language representations and centroid similarity. *Applied Intelligence*, 53(7):8061–8072.

[Liu et al., 2018] Liu, X., Hu, Y., and Wang, J. (2018). A hybrid question answering system. In *Proceedings of the International Conference on Computer Science and Artificial Intelligence*, pages 1234–1238.

[Liu et al., 2020] Liu, Z., Wang, Y., and Li, J. (2020). A knowledge graph embedding-based question answering system. In *Proceedings of the International Conference on Data Mining and Knowledge Discovery*, pages 567–571.

[Lu et al., 2019] Lu, X., Pramanik, S., Saha Roy, R., Abujabal, A., Wang, Y., and Weikum, G. (2019). Answering complex questions by joining multi-document evidence with quasi knowledge graphs. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '19. ACM.

[Marwan Omar, 2022] Marwan Omar, Soohyeon Choi, D. N. D. M. (2022). Robust natural language processing: Recent advances, challenges, and future directions. *arXiv preprint arXiv:2201.00768*.

[Maynez et al., 2020] Maynez, J., Narayan, S., Bohnet, B., and McDonald, R. (2020). On faithfulness and factuality in abstractive summarization. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

[Mikolov et al., 2013] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *Proceedings of Workshop at ICLR*, 2013.

[Min et al., 2019] Min, S., Chen, D., Zettlemoyer, L., and Hajishirzi, H. (2019). Knowledge guided text retrieval and reading for open domain question answering. *CoRR*, abs/1911.03868.

[Nadeau and Sekine, 2007] Nadeau, D. and Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30.

[NASTASE et al., 2015] NASTASE, V., MIHALCEA, R., and RADEV, D. R. (2015). A survey of graphs in natural language processing. *Natural Language Engineering*, 21(5):665–698.

[Noy et al., 2019] Noy, N., Gao, Y., Jain, A., Narayanan, A., Patterson, A., and Taylor, J. (2019). Industry-scale knowledge graphs: lessons and challenges. *Commun. ACM*, 62(8):36–43.

[Opdahl and Nunavath, 2020] Opdahl, A. L. and Nunavath, V. (2020). Big data. *CoRR*, abs/2012.09109.

[Peterson, 2009] Peterson, L. E. (2009). K-nearest neighbor. *Scholarpedia*, 4(2):1883. revision #137311.

[Probierz et al., 2023] Probierz, B., Hrabia, A., and Kozak, J. (2023). A new method for graph-based representation of text in natural language processing. *Electronics*, 12(13).

[Raffel et al., 2019] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

[Rajpurkar et al., 2016] Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250.

[Reimers and Gurevych, 2019] Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

[Sammut and Webb, 2010] Sammut, C. and Webb, G. I., editors (2010). *TF–IDF*, pages 986–987. Springer US, Boston, MA.

[Saxena et al., 2020] Saxena, A., Tripathi, A., and Talukdar, P. (2020). Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4498–4507, Online. Association for Computational Linguistics.

[Schneider et al., 2022] Schneider, P., Schopf, T., Vladika, J., Galkin, M., Simperl, E., and Matthes, F. (2022). A decade of knowledge graphs in natural language processing: A survey.

[Seonwoo et al., 2022] Seonwoo, Y., Yoon, S., Dernoncourt, F., Bui, T., and Oh, A. (2022). Virtual knowledge graph construction for zero-shot domain-specific document retrieval. In Calzolari, N., Huang, C.-R., Kim, H., Pustejovsky, J., Wanner, L., Choi, K.-S., Ryu, P.-M., Chen, H.-H., Donatelli, L., Ji, H., Kurohashi, S., Paggio, P., Xue, N., Kim, S., Hahm, Y., He, Z., Lee, T. K., Santus, E., Bond, F., and Na, S.-H., editors, *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1169–1178, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

[Sparck Jones, 1988] Sparck Jones, K. (1988). *A statistical interpretation of term specificity and its application in retrieval*, page 132–142. Taylor Graham Publishing, GBR.

[Sutskever et al., 2013] Sutskever, I., Martens, J., Dahl, G., and Hinton, G. (2013). On the importance of initialization and momentum in deep learning. In Dasgupta, S. and McAllester, D., editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1139–1147, Atlanta, Georgia, USA. PMLR.

[Team et al., 2024] Team, G., Anil, R., Borgeaud, S., Alayrac, J.-B., Yu, J., and Soricut, R. (2024). Gemini: A family of highly capable multimodal models.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *CoRR*, abs/1706.03762.

[Wang et al., 2017] Wang, Q., Mao, Z., Wang, B., and Guo, L. (2017). Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743.

[Wang et al., 2022] Wang, X., Wang, H., and Yang, D. (2022). Measure and improve robustness in NLP models: A survey. In Carpuat, M., de Marneffe, M.-C., and Meza Ruiz, I. V., editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies, pages 4569–4586, Seattle, United States. Association for Computational Linguistics.

[Wang et al., 2023] Wang, Y., Lipka, N., Rossi, R. A., Siu, A., Zhang, R., and Derr, T. (2023). Knowledge graph prompting for multi-document question answering.

[Wiegreffe and Marasovic, 2021] Wiegreffe, S. and Marasovic, A. (2021). Teach me to explain: A review of datasets for explainable NLP. *CoRR*, abs/2102.12060.

[Wu et al., 2019a] Wu, X., Wu, J., Fu, X., Li, J., Zhou, P., and Jiang, X. (2019a). Automatic knowledge graph construction: A report on the 2019 icdm/icbk contest. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 1540–1545.

[Wu et al., 2019b] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. S. (2019b). A comprehensive survey on graph neural networks. *CoRR*, abs/1901.00596.

[Xu et al., 2018] Xu, K., Hu, W., Leskovec, J., and Jegelka, S. (2018). How powerful are graph neural networks? *CoRR*, abs/1810.00826.

[Xu and Lapata, 2020] Xu, Y. and Lapata, M. (2020). Query focused multi-document summarization with distant supervision.

[Yang et al., 2023] Yang, C., Xiong, G., Zhang, Q., Shi, J., Gou, G., Li, Z., and Liu, C. (2023). Few-shot encrypted traffic classification via multi-task representation enhanced meta-learning. *Computer Networks*, 228:109731.

[Ye et al., 2023] Ye, H., Zhang, N., Chen, H., and Chen, H. (2023). Generative knowledge graph construction: A review.

[Ying et al., 2018] Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., and Leskovec, J. (2018). Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, KDD '18. ACM.

[Zhang et al., 2024a] Zhang, J., Zhang, H., Zhang, D., Liu, Y., and Huang, S. (2024a). End-to-end beam retrieval for multi-hop question answering.

[Zhang et al., 2019] Zhang, M., Chen, S., and Liu, Y. (2019). A context-aware question answering system. In *Proceedings of the International Conference on Natural Language Processing and Information Retrieval*, pages 856–860.

[Zhang and Ge, 2019] Zhang, T. and Ge, S. S. (2019). An improved tf-idf algorithm based on class discriminative strength for text categorization on desensitized data. In *Proceedings of the 2019 3rd International Conference on Innovation in Artificial Intelligence*, ICIAI '19, page 39–44, New York, NY, USA. Association for Computing Machinery.

[Zhang et al., 2024b] Zhang, Z., Fang, M., and Chen, L. (2024b). Retrievalqa: Assessing adaptive retrieval-augmented generation for short-form open-domain question answering.

Appendix

3.7. Methodology

3.7.1. NLP Features Decision

The selection of these specific NLP features was the result of careful consideration and preliminary experimentation. While other advanced NLP techniques, such as topic modeling, were initially considered, they were not included in the final implementation. Our decision to focus on text embeddings, TF-IDF, and NER was based on several key factors:

- **Computational Efficiency:** These methods offer an optimal balance between performance and computational cost, which is crucial for maintaining system scalability, especially when dealing with large document corpora.
- **Interpretability:** The chosen features provide clear, interpretable results, which is beneficial for both system development and potential future expansions. This interpretability aids in understanding the model's decision-making process and facilitates debugging and improvement.
- **Complementary Information:** Text embeddings capture semantic relationships, TF-IDF focuses on term importance, and NER identifies key entities. This combination provides a well-rounded representation of the text without significant overlap, ensuring that different aspects of the text are captured effectively.
- **Proven Effectiveness:** These techniques have demonstrated robust performance across various NLP tasks and domains, as evidenced by numerous studies in the literature [citations].

While more techniques could potentially offer additional insights, our preliminary experiments suggested that the added complexity and computational overhead did not yield significant improvements in retrieval performance for our specific use case. However, the modular nature of our system allows for the integration of additional NLP features in future iterations, should they prove beneficial.

3.7.2. NLP Heuristics

3.7.3. Knowledge Graph Construction

3.7.4. GNN Model Architecture and Training

For both GraphSAGE and GCN models, we used the following architecture and training parameters:

- Hidden layers: 16 dimensions
- Output layer: 1 neuron (for regression task)
- Training epochs: 200
- Optimizer: Adam
- Learning rate: 0.001
- Momentum: 0.9
- Weight decay: 5e-4

These parameters were chosen based on common practices in the field and default values often used in graph neural network implementations:

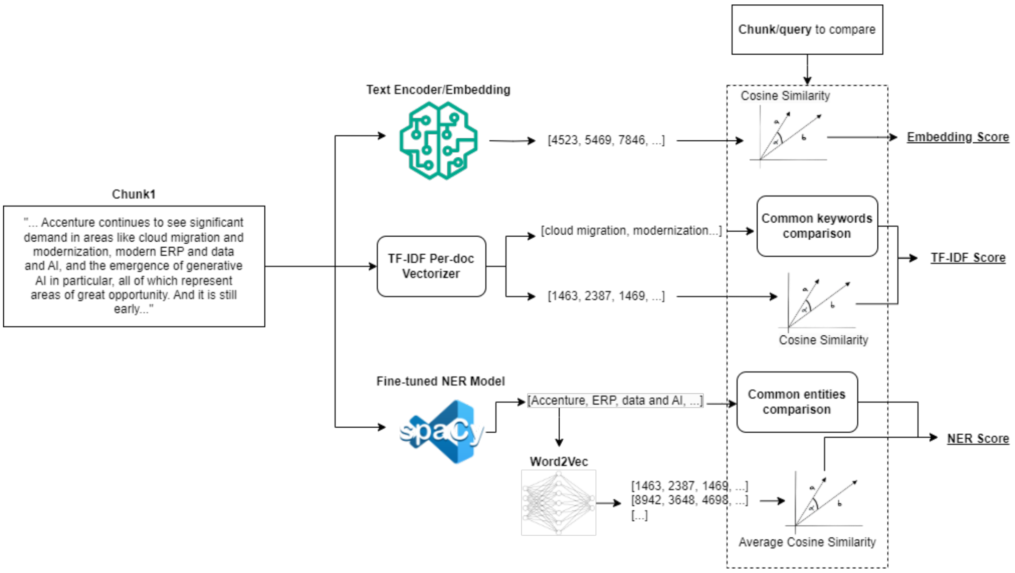

Chunk/query to compare
Text Encoder/Embedding
Cosine Similarity
[4523, 5469, 7846, ...]
Embedding Score
Chunk1
"... Accenture continues to see significant demand in areas like cloud migration and modernization, modern ERP and data and AI, and the emergence of generative AI in particular, all of which represent areas of great opportunity. And it is still early..."
TF-IDF Per-doc Vectorizer
[cloud migration, modernization...]
Common keywords comparison
TF-IDF Score
[1463, 2387, 1469, ...]
Cosine Similarity
Fine-tuned NER Model
[Accenture, ERP, data and AI, ...]
Common entities comparison
NER Score
Word2Vec
[1463, 2387, 1469, ...]
[8942, 3648, 4698, ...]
[...]
Average Cosine Similarity

Figure 3.1: NLP features integration

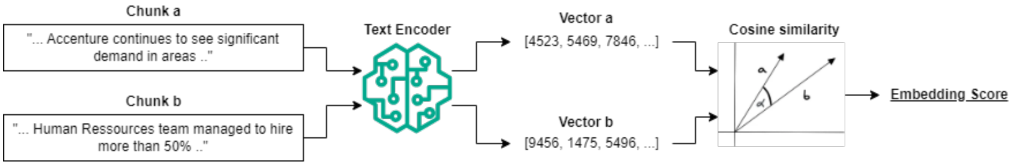

Chunk a
"... Accenture continues to see significant demand in areas .."
Chunk b
"... Human Ressources team managed to hire more than 50% .."
Text Encoder
Vector a
[4523, 5469, 7846, ...]
Vector b
[9456, 1475, 5496, ...]
Cosine similarity
Embedding Score

Figure 3.2: Example chunks embedding score

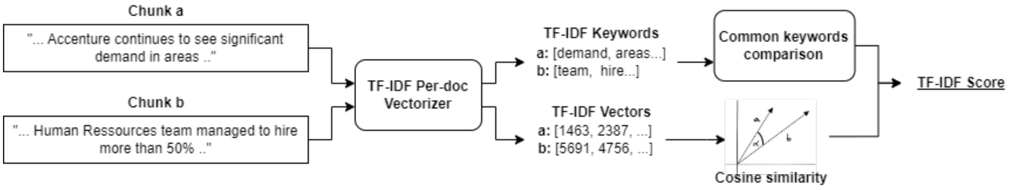

Chunk a
"... Accenture continues to see significant demand in areas .."
Chunk b
"... Human Ressources team managed to hire more than 50% .."
TF-IDF Per-doc Vectorizer
TF-IDF Keywords
a: [demand, areas...]
b: [team, hire...]
Common keywords comparison
TF-IDF Score
TF-IDF Vectors
a: [1463, 2387, ...]
b: [5691, 4756, ...]
Cosine similarity

Figure 3.3: Example chunks tf-idf score

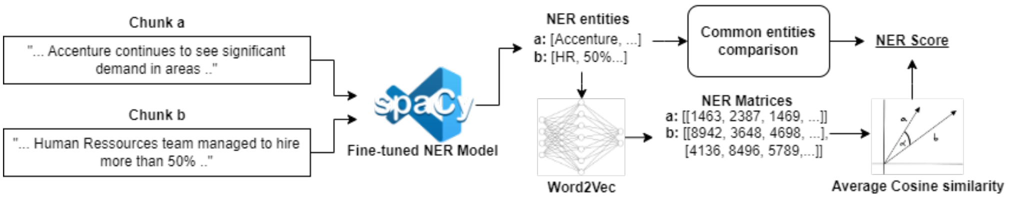

Chunk a
"... Accenture continues to see significant demand in areas .."
Chunk b
"... Human Ressources team managed to hire more than 50% .."
Fine-tuned NER Model
NER entities
a: [Accenture, ...]
b: [HR, 50%...]
Common entities comparison
NER Score
NER Matrices
a: [[1463, 2387, 1469, ...]]
b: [[8942, 3648, 4698, ...],
[4136, 8496, 5789,...]]
Word2Vec
Average Cosine similarity

Figure 3.4: Example chunks NER score

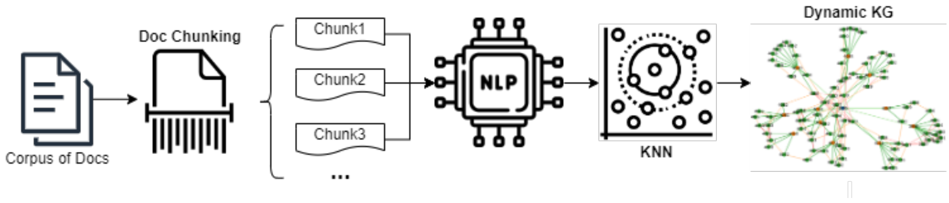


Figure 3.5: KarGus Dynamic Knowledge Graph Construction

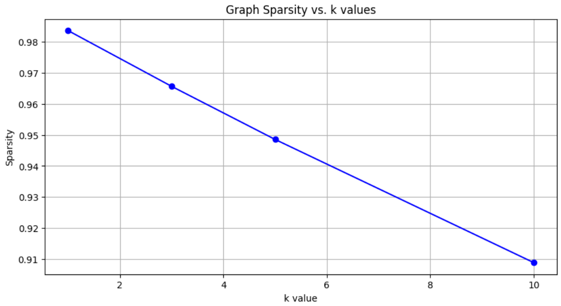


Figure 3.6: Graph Average Sparsity Plot

1. Hidden dimension (16): This dimension was selected as a balance between model complexity and computational efficiency, following similar approaches in graph-based tasks [Kipf and Welling, 2016].
2. Output neuron (1): As our task is a regression problem (predicting relevance scores), a single output neuron is appropriate.
3. Epochs (200): This number was chosen to allow sufficient training time while considering computational constraints.
4. Optimizer (Adam): Adam is widely used due to its adaptive learning rate properties and good performance across a variety of tasks [Kingma and Ba, 2015].
5. Learning rate (0.001): This is a common default value for the Adam optimizer, providing a good balance between convergence speed and stability.
6. Momentum (0.9): This value helps accelerate stochastic gradient descent in the relevant direction and dampens oscillations, and is a standard value used in many deep learning applications [Sutskever et al., 2013].
7. Weight decay (5e-4): This value was chosen to provide regularization and prevent overfitting, based on common practices in graph neural network literature [Hamilton et al., 2017a].

It's important to note that these parameters were not optimized through extensive empirical testing for our specific task. Future work could involve a more thorough exploration of the parameter space to potentially improve model performance.

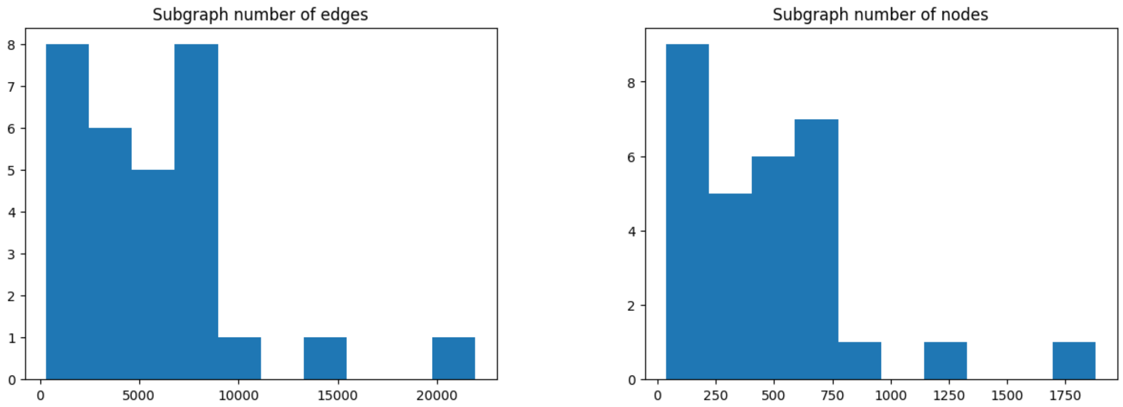
3.8. Results

3.8.1. Knowledge Graph Analysis

3.8.2. NLP Features Analysis

3.8.3. Pathway Analysis

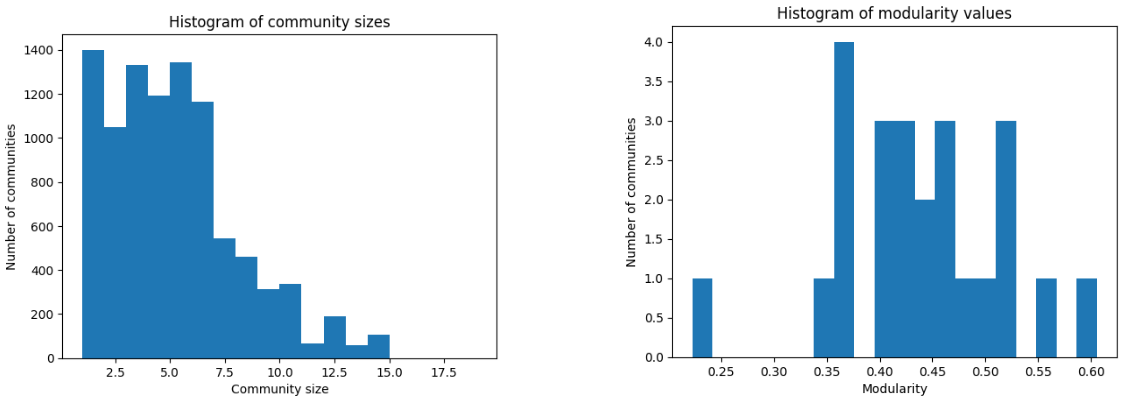
3.8.4. Heuristics Weights



(a) Average Number of Edges

(b) Average Number of Nodes

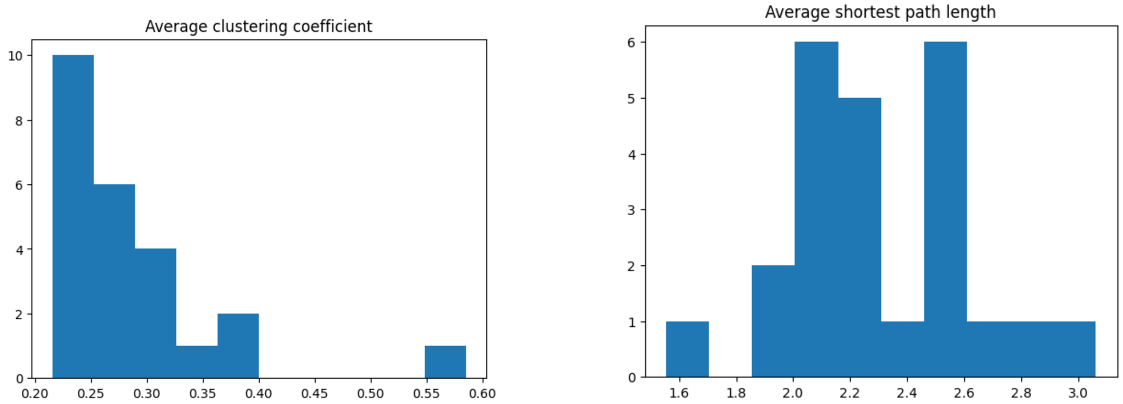
Figure 3.7: Average Number of Edges and Nodes in the Subgraphs



(a) Histogram of Community Sizes

(b) Histogram of Modularity Values

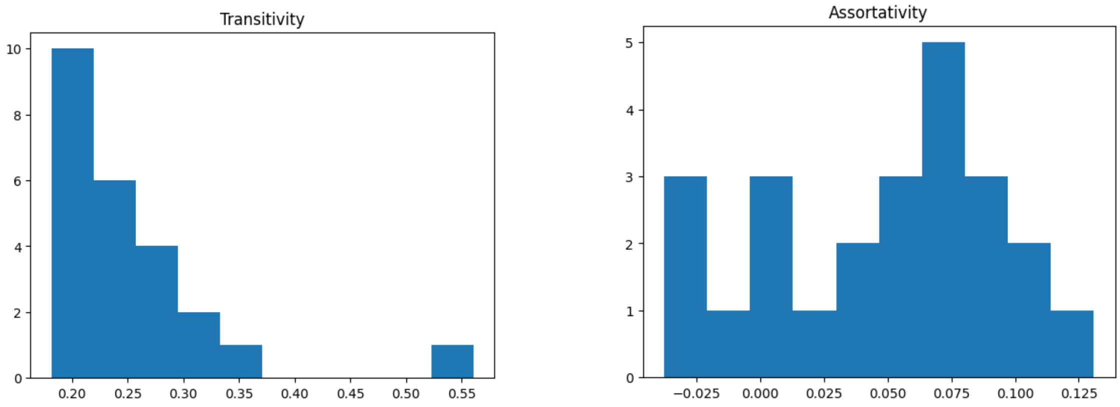
Figure 3.8: Community Detection and Modularity



(a) Average Clustering Coefficient

(b) Average Shortest Path Length

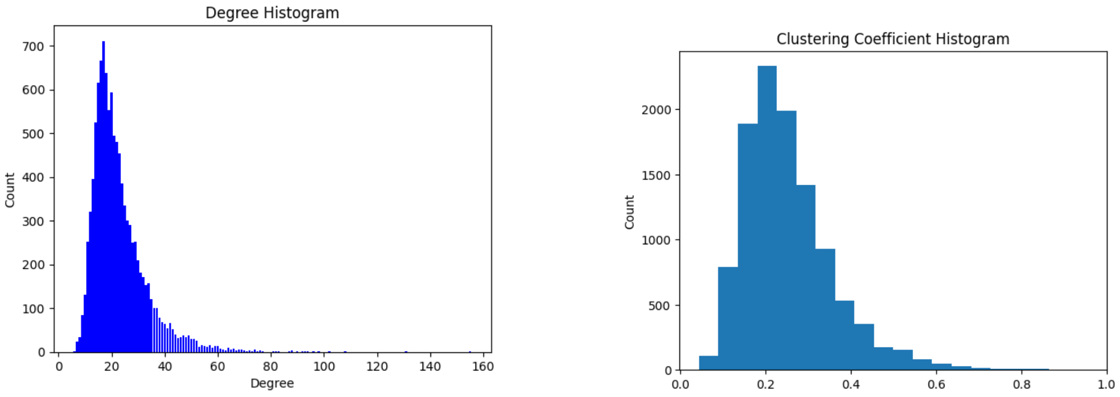
Figure 3.9: Average Clustering Coefficient and Shortest Path Length



(a) Transitivity

(b) Assortativity

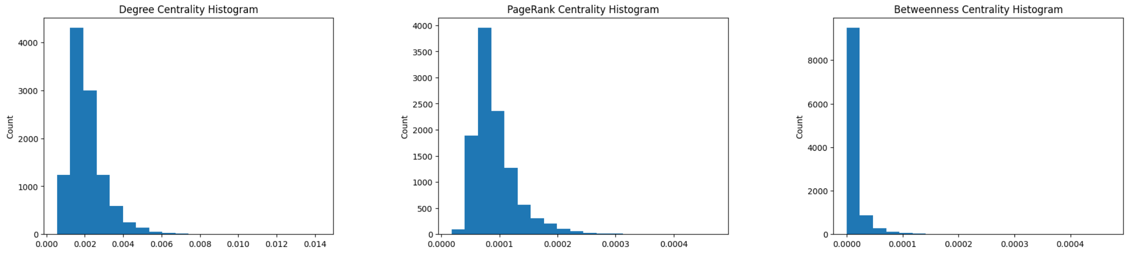
Figure 3.10: Transitivity and Assortativity



(a) Degree Histogram

(b) Clustering Coefficient Histogram

Figure 3.11: Degree and Clustering Coefficient Histograms

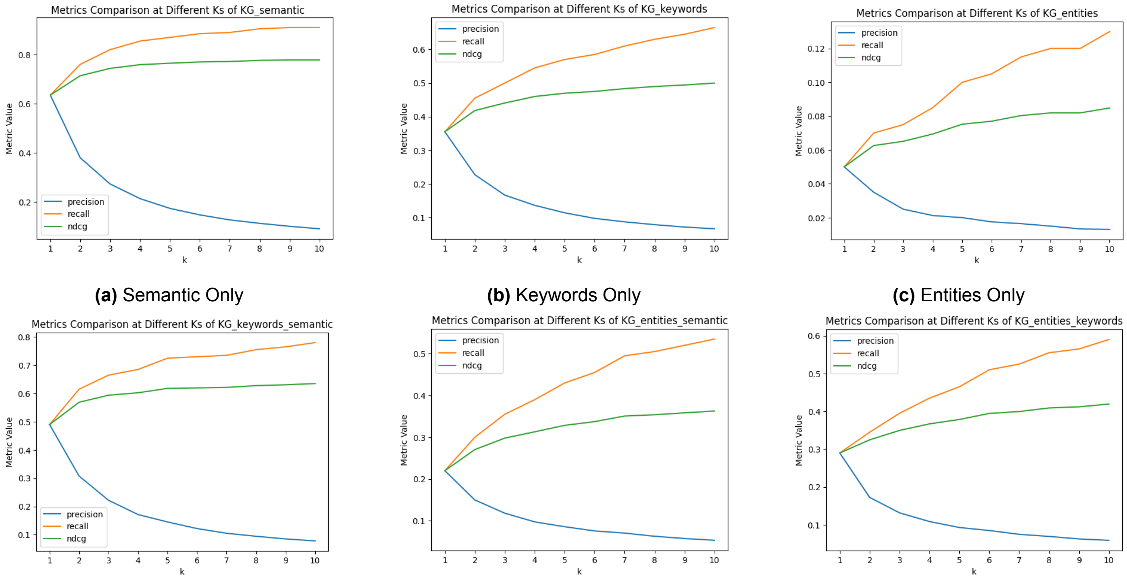


(a) Degree Centrality Histogram

(b) PageRank Centrality Histogram

(c) Betweenness Centrality Histogram

Figure 3.12: Centrality Histograms: Degree, PageRank, and Betweenness



(a) Semantic Only

(b) Keywords Only

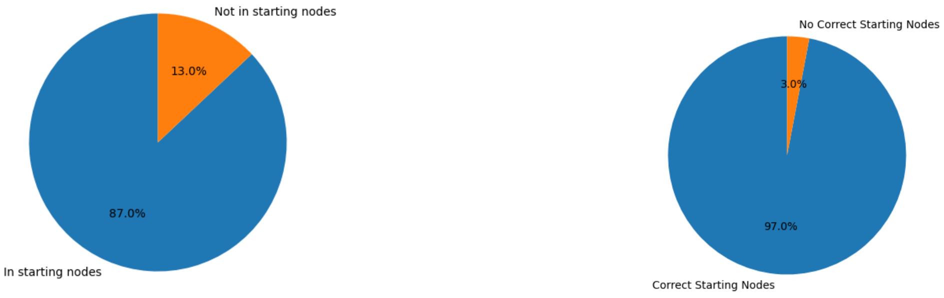
(c) Entities Only

(d) Semantic + Keyword

(e) Semantic + Entities

(f) Keywords + Entities

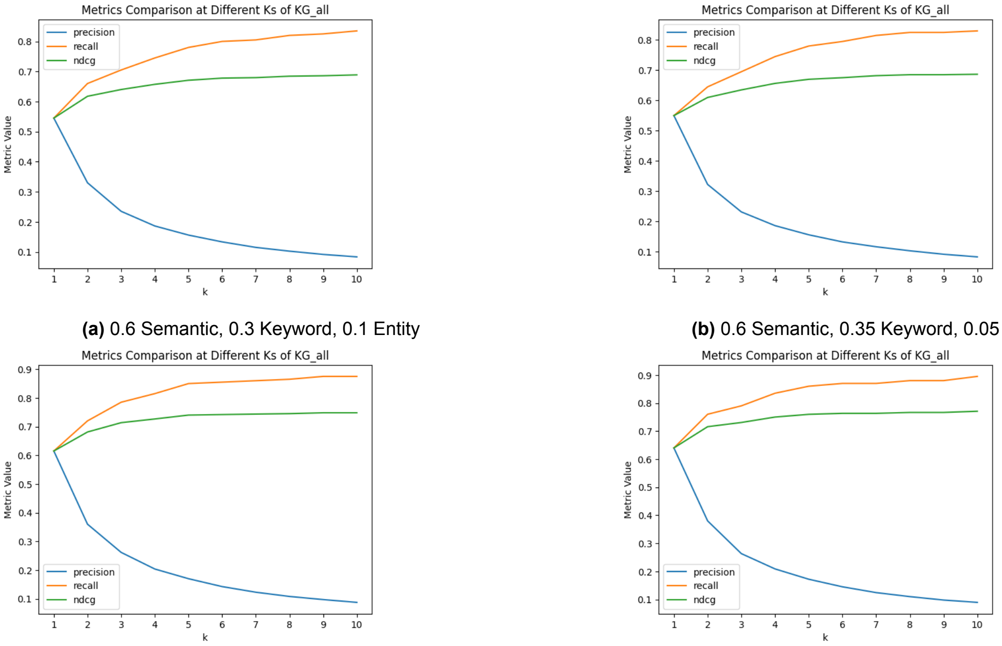
Figure 3.13: Performance of Balanced Different Feature Combinations



(a) Ground truth node presence in starting nodes

(b) Starting nodes in right document ratio

Figure 3.14: Ground Truth Presence and Starting Node Accuracy



(a) 0.6 Semantic, 0.3 Keyword, 0.1 Entity

(b) 0.6 Semantic, 0.35 Keyword, 0.05 Entity

(c) 0.75 Semantic, 0.2 Keyword, 0.05 Entity

(d) 0.8 Semantic, 0.15 Keyword, 0.05 Entity

Figure 3.15: Performance with Different Heuristic Weights

3.8.5. Graph Neural Network

Table 3.1: Comprehensive Results of GCN and SAGE Models during Training

Dataset Size	Model	Edge Features	Train MSE	Val MSE	Test MSE
100	GCN	With	0.0237	0.0235	0.0401
		Without	0.0239	0.0238	0.0407
	SAGE	With	0.0438	0.0371	0.0408
		Without	0.0313	0.0270	0.0418
1,000	GCN	With	0.0138	0.0122	0.0138
		Without	0.0140	0.0118	0.0142
	SAGE	With	0.0150	0.0115	0.0147
		Without	0.0170	0.0170	0.0172
10,000	GCN	With	0.0015	0.0014	0.0011
		Without	0.0017	0.0016	0.0012
	SAGE	With	0.0011	0.0010	0.0007
		Without	0.0033	0.0032	0.0028
100,000	GCN	With	0.0005	0.0006	0.0005
		Without	0.0005	0.0006	0.0005
	SAGE	With	0.0027	0.0028	0.0027
		Without	0.0051	0.0051	0.0050
1,000,000	GCN	With	0.0006	0.0006	0.0006
		Without	0.0004	0.0004	0.0004
	SAGE	With	0.0054	0.0054	0.0054
		Without	0.0058	0.0058	0.0058
2,170,600	GCN	With	0.0007	0.0007	0.0007
		Without	0.0004	0.0005	0.0004
	SAGE	With	0.0026	0.0027	0.0027
		Without	0.0059	0.0059	0.0059