VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

REPORT

CAPSTONE PROJECT

DEVELOPMENT OF AN LLM-BASED ACADEMIC AFFAIRS QA SYSTEM USING CONTEXT-ENABLED KNOWLEDGE GRAPH

MAJOR: COMPUTER SCIENCE

THESIS COMMITTEE	: **COMPUTER SCIENCE - 01**
SUPERVISORS	: **QUAN THANH THO, Ph.D.**
	: **BUI CONG TUAN, MEng.**
REVIEWER	: **TRAN TUAN ANH, Ph.D.**
	—o0o—
STUDENTS	: **HO NGUYEN NGOC BAO - 2052036**
	: **NGUYEN VIET PHUONG - 2052659**

HO CHI MINH CITY, JUNE 2024

VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

REPORT

CAPSTONE PROJECT

DEVELOPMENT OF AN LLM-BASED ACADEMIC AFFAIRS QA SYSTEM USING CONTEXT-ENABLED KNOWLEDGE GRAPH

MAJOR: COMPUTER SCIENCE

THESIS COMMITTEE	: **COMPUTER SCIENCE - 01**
SUPERVISORS	: **QUAN THANH THO, Ph.D.**
	: **BUI CONG TUAN, MEng.**
REVIEWER	: **TRAN TUAN ANH, Ph.D.**
	—o0o—
STUDENTS	: **HO NGUYEN NGOC BAO - 2052036**
	: **NGUYEN VIET PHUONG - 2052659**

HO CHI MINH CITY, JUNE 2024

Declaration of Authenticity

My team, which consists of Nguyen Viet Phuong and Ho Nguyen Ngoc Bao, declares that this Capstone Project was composed and implemented entirely by us under the guidance and supervision of Assoc. Prof. Quan Thanh Tho and Mr. Bui Cong Tuan at the Faculty of Computer Science and Engineering, Vietnam National University - Ho Chi Minh City University of Technology.

In the process of researching and implementing this Capstone Project, we have referenced multiple previous studies from other authors and ourselves. All of them have been fully and clearly stated in the References part.

This Capstone Project has not been published in any form under any circumstances.

The authors,
Nguyen Viet Phuong - Ho Nguyen Ngoc Bao

Acknowledgement

First and foremost, we would like to express our sincere thankfulness to Assoc. Prof. Quan Thanh Tho and Mr. Bui Cong Tuan. Thank you for allocating your valuable time to guide, orientate and encourage us throughout the outline and implementation phase of this Graduation Project.

Next, we are grateful that the lecturers at the Faculty of Computer Science and Engineering, Vietnam National University - Ho Chi Minh City University of Technology have delivered us sufficient fundamental knowledge, which serves as the springboard so that we can compose this thesis and contribute to the Vietnamese Computer Science field.

Last but not least, we would like to thank our families for always supporting us on our journey at university. Our parents' diligence has become the largest source of motivation for us to work hard. Furthermore, seniors at our companies also conscientiously guide and train us in a professional working environment when we take part in practical application projects, which really shapes our individual thoughts and behaviors to be better.

The authors,

Nguyen Viet Phuong - Ho Nguyen Ngoc Bao

Abstract

This project explores the use of prompting for Large Language Models (LLM) along with a simple retrieval method in order improve the task of question answering using a knowledge graph (KG) as context. Our approach includes a novel way to extract parametric memory from the LLM to augment retrieved triples from KG for question answering.

Key Highlights:

- KG-Enhancement LLMs: We explore methods to integrate the KG with LLMs, demonstrating its potential to improve understanding and response generation.
- Background Knowledge Retrieval: We devise a set of prompts to extract the LLM's background knowledge as additional context.

Project Contributions:

- Improved Question Answering Accuracy: The KG along with the background knowledge of an LLM serves to poignantly answer user questions, especially in KBQA downstream tasks as shown in the work.
- Practical Implications: Our work provides valuable insights for researchers and developers working on LLMs and KGs, and paves the way for the development of more sophisticated and accurate question-answering systems.

This project showcases the efficacy of KG in enhancing LLM capabilities for question answering. Leveraging insights from KG integration, and the nuances stored inside each LLM, we pave the way for more effective, personalized, and informative LLM interactions with users.

Contents

List of Tables

List of Figures

1

Introduction

In Chapter 1, we will present an overview of the urgency of the topic, objectives, and scope of the project’s research. The report outline will also be presented.

1.1 Problem statement

The integration of knowledge graphs has significantly revolutionized the way we represent and understand knowledge, mirroring the intricate thought processes of the human mind. This powerful tool has allowed us to establish meaningful connections between various concepts and entities, thereby creating a comprehensive network of information. Consequently, there has been a surge of interest in leveraging the capabilities of knowledge graphs for the purpose of question answering. This task, which involves providing accurate and concise responses to user queries, can greatly benefit from the rich semantic relationships inherent in knowledge graphs, enabling us to mimic human-like reasoning and deliver more precise and relevant answers.

Previous methods for question answering on knowledge graphs involve mostly employing traditional information retrieval techniques, semantic parsing, and rule-based systems. However, these approaches often struggle with the complexity and ambiguity of natural language, leading to suboptimal performance. Enters large language models, large language models have emerged as a promising solution to the challenges faced by traditional methods in question answering on knowledge graphs. These models, trained on vast amounts of text data, have the ability to understand and generate human-like text, thereby providing a more natural and intuitive interface for users. As such, there have been many works within recent years on integrating large language models with knowledge graphs, be it reasoning over knowledge graphs or leveraging knowledge graphs as context for question answering system.

Understanding these trends, we would like to propose an approach to combine large language models and knowledge graphs to alleviate the work of our university's Academic Affairs office for our university project.

1.2 Goals

The main goal of this project is to develop a pipeline for question answering based on a knowledge graph with a large language model. To do this, we will examine past works related to querying and reasoning on knowledge graphs as well as as the integration of large language models in the process. After which, we set out to find novel approaches to enhance the accuracy and efficiency of the question-answering process. This may involve exploring techniques such as knowledge graph embedding, entity linking, and relation extraction to better integrate the structured data of the knowledge graph with the unstructured data of the natural language questions.

1.3 Scope

In our Knowledge Graph research project, we have identified two challenges that we aim to address:

- Knowledge Retrieval is a crucial component of this project, as information from retrieved from knowledge graphs as well as other knowledge bases are vital in a question-answering system. The main task involved will be Entity Linking, with additional subtasks like transforming user question into a knowledge graph query, etc.
- Integrating knowledge graph into LLMs: Retrieved knowledge from LLMs are rarely used as is, rather, some additional steps are required in order to allow LLMs to use information from knowledge effectively in the question answering task.

1.4 Report structure

There are five chapters in this project's proposal:

- Chapter 1 outlines the problems, goals, and scope of the project.
- Chapter 2 focuses on providing a literature review relevant to the project, covering topics such as natural language processing and large language models.
- Chapter 3 discusses related works and recent developments in integrating knowledge graphs into Large Language Models (LLMs).
- Chapter 4 details our proposed approaches and the flow of our project.
- Chapter 5 goes into our experiment setups and our results and analysis.
- Chapter 6 provides a summary of the entire project proposal and outlines plans for future development.

2

Background Knowledge

In this chapter, we will discuss the essential knowledge for our work, including various approaches and our chosen solution. We will delve into the advantages and disadvantages of each approach, ultimately highlighting our rationale behind our chosen solution. This chapter will provide relevant knowledge for knowledge graphs along with large language models.

2.1 Large Language Models

Large Language Models (LLMs) are advanced artificial intelligence (AI) models trained on massive datasets of text and code. They use deep learning techniques, particularly transformer networks, to understand, summarize, generate, and predict new content. LLMs are considered "general-purpose" because they can excel at various tasks without specific training for each one. LLMs and their widespread popularity can be largely traced back to an incredibly influential research into the transformers architecture, published by Google back in 2017 [111].

There are 3 main types of LLMs, each will be discussed down below.

2.1.1 Encoder-only

Encoder-only large language models (LLMs) are a powerful type of LLM architecture that utilize only the encoder portion of a Transformer model. While Transformers typically have both an encoder and decoder, encoder-only models focus solely on understanding and representing the input information. This makes them particularly adept at tasks that require comprehending and analyzing language, including but not limited to: classification and information retrieval.

Advantages of Encoder-only LLMs:

- Efficient and lightweight: Removing the decoder makes encoder-only models computationally cheaper and faster to run, making them more scalable for real-world applications.
- Strong context comprehension: Focusing solely on the encoder allows for deeper understanding of the input text, leading to improved performance in tasks like classification and information retrieval.
- Simple and interpretable: Compared to decoder-based models, the architecture of encoder-only models is easier to understand and interpret, making them valuable for analyzing and debugging model behavior.

Disadvantages of Encoder-only LLMs:

- Limited output generation: Encoder-only models mainly focus on understanding input, not generating new text. This restricts their use for tasks like creative writing, dialogue generation, and machine translation, where generating novel sequences is crucial.

- Potential for factual errors: Since they don't directly generate text, encoder-only models rely on pre-trained data, which can contain biases or factual inaccuracies. These can be reflected in their outputs, leading to misleading information.

- Limited interpretability in complex tasks: While their architecture is simpler, interpreting the reasoning behind their outputs can be challenging in complex tasks. This makes it difficult to debug errors or understand how the model arrived at its conclusions.

2.1.2 Decoder-only

Decoder-only large language models (LLMs) are a rising star in the world of artificial intelligence, particularly in the realm of text generation. Unlike traditional LLMs, which often rely on both an encoder and decoder architecture, decoder-only models focus solely on the decoder component. This seemingly simple difference has led to remarkable advancements in the model's capabilities, making them particularly adept at tasks like text generation.

Advantages of Decoder-only LLMs:

- Simpler architecture: This translates to faster training times and lower computational requirements, making them more accessible for wider deployment.

- Zero-shot learning: Decoder-only models can often perform well on new tasks without any additional fine-tuning, showcasing their remarkable adaptability.

- Flexibility in generation: They can generate different creative text formats, like poems or code, with minimal adjustments, demonstrating their versatility.

Disadvantages of Decoder-only LLMs:

- Lack of understanding: Decoder-only models excel at pattern recognition and mimicking language styles, but they might not understand the true meaning or context of the text they generate.

- Limited reasoning and logic: Without an encoder to analyze the input context, decoder-only models struggle with tasks requiring logical reasoning or making inferences.
- Vulnerability to manipulation: The flexibility of decoder-only models can make them susceptible to manipulation through the prompts used. Malicious actors could potentially use carefully crafted prompts to generate harmful or misleading content.

2.1.3 Encoder-Decoder

Encoder-Decoder LLMs are essentially LLMs with both the encoder and the decoder portions. They excel at tasks that involve understanding an input sequence (like a sentence or paragraph) and generating a corresponding output sequence, with the output potentially having a different length and structure. They support a wider variety of tasks compared to Encoder-only or Decoder-only, which are: machine translation, question answering, text summarization and dialogue generation.

Advantages of Encoder-Decoder LLMs:

- Versatility and adaptability: They can handle the aforementioned tasks with surprising accuracy, with the possibility of being fine-tuned into more specialist roles, like code generation, if needed.
- Large-scale learning: They are trained on massive amounts of data, enabling them to capture complex linguistic patterns and generate highly accurate and creative outputs.

Disadvantages of Encoder-Decoder LLMs:

- Data bias: They can inherit biases present in their training data, leading to unfair or inaccurate outputs.
- Explainability: Understanding the internal workings of these models can be challenging.

- Resource-intensive: Training and running LLMs require significant computational resources.

2.2 Retrieval Augmented Generation

As mentioned above, generative artificial intelligence (AI) showcases its prowess in crafting text responses through large language models (LLMs) trained on massive datasets [30, 71, 86, 120], which are usually referred to as pre-trained models. These responses are trained to be harmless and user-friendly [7], and can offer comprehensive answers applicable to various queries, often referred to as prompts.

However, the limitation lies in the training process itself, confined solely to the data utilized for training a generalized LLM. These datasets might be outdated by weeks, months, or even years, and in certain corporate AI chatbots, it might lack specific documents or knowledge about the organization's products or services. Consequently, this discrepancy can lead to untruthful responses, which are usually referred to as [45], undermining trust in the technology among both customers and employees.

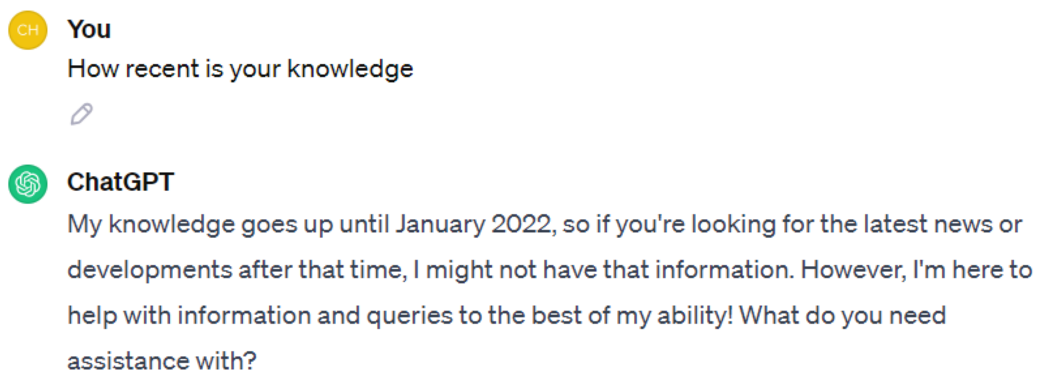

Fig. 2.1: LLM pre-trained models can only answer questions about events, people, and locations prior to the time its trained datasets are collected.

Introducing Retrieval-Augmented Generation (RAG) [61], a solution aimed at optimizing LLM outputs by incorporating targeted information without fine-tuning the pre-trained model. These supplementary data are always more current than the LLM data, can be managed and updated in real-time, and are tailored to a particular organization or institution. Thus, the generative AI system becomes capable of furnishing more contextually relevant responses based on the most recent and pertinent data available.

Following the release of "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks" [61], a 2020 paper authored by Patrick Lewis and a team at Facebook AI Research, many inceptions and variations of RAG gained traction among the generative AI community. This concept has garnered support from academic and industry researchers as a means to significantly enhance the effectiveness of generative AI systems without the cost inefficiency of fine-tuning a new model.

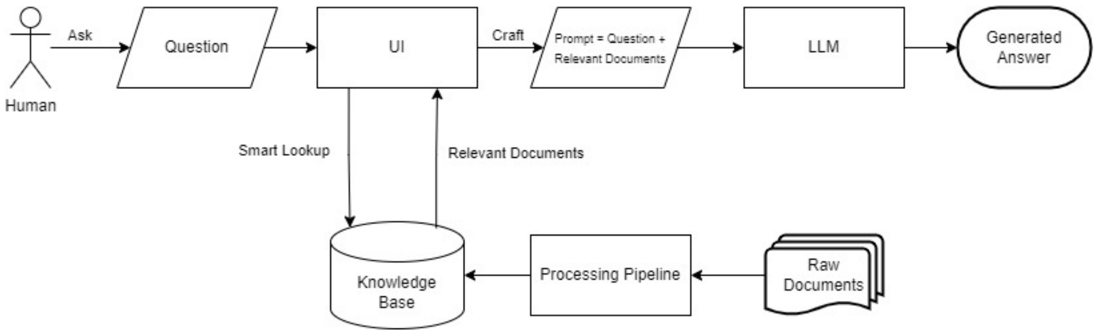

Fig. 2.2: General RAG pipeline.

To illustrate, consider a sports league aiming to enable fans and media to utilize chat interfaces for accessing data, and querying about players, teams, historical aspects, rules, current statistics, and standings. While a generalized LLM could handle inquiries regarding history, rules, or descriptions of team stadiums, it would lack the capability to discuss recent games or provide updates on specific player injuries due to its limited information. Updating the LLM comprehensively requires substantial computational resources, rendering it impractical to keep it constantly up-to-date and helpful.

In contrast, the sports league possesses various other sources of information such as databases, data repositories, player biographies, and comprehensive news feeds covering each game. RAG allows the generative AI to integrate this diverse information. Consequently, the chat interface can deliver more timely, contextually appropriate, and accurate information to users.

In essence, RAG can serve as a facilitator, enabling LLMs to furnish more accurate and contextually rich responses, thereby significantly improving their efficacy [105]. In summary, the RAG system has a couple of advantages:

- Cost-effective implementation: The computational and financial costs of fine-tuning the LLM models for organization or domain-specific information are higher than RAG implementation.

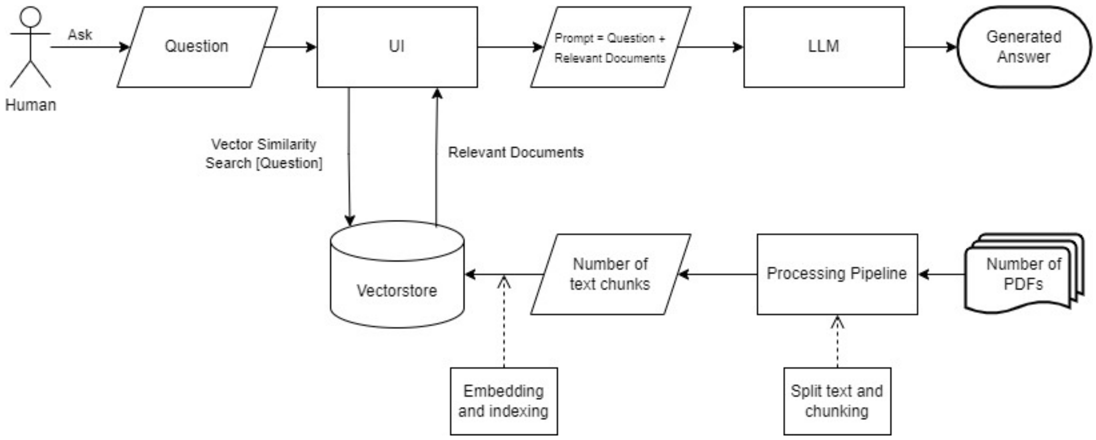

Fig. 2.3: Vectorstore RAG system.

- Current and tailored information: RAG allows developers to feed the latest research, statistics, or news to the generative models.

- Enhanced user trust: The system output can include citations or references to sources so users can confirm themselves.

But just like any other technique in machine learning, RAG is not magic and contains flaws that you might want to consider whenever implemented. Different RAG systems have different pros and cons so for the simplicity of this section, we will consider some disadvantages or flaws of using vectorstore RAG system:

- Multi-hop QA: Given a complex question that requires multiple different retrieved documents to answer, one simple vectorstore RAG can't handle that. Maybe another RAG system like Knowledge Graph RAG can but bear in mind that this is a common flaw of these RAG systems.

- Information Loss: To generate the answer we have to chunk the text and generate embedding for the chunks, retrieve the chunks by semantic similarity search, and generate a response based on the text of the top k chunks. The steps are not ensured to provide enough context for the LLM model to answer. Given the chunk size and overlap you choose, the original documents might be directed into non-sense chunks. Semantic similarity search alone can't retrieve all the useful information given a complex multi-hop-ish question.

- Reduce LLM models output performance: Feeding the limited input-sized models with lengthy context will vastly reduce its performance and clarity in its response [67].

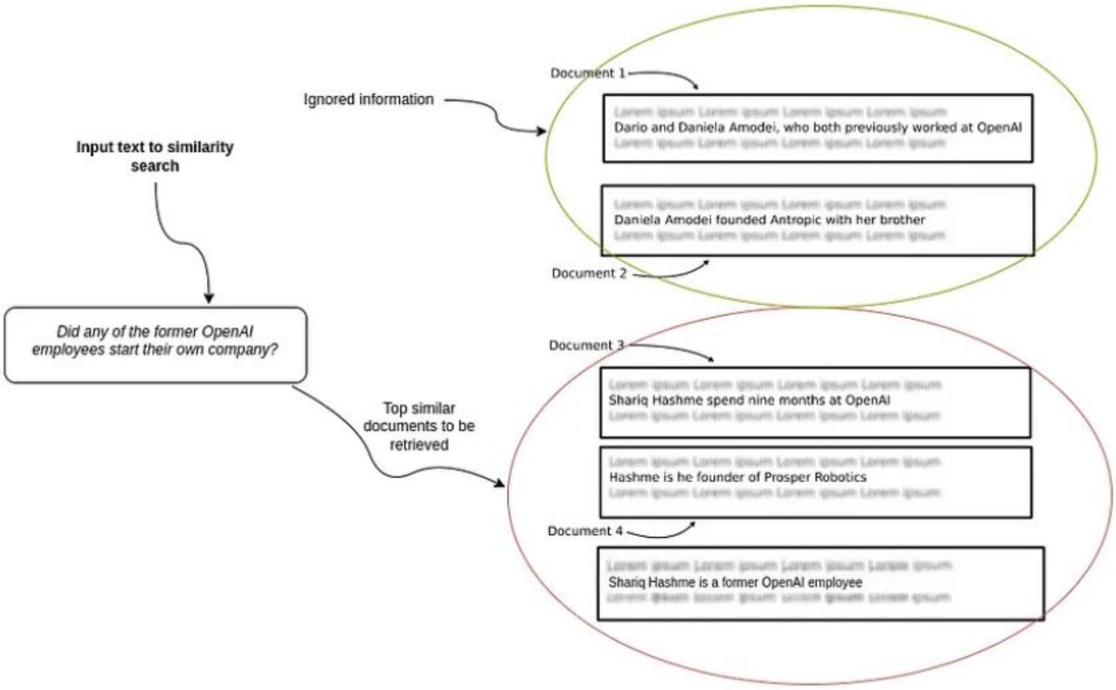

Fig. 2.4: An example where the similarity search might return some duplicated information and ignore other relevant information.

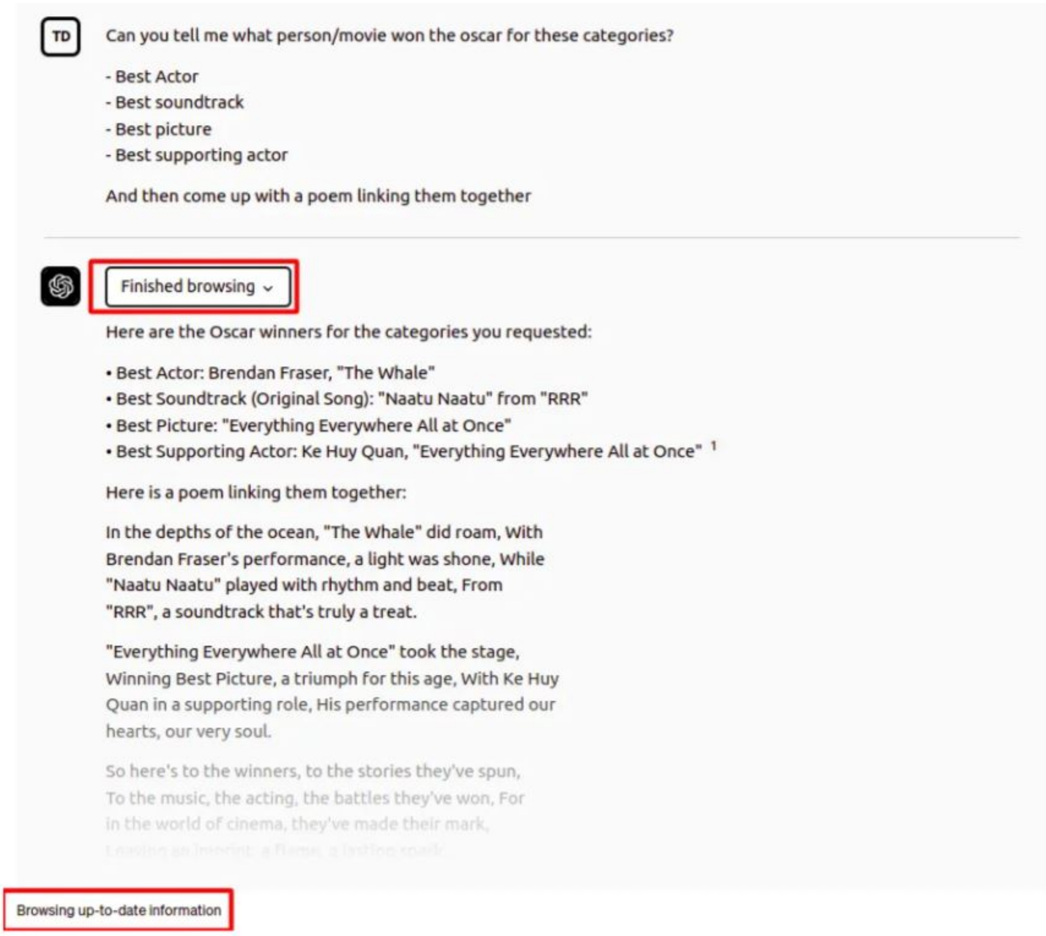

Fig. 2.5: GPT-4 with browsing plugin can be considered a successful RAG system.

2.3 Knowledge Graph

A *Knowledge Graph (KG)* [89, 93, 74, 37, 44] is a knowledge base that uses a graph-structured data model or topology to represent and operate on data. Knowledge graphs are often used to store interlinked descriptions of entities – objects, events, situations, or abstract concepts – while also encoding the semantics or relationships underlying these entities [122, 81].

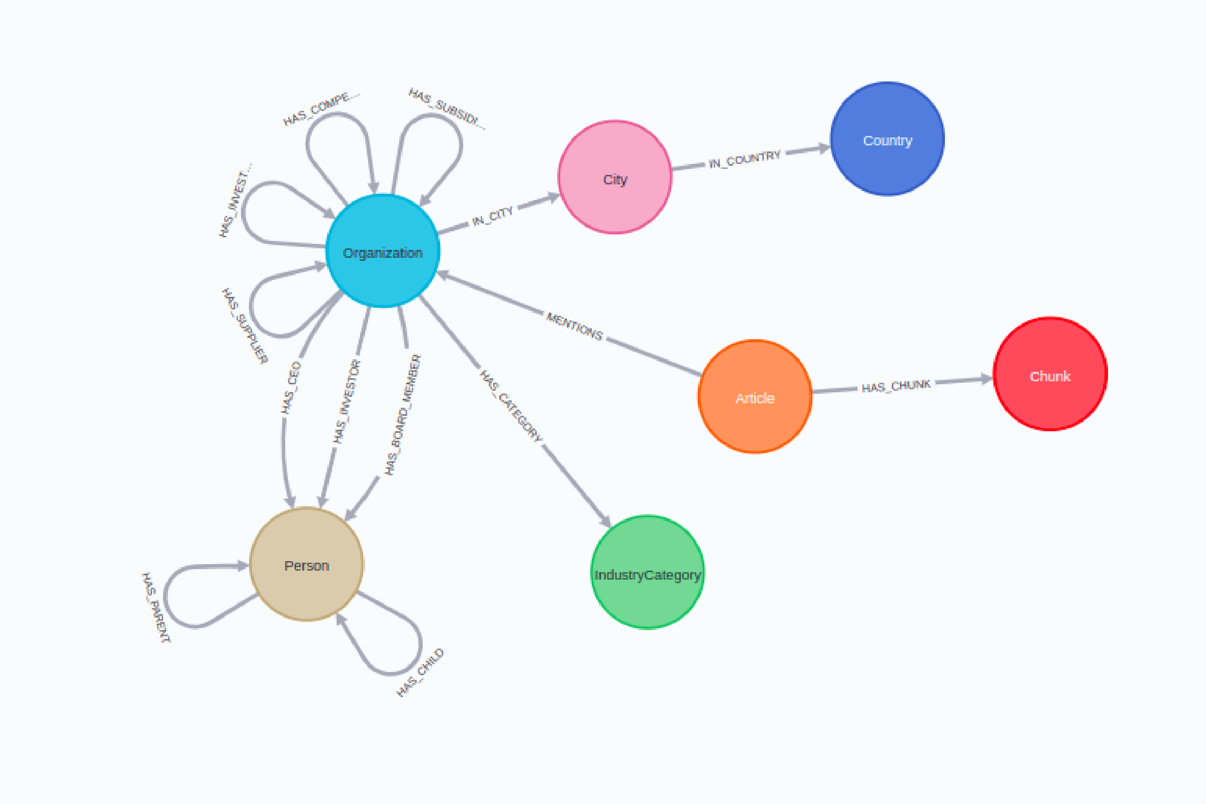

Fig. 2.6: This figure is a knowledge graph representation of a subset of Diffbot's database [80]

Knowledge graphs are historically associated with and used by famous search engines like Bing, Google, and Yahoo; knowledge engines and question-answering services such as WolframAlpha, Apple's Siri, and Amazon Alexa; and social networks such as LinkedIn and Facebook [122].

Formerly only used in search engines and recommender systems, recent developments in graph neural networks and representation learning have brought noticeable traction to the knowledge graph, which is increasingly used in scientific research [79]. In the scope of this work, we discuss the concept knowledge graph as a type of database that stores information in the form of a graph, with nodes representing entities and edges representing relationships between them. They are designed to store complex, hierarchical, and interconnected data. Debatable speaking, a complete knowledge graph system is comparable to other database management systems such as Relational Database Management Systems (RDBMS), NoSQL Database, and Object-Oriented Database Management System (OODBMS).

In summary, a knowledge graph consists of two main components: entities/nodes

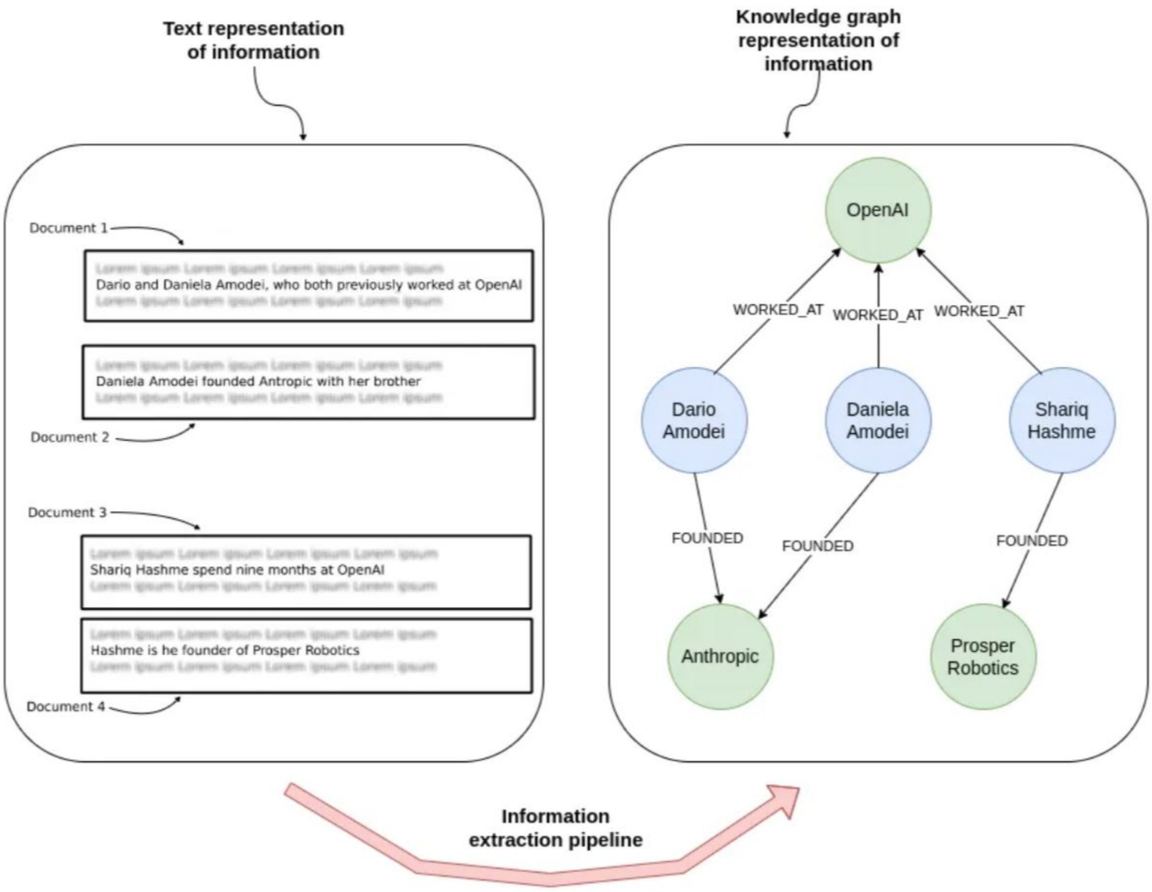

Fig. 2.7: A visualization on extracting entities and relationships from text to construct a knowledge graph[13]

(People, places, objects, events, or concepts) and relationships (How these entities are connected). Knowledge graphs are useful in a variety of applications, such as:

- Search engines: Knowledge graphs can help improve the accuracy and relevance of search results by providing a deeper understanding of the relationships between entities and concepts.
- Recommendation systems: Knowledge graphs can be used to build recommendation systems that take into account the complex relationships between entities, such as "if you like this movie, you may also like this book."
- Fraud detection: Knowledge graphs can be used to detect fraud by analyzing patterns of relationships between entities and identifying suspicious behavior.
- Predictive maintenance: Knowledge graphs can be used to predict when maintenance is required by analyzing patterns of relationships between entities and identifying potential failures.

Particularly for this work, the target application of knowledge graphs is question-answering with LLMs. Integrating knowledge graphs into LLMs offers 2 main benefits:

- Improved Accuracy and Factuality: Knowledge graphs provide LLMs with access to a structured and verified source of information, reducing the risk of factual errors or hallucinations commonly seen in pure LLM outputs. This is crucial for tasks like question answering, summarization, and report generation.
- Enhanced Relevance and Coherence: Knowledge graphs can guide the LLM toward relevant information based on the context of the user's request. This leads to more focused and on-topic outputs, avoiding irrelevant tangents or digressions.

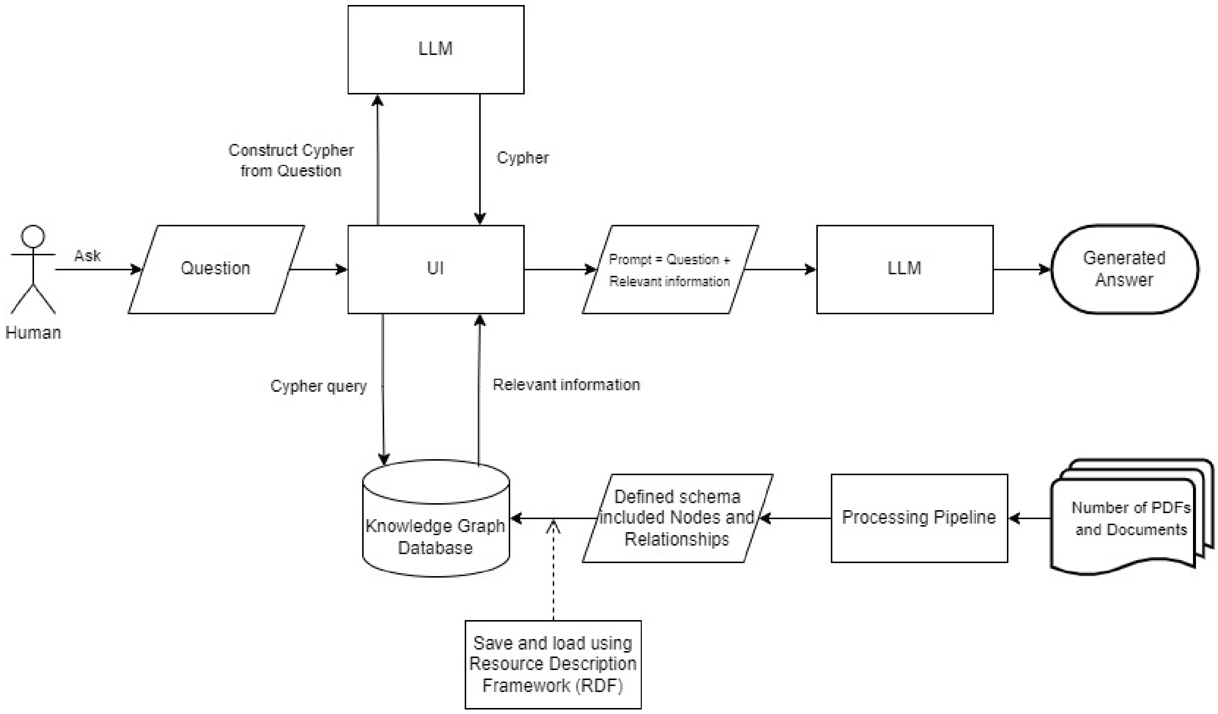

Fig. 2.8: The figure is an example pipeline of KG + RAG. Cypher, which will be explained below, is a SQL-type query language for some DBMS using a knowledge graph database.

However, knowledge graphs still have their limitations:

- Sparsity and lack of uncertainty/noise handling: Knowledge graphs usually build up from a given Schema (Nodes' properties and Relationships) but can such knowledge graphs predict a missing link in the relationships of the schema (e.g. In one schema there is a relation "is_father_of" between A and B, B and C. Can the system infer that A has a relation "is_grandfather_of" to C and add a new edge together with the new relationship? This problem can be discussed in future work.

- Difficulty in constructing high-quality knowledge graphs: Knowledge graphs can guide the LLM toward relevant information based on the context of the user's request but there are many different pipelines for constructing it in the community. A solution like Neo4j might be great in terms of exposure but not a full-fledged solution that can build, manage, and maintain KG in terms of time and space (e.g. a KG system used by an institute in 2019 can be maintained, updated, and queried with new information in 2020)

- Inefficient evaluation and benchmarking: Estimation of the accuracy of a large-scale knowledge graph (KG) often requires humans to annotate samples from the graph, research on keeping human annotation costs low steadily gained traction in recent years but no SOTA yet. [31]

3

Related works

In Chapter 3, we will explore previous research relevant to Knowledge Base Question Answering (KBQA) and Knowledge Graph-Augmented Large Language Models (KG-Augmented LLMs), examining existing approaches and considering their strengths and limitations. Researchers have predominantly focused on two overarching methodologies in KBQA: Information Retrieval-based (IR-based) and Semantic Parsing-based (SP-based) approaches.

IR-based methods, exemplified by works like KV-Mem [76], GRAFT-Net [98], PullNet [97], and others [88, 91, 125, 35, 133, 29], prioritize retrieving relevant information directly from Knowledge Bases (KBs). These methods often construct subgraphs to facilitate answer determination in a smaller solution space, denoted this step as information retrievers (IF) module i.e. subgraph retrieval. Then, KBQA method following this line performs reasoning on that retrieved subgraph to predict the final answer, denoted this step as a reasoning module i.e. answer generator, machine-reader model (MRC).

On the other hand, SP-based methods concentrate on translating queries into logical forms executable against KBs, with notable examples including STAGG [129], UHop [19], TextRay [9], and several others [57, 56, 34, 18, 92, 123, 70, 68, 124, 134]. These SP-based techniques vary in their strategies, ranging from step-wise query graph generation using ranking features [57, 56, 34, 18] to sequence-to-sequence model utilization [17, 22, 127, 38, 40, 33, 15, 63] for S-expression (usually a SPARQL) generation.

By positioning our work within this landscape of KBQA research coupled with the advancement of LLMs [60, 24, 14], we aim to contribute novel insights and advancements while building upon the collective efforts of previous studies. Through the synthesis of existing methodologies and our proposed framework, we strive to address critical gaps and push the boundaries of KBQA systems.

3.1 Knowledge Graphs (KGs)

Various approaches and techniques have been proposed for constructing knowledge graphs, utilizing a range of data sources and extraction methods. This section provides a brief overview of some significant works in this field, highlighting its proficiency and unique attributes in retrieving information.

An overview systematic of graph construction is discussed in a survey on knowledge graph construction [137].

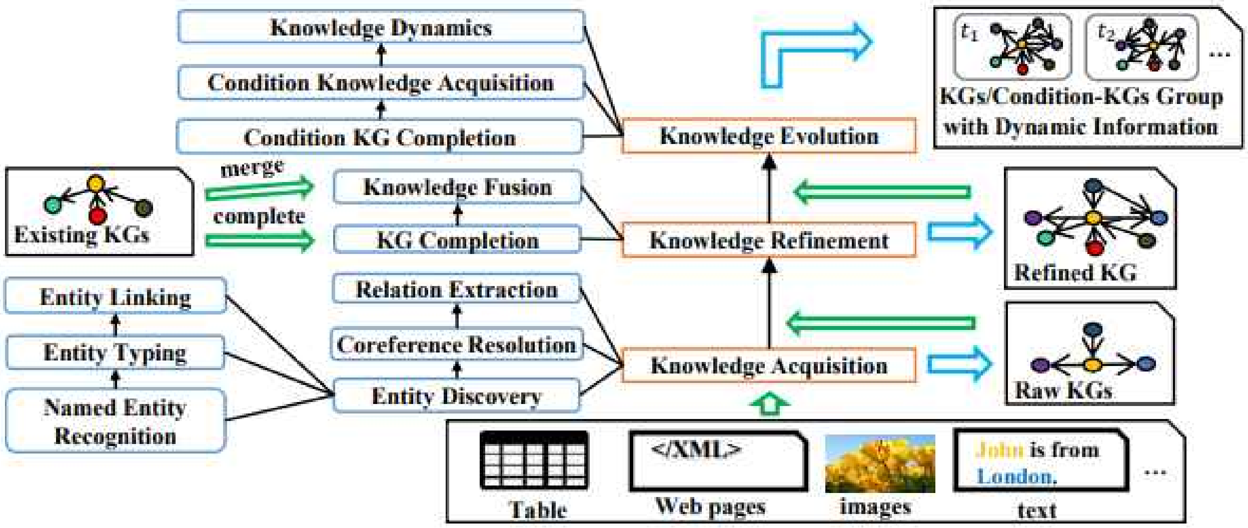

Fig. 3.1: The general process of constructing a knowledge graph

Numerous knowledge graphs, such as Freebase [10] and Wikidata [85], have emerged through crowd-sourcing efforts. While these graphs prove effective, the construction process itself is a laborious and resource-intensive manual undertaking. Consequently, several papers have explored automating this process to reduce the resources required for graph construction, thereby advancing the field.

The development of knowledge graphs is commonly categorized into two main types: top-down and bottom-up. The top-down approach involves initially defining the ontology (or data schema) and then extracting knowledge based on that predefined structure. Conversely, the bottom-up approach seeks to discover and extract knowledge from the data, followed by the definition of the ontology for the knowledge graph based on the acquired data.

The top-down approach, given its reliance on specialists with specific domain knowledge, introduces potential barriers due to its domain-specific nature. Consequently, for this project, we have chosen the bottom-up approach. This approach is expected to alleviate domain-related obstacles, enabling a more automated graph construction process.

To gain further insights into the Knowledge Graph Development process, the paper titled "Defining a Knowledge Graph Development Process Through a Systematic Review" [103] presents an analysis of 57 graphs spanning the years 2016 to 2021. The distribution of these 57 graphs is detailed in the table below.

Year	Count of articles
2016	1
2017	5
2018	10
2019	11
2020	24
2021	6
Type of article	
Domain-specific	47
Methodological	10
Type of KG development	
Bottom-up	41
Top-down	16

Fig. 3.2: Summary Details of the Selected Articles [103]

It's clear that the majority of the articles focused on the development of domain-specific knowledge graphs, with about 47 articles covering this topic. These articles presented knowledge graphs built for specific purposes and the techniques used in their development. Another category of articles provided a more theoretical overview of knowledge graphs and development methods.

Most of the articles focused on the bottom-up knowledge graph development approach (Table 1). While many articles did not specify the type of development approach used, the distinction was made based on whether ontology development was the first step of knowledge graph development.

It's evident that there is a lack of a general method for constructing knowledge graphs. Even when adopting a general methodological approach, challenges arise in data source formatting. Structured data, which has an explicit format, is often found in tables or relational databases. Semi-structured data has a certain structure, exemplified by XML data. On the other hand, unstructured data, such as text, lacks a predefined structure. Extracting knowledge from semi-structured and unstructured data, especially text, requires more effort and sophisticated techniques compared to the relatively straightforward identification of entities and relationships in structured data.

In our real-world approach, unstructured data constitutes the primary data source, while other types are scarce and necessitate human labor. Some studies, like [4] or [2], have undertaken the construction of knowledge graphs from textual content. However, even in the initial step of knowledge acquisition, they rely on labeled datasets to train models for obtaining the necessary nodes for the graph. This reliance introduces a domain-specific barrier, as previously discussed.

Conclusion

In order to create a knowledge graph systematically, we need to follow three key steps: knowledge acquisition, knowledge refinement, and knowledge evolution. Our main focus in this project is on the first step—knowledge acquisition—which involves gathering elements from various types of data to establish the base of a knowledge graph. This includes tasks such as entity recognition (or entity extraction), coreference resolution, and relation extraction. Entity recognition tasks aim to identify instances of entities within the data, while coreference resolution tasks locate pairs of mentions that refer to the same entity. Subsequently, relation extraction tasks establish the semantic relationships between entities. We will explore relevant downstream tasks performed on Knowledge Graphs (KGs) or Knowledge Bases (KBs) in the following discussion.

3.2 Knowledge Base Question Answering (KBQA)

As mentioned above, Knowledge Base Question Answering (KBQA) has seen significant progress in recent years. Each category offers distinct mechanisms and advantages for tackling the complexities of KBQA downstream tasks.

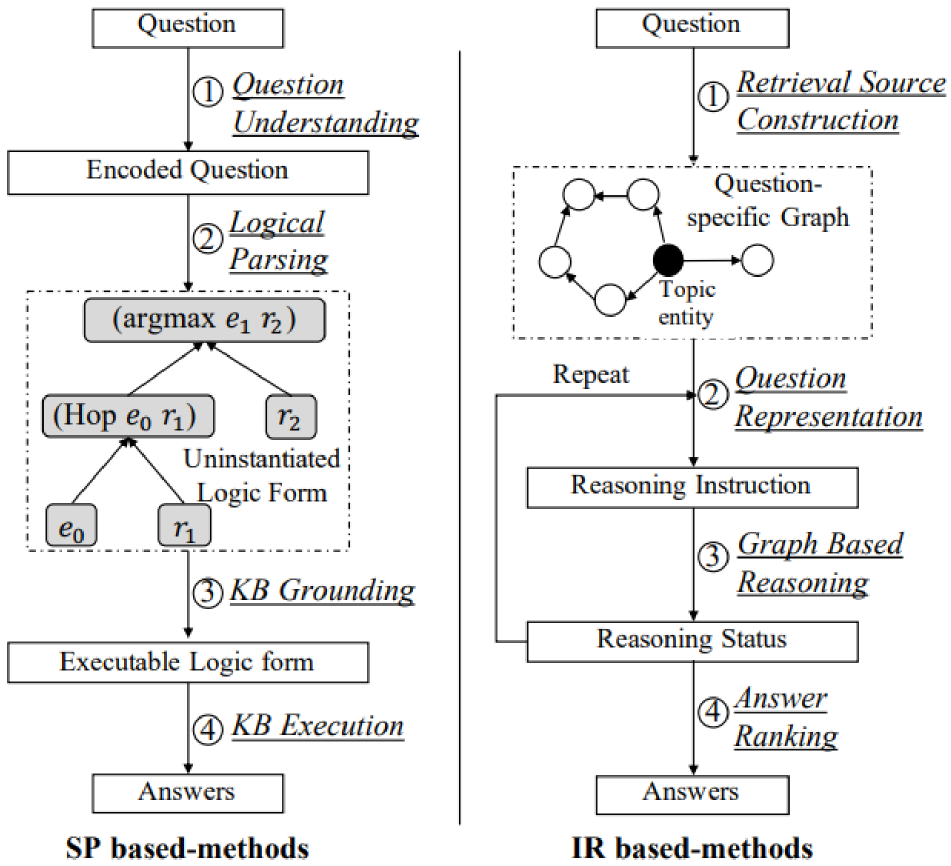

Figure 2: Illustration of two mainstream approaches for complex KBQA.

Fig. 3.3: Two mainstream approaches for complex KBQA [58]

Information Retrieval-based Methods (IR-KBQA)

IR-based methods, also coined as Direct-Answer-Prediction methods, primarily focus on retrieving relevant subgraphs from Knowledge Bases (KBs) based on natural language questions and then ranking the entities or paths to determine the best answers. Early approaches like KV-Mem [77] used distinct encoding during memory access to enhance document reading. GRAFT-Net [98] and PullNet [97] constructed question-specific subgraphs to extract answers. Advanced methods such as EmbedKGQA [88], NSM+h [35], and TransferNet [91] introduced innovative mechanisms for optimizing subgraph-related KBQA models.

Recent works have highlighted the importance of constructing question-specific subgraphs through neural models. For instance, GraftNet [98] used personalized PageRank scores, while PullNet [97] proposed a framework to iteratively construct subgraphs and reduce the solution space. He et al. [35] enhanced neural state machines with intermediate supervision signals. UniKGQA [47] unified subgraph extraction and answer reasoning via semantic matching and propagation modules.

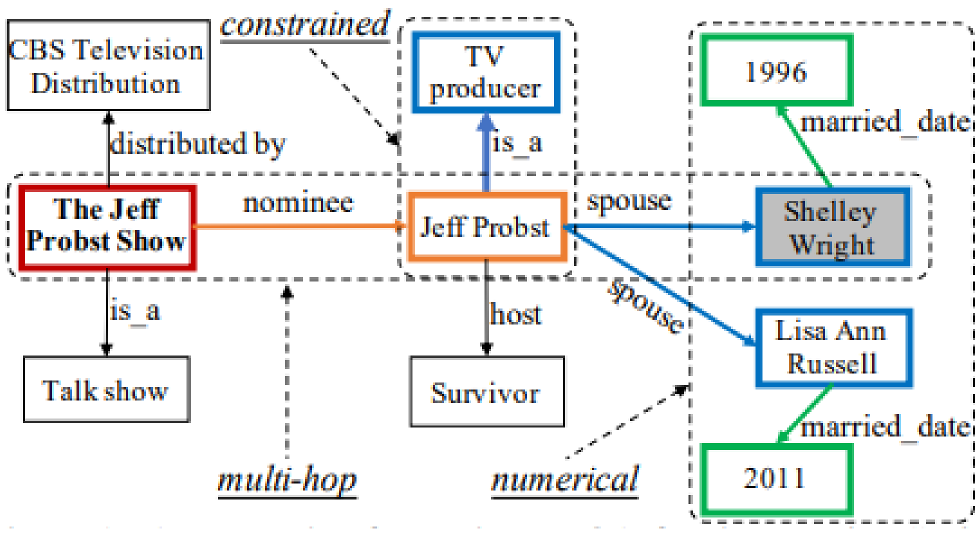

Fig. 3.4: An intermediate subgraph, usually retrieved from step in the process of an IR-KBQA pipeline. The topic entity and the answer entity are shown in the bold font and shaded box respectively [58]

Despite the effectiveness of these approaches, they often face challenges related to the interpretability of the reasoning process and the efficiency of knowledge retrieval. To address these issues, TIARA [92] proposed multi-grained retrieval to improve robustness and efficiency, while recent methods like ChatKBQA [70] employed large language models (LLMs) with fine-tuning techniques for better semantic parsing and knowledge retrieval. However, these two methods follow the line of Semantic Parsing-based Methods, which empirical results show that it often produces more accurate answers than IR-based ones.

Semantic Parsing-based Methods (SP-KBQA)

SP-based methods aim to translate natural language questions into executable logical forms, such as SPARQL or S-expressions, which can then be executed against KBs. Classic SP-based approaches include STAGG [129] and UHop [19], which used step-wise query graph generation. Seq2Seq generation methods [17, 15] have also become popular, leveraging pre-trained language models (PLMs) for translating questions into logical forms.

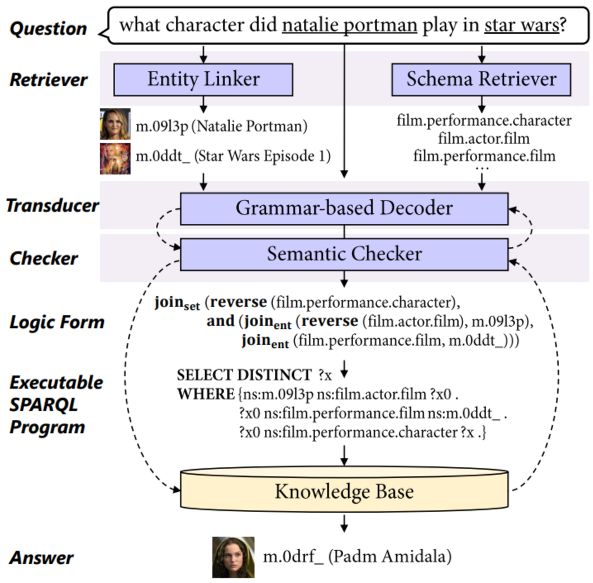

Fig. 3.5: Logical Forms (LFs) constructing, grounding, and checking using a fine-tuned seq2seq by ReTraCk [17]

Recent advancements have focused on improving the efficiency and accuracy of semantic parsing. Techniques like case-based reasoning [22], constrained decoding [124], and multi-grained retrieval [92] have shown promising results. Moreover, methods such as RnG-KBQA [127], GMT-KBQA [38], and FC-KBQA [134] have employed PLMs to enhance the generation of logical forms and reduce errors.

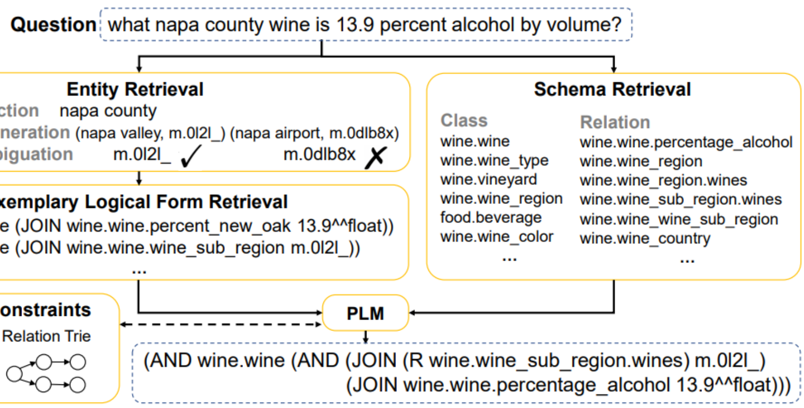

Fig. 3.6: Logical Forms (LFs), usually hold an important role in most SP-KBQA systems as they will be executed against a query engine i.e. Wikidata [112] to achieve the final answer, iteratively drafted and mass-produced leveraging the power of LLMs [92]

Other Methods

Several methods don't follow either the traditional two-staged IF-KBQA or SP-KBQA. DeCAF [131] jointly generates both logical forms from the SP-based method and direct answers from the IR-based method, and then leverages the merits of both of them to get the final answers.

Think-on-Graph (ToG) [99], a KG-Augmented LLM method, introduced a novel paradigm where KGs and LLMs work in tandem, complementing each other's capabilities in each step of graph reasoning. ToG does retrieve subgraphs on its implementation but is not traditional IR-based as it iterates between exploration and reasoning steps while performing a beam search on KG.

Chain-of-Knowledge (CoK) [64] introduces Adaptive Query Generator (AQG) following the spirit of SP-based methods but the implementation focuses more on LM Prompting (Prompt Engineering) as it is being compared with Chain-of-Thought (CoT) [117], Self-Consistency (SC) [116], Verify-and-Edit (VE) [136], ReAct [126].

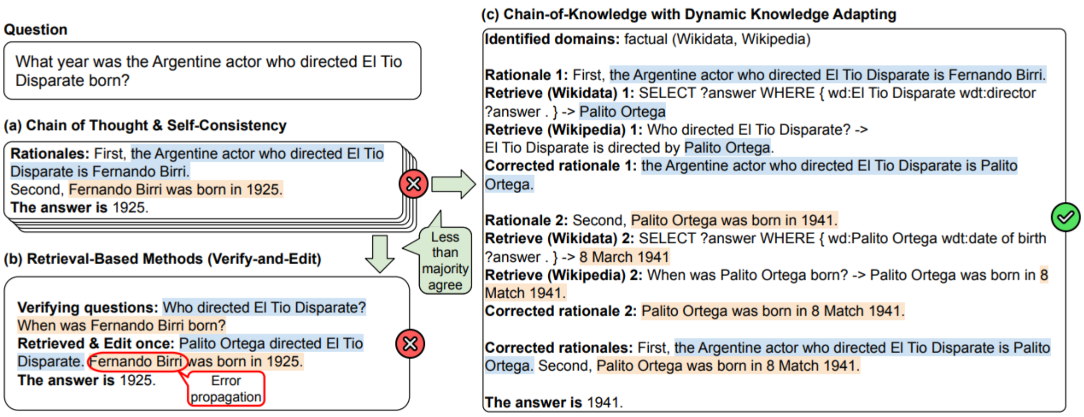

Fig. 3.7: Adaptive Retrieval Strategy

Conclusion

Overall, both IR-based and SP-based methods have their unique strengths and challenges. While IR-based methods excel in end-to-end training and retrieval efficiency, SP-based methods offer more interpretable reasoning processes but might suffer from overhead and inflexibility. The integration of advanced retrieval techniques, knowledge graph embeddings, and the latest capable LLMs continues to push the boundaries of KBQA, enabling new, more accurate, and robust question-answering systems.

3.3 KG-Augmented LLMs

The emergence of large language models (LLMs) like ChatGPT [14] and open-source models such as Llama-2 [106] and ChatGLM2 [27] has significantly impacted KBQA. These models, with their powerful semantic parsing capabilities, have simplified many traditional NLP tasks but still fell prey to 'hallucination'. Techniques like Chain-of-Thought (CoT) [117] and Graph-of-Thought (GoT) [8] have been introduced to mitigate hallucinations and improve reasoning over KGs.

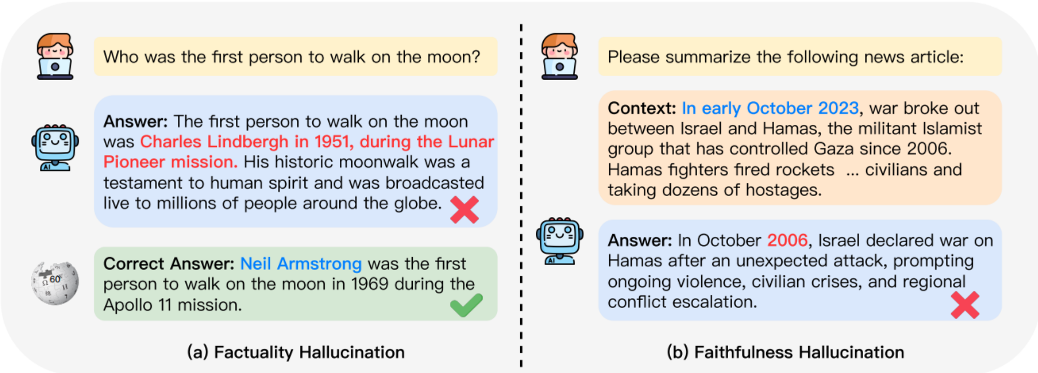

Fig. 3.8: LLMs usually fell prey to 'hallucination' [39]

KGs provide an extensive repository of real-world knowledge structured in triplet forms, offering advantages in explicit knowledge representation. Initial studies [135, 100, 114, 115] focused on the use of corpora derived from KG triplets to train LLMs, thereby improving their performance across various tasks. However, these approaches directly embed knowledge from KGs into LLM parameters, implying that any update to the KG necessitates LLM retraining. This poses flexibility and transparency challenges. More recent studies have incorporated relevant KG knowledge into prompts to enhance LLM inference, primarily divided into two strategies: "Reasoning before Retrieval" and "Retrieval before Reasoning".

Reasoning before Retrieval

In the "Reasoning before Retrieval" approach, Chain-of-Knowledge [64] employs LLMs to decompose questions into sub-questions, using fine-tuned models for query generation and retrieving sub-question answers to form the final response. StructGPT [46] starts with the question entity, uses LLMs to identify related relationships and entities, and extends the search in a single-chain manner until the answer is located. Despite its effectiveness in reducing the search range, this strategy can face situations where the answer entity lies outside the predetermined search boundary. Furthermore, methods for directly generating queries, as described in Li et al. [64], heavily rely on KG completeness, failing to fully leverage the potential for deep and dynamic integration between LLMs and KGs.

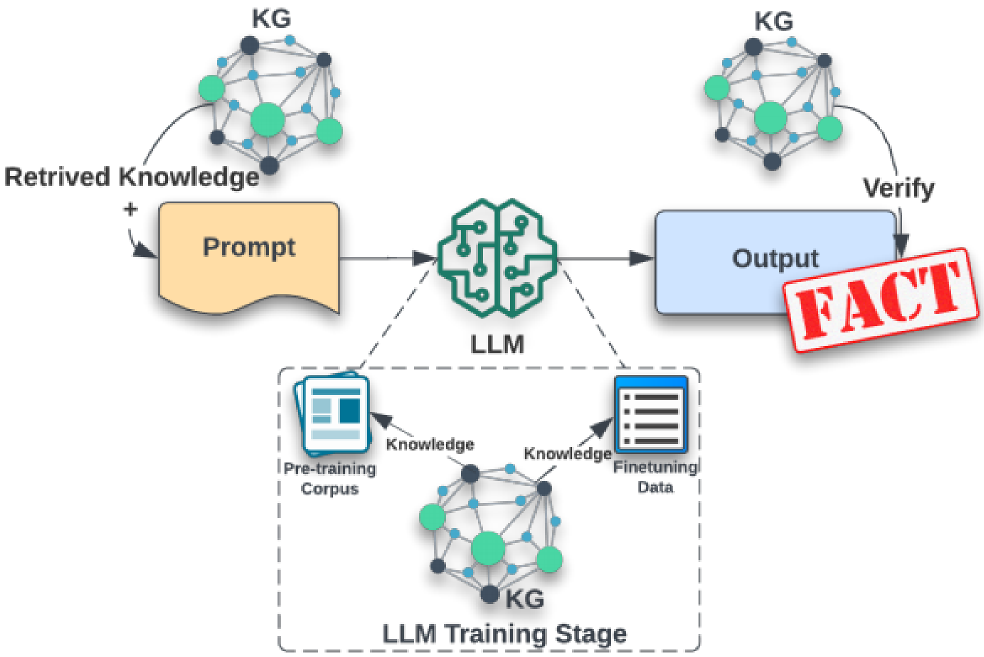

Fig. 3.9: The KG-Augmented LLMs or KG-Enhanced LLMs [3]

Retrieval before Reasoning

The "Retrieval before Reasoning" strategy, exemplified by KAPING [6], samples top K triples for prompting based on semantic similarity, focusing on retrieving single-hop triples. However, this falls short of multi-hop reasoning and often rephrases retrieved facts in its responses, not fully utilizing the profound reasoning capabilities of LLMs. Drawing from human cognitive behavior, Wang et al. [113] guide the model to reason step by step based on retrieved triples. Despite this, its integrated retrieval component incurs additional training costs and shows limited transferability across different KGs. In contrast, Li et al. [65] demonstrate significant flexibility in multi-hop knowledge subgraph retrieval using ChatGPT and effectively engage the LLM's implicit knowledge, enabling more comprehensive joint reasoning with external explicit knowledge.

Conclusion

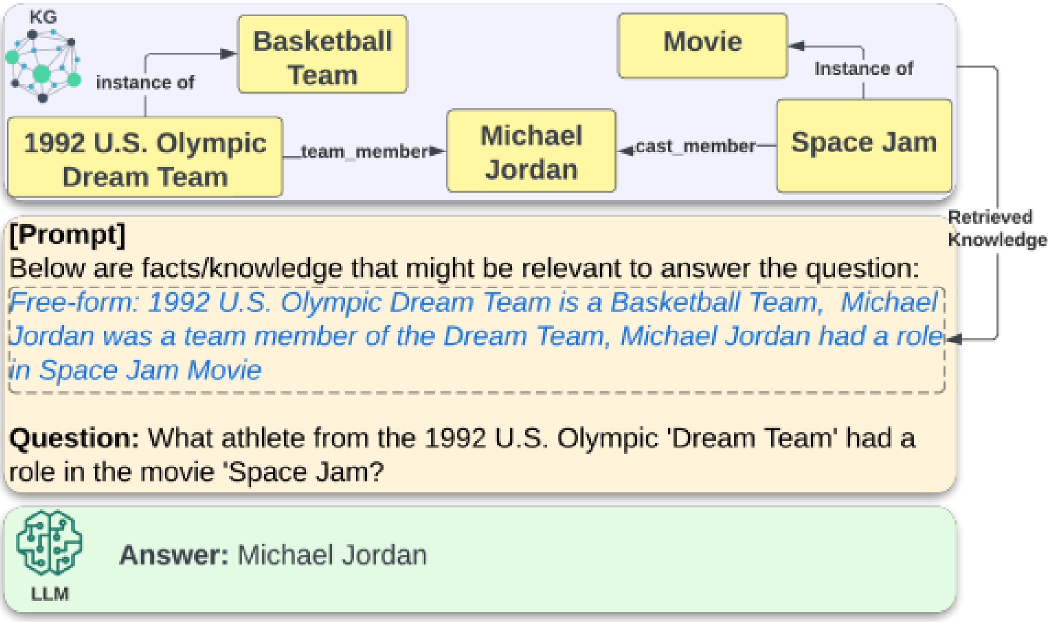

Fig. 3.10: The KG-Augmented LLMs at inference step also coined as Knowledge-aware inference [3]

The integration of Knowledge Graphs (KGs) with Large Language Models (LLMs) represents a significant advancement in the realm of Knowledge-Based Question Answering (KBQA). The hybrid approach of combining KGs with LLMs can be implemented through "Reasoning before Retrieval" and "Retrieval before Reasoning" strategies. Each method presents unique advantages and challenges but still can sometimes fall short in handling complex, multi-hop reasoning tasks.

The future of KG-augmented LLMs lies in optimizing the balance between these two strategies. Emphasis should be placed on developing adaptive systems that can dynamically switch between reasoning and retrieval processes, depending on the nature of the query. This will enhance both the accuracy and efficiency of KBQA systems. Additionally, addressing the challenges of KG completeness and the seamless integration of implicit and explicit knowledge remains crucial for the continued advancement of this field.

3.4 Entity Linking (EL), Retrieval, and Knowledge Graph Embeddings (KGEs)

We delve into critical components and methodologies integral to Knowledge Base Question Answering (KBQA) systems (especially following the line of Information Retrieval-based architecture i.e. IR-KBQA), particularly focusing on Entity Linking (EL), two main types of Retrieval, and Knowledge Graph Embeddings (KGEs). These elements are foundational in enabling the efficient retrieval and accurate interpretation of information from vast knowledge bases, crucial in reducing the solution space of KBQA tasks into subgraphs containing the answer candidate. By exploring the evolution and nuances of these methods, we aim to highlight their roles and interconnections within the KBQA landscape. Each subsection provides a detailed examination of the techniques, their contributions, and the comparative performance, setting the stage for a comprehensive understanding of current advancements and future directions in KBQA research.

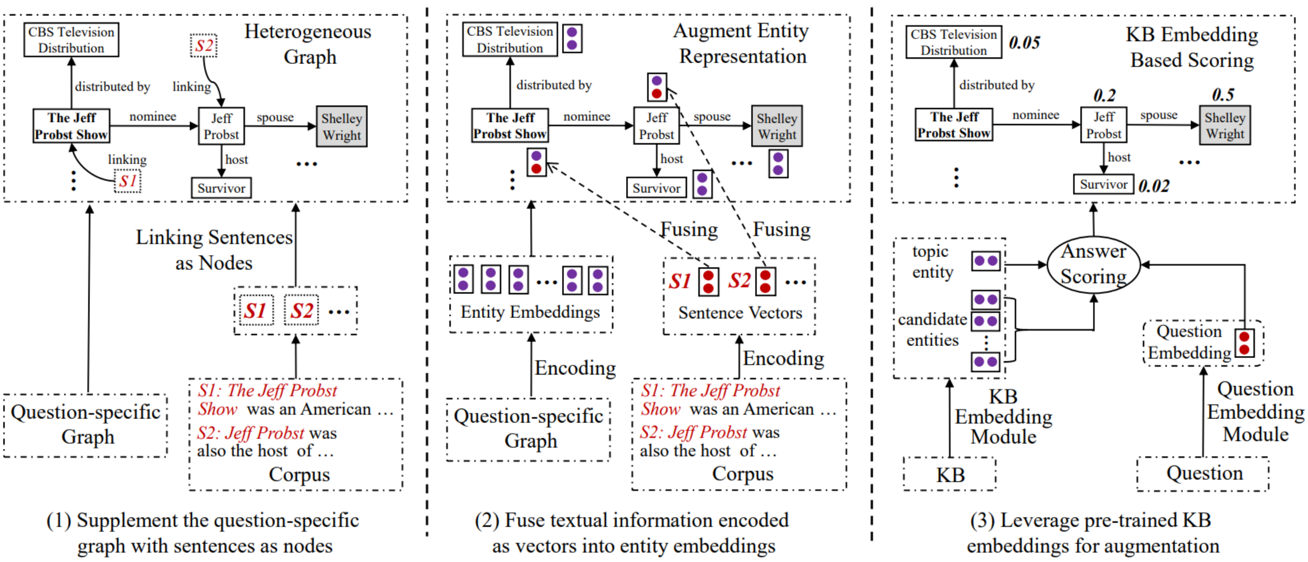

Fig. 3.11: Illustration of three categories of methods to supplement the incomplete KB, in a bottom-up style [59]

Entity Linking

Entity Linking is a pivotal component in Knowledge Base Question Answering (KBQA), where textual mentions are associated with corresponding entities in a knowledge base. This process enables systems to proceed with downstream tasks such as question answering, information retrieval, and semantic analysis effectively. By linking entities mentioned in natural language queries to their counterparts in knowledge bases, KBQA systems can accurately interpret user intents and provide precise answers. It's usually the first step of most 2-staged KBQA systems (both IR-KBQA and SP-KBQA).

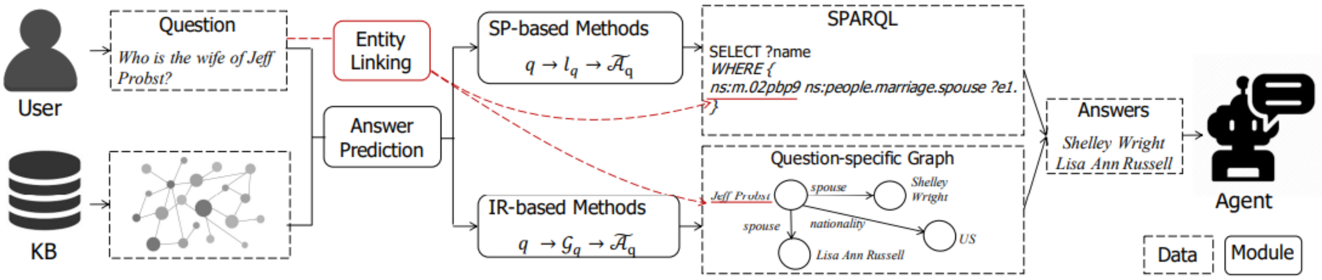

Fig. 3.12: The entity linking procedure is shown in the architecture of KBQA systems [59]

Sparse Retrieval

Sparse Retrieval methods, such as TF-IDF [95] and BM25 [87], have long played a crucial role in KBQA by efficiently retrieving candidate documents or passages from a large corpus. These methods rely on the frequency and distribution of terms within documents to determine relevance to a given query, making them valuable components in the initial phase of the KBQA pipeline. Despite their simplicity, they often provide strong baselines for various retrieval tasks, particularly in scenarios where computational resources are limited or the corpus size is extensive.

Dense Retrieval

Dense Retrieval methods, such as Dense Passage Retrieval (DPR) [52] and SimCSE [32], have been incorporated into KBQA [22, 131, 70] to improve the retrieval of relevant knowledge. These methods map queries and documents into dense vectors, facilitating efficient retrieval based on vector similarities.

Knowledge Graph Embedding

Incorporating knowledge graph embeddings has also been a focus of recent research. Methods like TransE [12] and RotatE [101] represent entities and relations in high-dimensional spaces, enabling effective multi-hop question answering. ConvE [23] and

InteractE [110] utilized convolutional neural networks to learn scoring functions between entities and relations, enhancing the reasoning capabilities over KGs.

Comparison

Across SimCSE, Contriever [42], and BM25, ChatKBQA [70] had analyzed the efficiency of retrieval in the retrieval phase reveals the efficacy of the Generate-then-Retrieve approach. Comparing entity retrieval (ER) and relation retrieval (RR) using logical form generation (AG-R) with traditional retrieval from natural language questions (NL-R). Their study's result showcases the superiority of AG-R over NL-R, particularly pronounced for RR.

Conclusion

In conclusion, our exploration into the realms of Entity Linking, Sparse Retrieval, Dense Retrieval, and Knowledge Graph Embeddings underscores the multifaceted landscape of Knowledge Base Question Answering (KBQA) systems. These components collectively form the backbone of KBQA, enabling accurate interpretation of user intents and facilitating precise answers through efficient retrieval and semantic analysis.

Across various retrieval models, including SimCSE, Contriever, and BM25, AG-R consistently outperforms NL-R, particularly pronounced in relation retrieval scenarios. This is also reflected in the comparison of different methods in KBQA conducted by us. Most of the time, the SOTA place of a KBQA downstream task's dataset is usually held by an AG-R method also known as the SP-KBQA system.

3.5 Parametric Memory vs Non-Parametric Memory

Even though LLMs can be wonderful black-box reasoners, blindly delegating the entire reasoning module to them can be quite catastrophic if facing ever-expanding question/-solution space [43, 53], or deploying in a real-world application [48] or the talking-point discussed in the section KG-Augmented LLMs above. We denoted Parametric Memory, Background Knowledge, and Implicit Knowledge all pointed at the same component stored inside LLMs. Several studies in this section aim to answer one simple question: can LLMs be used as knowledge bases?

Adaptive Retrieval Strategy

Adaptive Retrieval [72], by incorporating non-parametric memories only when necessary, uses a threshold to decide when to retrieve information, for example, if the asked entity frequently in both the question and the pre-trained data. For more popular queries, where LMs like GPT-3 have sufficient memorized knowledge, retrieval is skipped. For less popular queries, retrieval is employed to supplement the LM's knowledge. For optimization, the popularity threshold, which varies by relationship type, is determined using a development set to maximize adaptive accuracy. Larger models benefit more from Adaptive Retrieval. On the other hand, smaller models, which lack substantial memorized knowledge, automatically retrieve more frequently and thus see no significant performance gains from this method.

Generate Parametric Knowledge Before Read

In contrast to the previous retrieve-then-read pipeline, GENREAD [132] is essentially a generate-then-read pipeline that yields better results than traditional IR-based or SP-based methods without the need for retrieving subgraphs or construction logical form. Specifically, it first prompts a large language model to generate contextual documents based on a given question and then reads the generated document to produce the final answer. The reader can still be a large model i.e. InstructGPT [83] used under a zero-shot setting, or a small one i.e. FiD [41] fine-tuned with generated documents on the training split of the target dataset.

Conflicting Answer Candidates

Prior research on Retrieval-Augmented Generation (RAG) largely assumes that information from various knowledge sources is consistent, often neglecting how models integrate information from their parametric memory with retrieved knowledge. Chen et al. [16] introduces four perturbations—negation, future tense, modal verbs, and text infilling—to challenge models in question answering, testing their confidence by installing counterfactuals into the context space. Question-answering models usually leverage extensive knowledge sources, including either up to one hundred retrieved passages or large retrieved subgraphs coupled with the parametric knowledge within LLMs.The study proposes a calibration approach to discourage models from presenting a single answer when multiple conflicting answers are retrieved.

Conclusion

While LLMs excel as powerful black-box reasoners, relying solely on their parametric memory can be problematic, especially when faced with complex and ever-expanding question/solution spaces or real-world applications. Adaptively incorporating non-parametric memories based on query popularity, shifting from traditional retrieval-based pipelines to a generate-then-read approach, addressing inconsistencies in retrieved information backed by a positive correlation between pre-training data frequency and memorization, suggesting that popular entities are better handled by LLMs. Overall, the integration of both parametric and non-parametric strategies, along with adaptive and generative approaches, is key to enhancing the robustness and accuracy of LLMs in all realm of bridging LLMs into useful applications.

3.6 LM Prompting and In-context Learning (ICL)

Language model (LM) prompting and In-context Learning (ICL) [14] involves providing specific inputs or contexts to guide the model's behavior and enhance its performance on various downstream tasks. This section discusses various techniques implemented in LLMs to gain knowledge and improve performance specifically for Knowledge-Based Question Answering (KBQA). These techniques include zero-shot learning, few-shot learning, the Chain-of-Thought approach, self-consistency, GraphPrompt, and Think-on-Graph.

I/O a.k.a Zero-shot

Zero-shot learning, also known as input-output (I/O) prompting, involves providing a language model with a task description or input without any prior examples. The model relies solely on its pre-trained knowledge to generate responses. This approach tests the model's ability to generalize from its existing knowledge base to new, unseen tasks, demonstrating its inherent capabilities. Researchers [119] have shown to improve zero-shot learning via instruction tuning but the limitation of computational cost still hinders this approach.

Few-shot

While large language models exhibit impressive zero-shot capabilities, they still struggle with more complex tasks in a zero-shot setting. Few-shot learning [14] enhances the language model's performance by providing it with a few examples of the task at hand. LLaMA [107] established that the few-shot comes into light when language models are scaled to a sufficient size [51]. This technique includes a small number of input-output pairs in the prompt, allowing the model to learn from these examples and apply similar patterns to generate responses for new inputs. Few-shot learning is particularly useful in scenarios where limited labeled data is available.

Chain-of-Thought

Chain-of-Thought (CoT) prompting encourages the model to generate intermediate reasoning steps before arriving at a final answer. By breaking down complex problems into a series of logical steps or smaller problems, this approach improves the model's ability to handle tasks that require multi-step reasoning and enhances the transparency and interpretability of its responses.

Self-consistency

Self-consistency involves generating multiple responses (mass-producing) to a single prompt and then selecting the most consistent or frequent answer based on some defined metrics. This method leverages the diversity of outputs from different model runs to improve the reliability and accuracy of the final response. By aggregating these outputs, self-consistency aims to mitigate the variability and uncertainty inherent in single-shot predictions.

GraphPrompt

GraphPrompt [69] is an innovative approach that incorporates graph structures into the prompting process. By representing contextual information and relationships as graphs, this method enhances the model's understanding of complex interdependencies within the data. GraphPrompt leverages the rich, structured representation of graphs to improve the performance of language models on tasks requiring deep contextual knowledge and relational reasoning.

Think-on-Graph

Think-on-Graph [99] is a cutting-edge technique that combines in-context learning with graph-based reasoning. This approach involves constructing and reasoning over graphs during the prompting process, allowing the model to "think" through complex problems by navigating and manipulating graph structures. Think-on-Graph aims to enhance the model's problem-solving abilities and performance on tasks that benefit from structured, relational thinking.

Conclusion

Overall, LM prompting and ICL have proved to empower LLMs for many NLP downstream tasks, including Knowledge-Based Question Answering (KBQA) which is focused on in this work. We discussed various prompting techniques that leverage minimal examples or context to guide LLMs towards superior KBQA performance. These techniques hold promise for improving performance in question answering [20], information extraction [28], and numerical reasoning tasks [62], as demonstrated in their respective research. Our work builds upon these findings by incorporating insights from studies on prompt construction methods, such as input-label pairing [78] and example retrieval [66]. While incorporating explanatory task instructions has been suggested to further enhance performance [55], this remains a direction for future work.

3.7 Chapter Conclusion

The groundwork has now been laid. We've explored established approaches in KBQA and KG-Augmented LLMs. We've identified the strengths and limitations of existing techniques, particularly in the areas of retrieval [70, 72, 132], reasoning [14, 117], and parametric memory utilization [16, 26, 36, 49]. These findings have raised a burning question: "Can we take full advantage of underlined LLMs' memory and their powerful capability to enhance the reasoning and retrieval step of such KBQA system?"

This epiphany steamed from that pure curiosity paves the way for us to delve into the core of our contribution in the next chapter, "Proposed Methods". Here, we'll present our novel methodologies for knowledge graph integration in question answering (precisely in the downstream KBQA task), building upon the valuable insights gleaned from prior research and collective effort. Our aim is not only to introduce these approaches but also to elucidate their theoretical underpinnings and practical applications. Through our proposed methods, we aspire to offer a comprehensive framework that contributes meaningfully to the advancement of integration methodologies in the grand scheme of KBQA systems and relevant fields.

4

Proposed Methods

Building upon the foundation laid in the previous chapter, Chapter 4 will introduce our proposed methodologies to address the limitations and shortcomings identified in the existing previous work. Here, we will give an overview of our proposed method, describing the theoretical foundation that underpins our approach mainly influenced by the original work of Baek et al. 2023 [6]. We will describe the synergistic blend of existing retrieval external knowledge sources [6, 46] with our invoking strategy to retrieve background knowledge from LLMs' parametric knowledge sources [132, 70] and augmented it into the original pipeline, providing insight into how our fusion method can lead to significant improvements in both Hits@1 and F1 when dealing with questions that might have appeared in the LLM pre-trained data [72].

Lastly, we will discuss the detail construction of our Knowledge Base Question Answering system, highlighting the integration of our proposed methods into a fully functioning system, drawing from our understanding of both IR-based and SP-based systems. Through these proposed methods, we aim to offer an effective and robust framework that pushes the boundaries of current KBQA systems, mostly IR-based methods. We strive to create a paradigm where improved reasoning ability complements a more efficient retrieval strategy to create a highly effective and versatile model. We hope our work contributes meaningfully to future advancements in the grand field of KBQA.

4.1 Preliminaries

In this section, we review laid-down concepts of our work. Followed by the problem statement i.e. task formulation for the information retrieval-based knowledge-base question-answering task.

Definition 1: Knowledge Base

Knowledge Base (KB). A KB $K = \{(s,r,o) \mid s \in \mathscr{E}, r \in \mathscr{R}, o \in \mathscr{E} \cup \mathscr{L}\}$ is an RDF graph, also known as **Knowledge Graph (KG)** where $G \subseteq \mathscr{E} \times \mathscr{R} \times \mathscr{E} \subseteq K$, consisting of the triples (s,r,o) where s is an entity, r is a relation, and o can be an entity or a literal. Each entity $e \in \mathscr{E}$ in the entity set $\mathscr{E}$ is represented by a unique ID, e.g., $e.id = "m.0fm2h"$, and can be queried to get the English label of the entity as $e.label =$"Benjamin Netanyahu". Each relation $r \in \mathscr{R}$ in the set of relations $\mathscr{R}$ consists of multiple levels of labels, e.g. $r = "government.government\ position\ held.appointed\ by"$. Besides, a literal $l \in \mathscr{L}$ is usually "integer" (e.g., $l = "32"$), "float" (e.g., $l = "3.2"$), "year" (e.g., $l = "1999"$), "year&month" (e.g., $l = "1999-12"$), or "date" (e.g., $l = "1999-12-31"$).

Definition 2: IR-based Knowledge Base Question Answering

For a given question q, the task of Knowledge Base Question Answering (KBQA) is to obtain the answer(s) $A_q \subseteq \mathscr{E}$ based on the knowledge graph G. Information retrieval (IR)-based KBQA methods aim to maximize the probability $\Pr(e \in A_q)$ to distinguish A_q from other entities.

Since exploring the entire knowledge graph G is notoriously well-known for being computationally expensive, most practical IR-KBQA methods operate under the assumption that a question-relevant subgraph $S_G^* \subseteq G$ exists (usually 1 to 4-hop), where $\Pr(e \in A_q \mid q, G) = \Pr(e \in A_q \mid q, S_G^*)$. Therefore, IR-KBQA methods are divided into two stages in practice. The first stage extracts a question-relevant subgraph to approximate $S_G^* \subseteq G$, and the second stage maximizes the probability of the correct answers. These stages are known as preliminary subgraph retrieval and answer reasoning (or final subgraph retrieval/re-ranking), respectively.

From a probabilistic perspective, preliminary subgraph retrieval models a latent distribution $\Pr_\phi$ and maximizes $\Pr_\phi(S_G^*)$, while answer reasoning models a distribution $\Pr_\psi$

on the retrieved subgraph to approximate Pr. We formulate these stages as follows:

$$S_G = \text{PreRet}(G, q, \text{Pr}_\phi), \quad A_{\text{cand}} = \{e \in S_G \mid \text{Pr}_\psi(e \in A_q \mid q, S_G) > \theta\}, \tag{4.1}$$

where PreRet denotes a preliminary subgraph retrieval, A_{cand} denotes the candidate answers, and θ is a confidence threshold for determining the answer set.

This work focuses on augmenting the subgraph, including the preliminary retrieved subgraph and also the final retrieval subgraph i.e. re-ranking phase. We argue that including triplet facts stored in LLMs (act as a reasoner and retriever) will enhance the consequent reasoning stage, i.e., the ideal subgraph S_G^* for question q is formed by a minimal set of evidence facts.

Given an appropriate similarity measure sim over graphs and questions, S_G^* should be the most similar subgraph to question q, i.e.,

$$\arg\max_{S_G \subset G} \text{Pr}_\phi(S_G|q) = \arg\max_{S_G \subset G} \text{sim}(S_G, q). \tag{4.2}$$

This subgraph of KG will be provided to the consequent answer reasoning.

Definition 3: Query Parsing or Question Analysis

Query Parsing. Query parsing is the process of understanding and interpreting a natural language query q to identify the topic entities $e \in \mathscr{E}$, topic relations $r \in \mathscr{R}$, and any constraints or conditions specified. This step involves natural language processing techniques to convert the query into a structured format that can be used to retrieve information from the knowledge base. A solely IR-based KBQA approach usually involves an Entity Linking step to map the topic entity $e \in \mathscr{E}$ of the question into the entity $\varepsilon \in$K stored in the hosted Knowledge Base i.e. Wikidata.

Definition 4: Subgraph Retrieval

Subgraph Retrieval (SR) proposed by Zhang et al. [133] is an algorithm that expands relation paths via a sequential decision process. A retrieved subgraph or candidate subgraph $G_q = (V_q, E_q)$ is a subgraph of the KB that is relevant to a given query q, which might hold the answer candidate A_{cand} to the query/question q. The vertices V_q in the subgraph correspond to the entities and literals that are pertinent to q, while the edges E_q correspond to the relations that interconnect these entities and literals within the context of the query. The process of generating G_q involves identifying and extracting the subset of triples (s,r,o) from K that are most relevant to q, ensuring that G_q effectively captures the necessary information to address the query.

Definition 5: Answer Generation

Answer Generation. Answer generation involves using the information contained in the retrieved subgraph G_q to formulate a precise and correct answer to the query. This step requires reasoning over the retrieved data to construct an appropriate response that satisfies the query's requirements. There are mainly two types of such generators: entity ranking generators and text generators.

1. **Entity ranking generator** which ranks the entities to obtain top-ranked entities as predicted answers.

2. **Text generator** which generates free text answers with vocabulary V.

This module can be formalized as:

$$\tilde{A}_q = \text{Answer Generation}(s^{(n)}, G_q, V),$$

where $s^{(n)}$ denotes the reasoning status at the last step. In the entity ranking paradigm, the entities contained in G_q are candidates for answer prediction $\tilde{A}_q$. In many cases, $\tilde{A}_q$ is obtained by selecting the entities with a score larger than the pre-defined threshold, where the score is derived from $s^{(n)}$. In the text generation paradigm, the answers are generated from vocabulary V as a sequence of tokens. During training, the objective of the entity ranking generator is usually to rank the correct entities higher than others in G_q. In comparison, the text generator is usually trained to generate gold answers (name of correct entities).

Problem Statement

The goal of the information retrieval-based knowledge base question-answering (IR-KBQA) task is to develop a system that can accurately and efficiently answer questions by leveraging the structured information contained within a knowledge base. Given a natural language query q, the system should:

1. **Question Analysis**: Identify the topic entities, relations, and any constraints or conditions specified.

2. **Subgraph Retrieval**: Extract the most relevant subgraph G_q from the knowledge base K that contains the necessary information i.e. answer candidate A_{cand} where $A_{\text{q}} \subseteq A_{\text{cand}}$ to answer the query.

3. **Generate the Answer**: Use the information in G_q to formulate a precise and correct answer to the query.

The challenge lies in the accurate parsing of the query i.e. the Entity Linking model, efficient retrieval of the pertinent subgraph i.e the Retrieval model, and precise generation of the answer i.e. the Reasoner model, ensuring that the system can handle diverse and complex queries while maintaining high performance and scalability.

4.2 Our Proposed Method

4.2.1 General Pipeline Overview

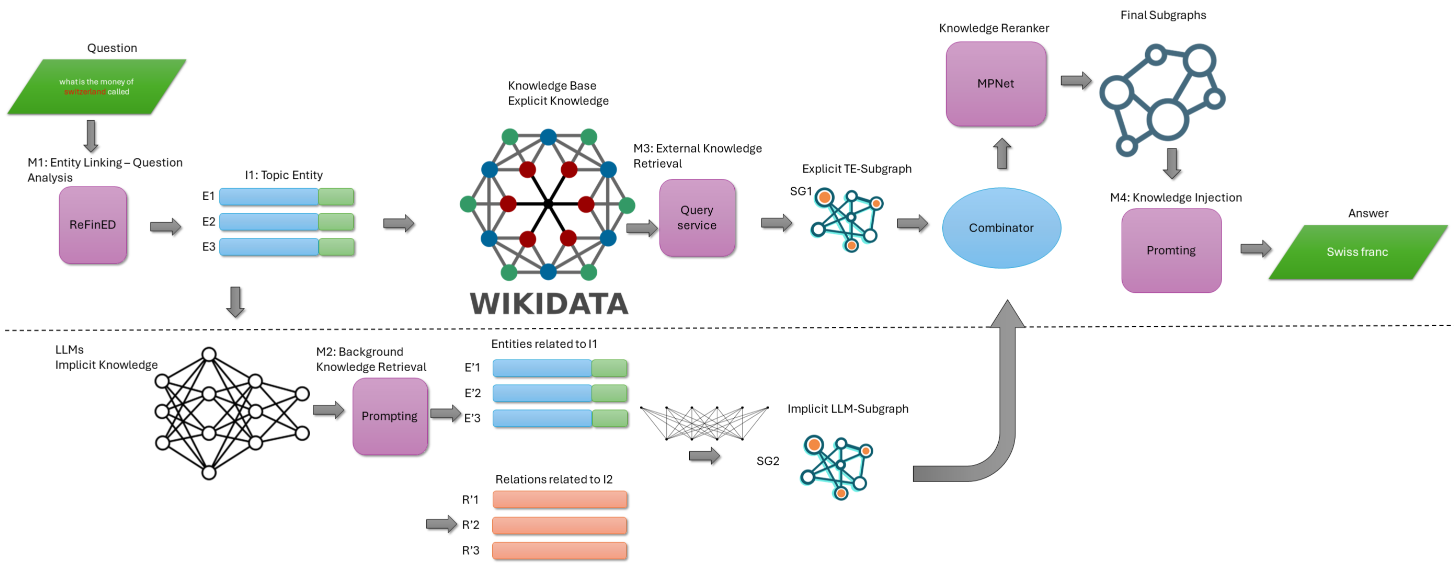

Fig. 4.1: Our proposed KBQA pipeline

In this section, we present our approach which is a pipeline for KGQA with LLMs, inspired by the works of KAPING [6]. Our pipeline includes four main components, including a short description of their respective inputs, processes, and outputs:

1. **Entity Linking - Question Analysis**
   - Input: User question.
   - Process: Extract the topic entity (entities) from the question.
   - Output: List of topic entities related to the question.
2. **Background Knowledge Retrieval**:
   - Input: User question.
   - Process: Generate an LLM-Subgraph based on the user question in a 3-step process by getting relevant entities (that might be related to the topic entity), getting relevant relations and finally assembling into a set of triples.
   - Output: Background knowledge triples, coined as LLM-Subgraph.
3. **External Knowledge Retrieval**:
   - Input: List of entities.
   - Process: Extract all triples/subgraphs related to the entities from an external knowledge graph database (1-hop).
   - Output: External knowledge triples, coined as External-Subgraph.
4. **Knowledge Injection**:
   - Input: Background knowledge triples, external knowledge triples, and user question
   - Process: Combine the background triples and external triples, fuse two subgraphs (LLM-based and External-based) via a Combinator, and then rerank them. The reranked triples are then used as context along with the user question in a prompt, the prompt will then be fed into a large language model to get an answer.
   - Output: Final answer, in textual string

4.2.2 Entity Linking - Question Analysis

This step involves the identification and extraction of the primary topic entity from the user's initial question. To accomplish this task, we employ the use of the ReFinED entity linking system, as documented in the research paper by Ayoola et al. in 2022 [5].

ReFinED [5], which stands for Representation and Fine-grained typing for Entity Disambiguation, is an advanced entity linking system developed by Amazon Science, designed to efficiently and accurately link entity mentions found in documents to their corresponding entities in Wikipedia or Wikidata, which collectively contain over 30 million entities. This system stands out due to its combination of accuracy, speed, and scalability, enabling it to handle web-scale datasets with significantly higher accuracy and at a much lower cost compared to traditional approaches.

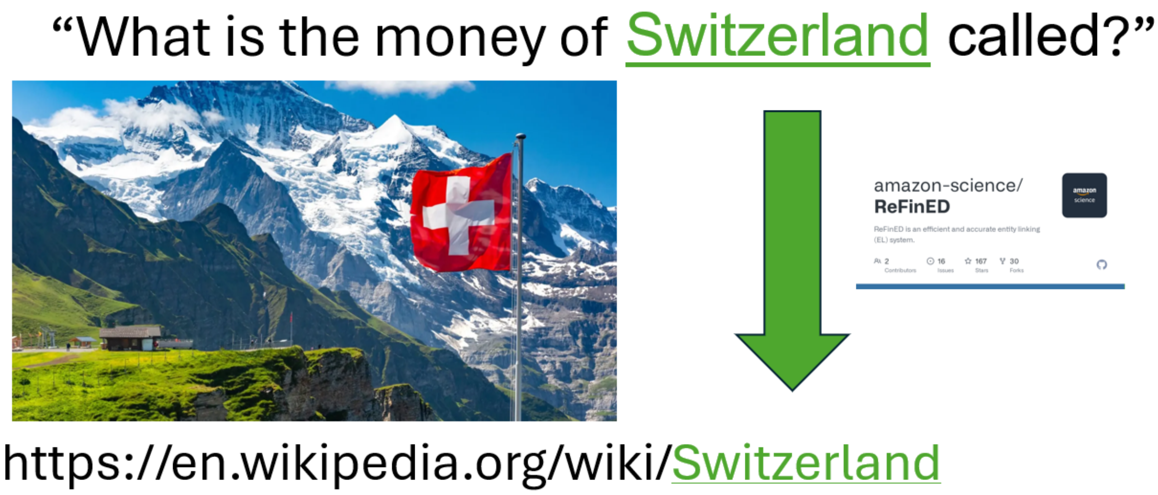

Fig. 4.2: Demonstration of the Question Analysis process of our method

Our decision to use ReFinED was influenced by its impressive performance as a zero-shot entity linking model. It's also the SOTA method on end-to-end Entity Linking (EL) downstream tasks in the grand scheme of KBQA. This means that ReFinED can accurately link entities even when it has not been previously exposed to them during its training phase. Additionally, ReFinED's low-resource requirements make it a practical and efficient choice for our implementation. For our implementation, we utilize these settings for the ReFinED system:

```
extractor = Refined.from_pretrained(
    model_name="questions_model",
```

```
    entity_set="wikidata",
)
```

This configuration allows us to leverage the pre-trained "questions_model" version of ReFinED, which has been specifically optimized for short, lowercase question text. The entity set "wikidata" allows us to query the Wikidata KG [112] as well as Freebase KG [11] with some mapping work, which will be discussed later.

Consider an example, we have a user query "what is the money of switzerland called". After running the query through this Entity Linking - Question Analysis step, we will obtain the entity "Switzerland".

4.2.3 Background Knowledge Retrieval

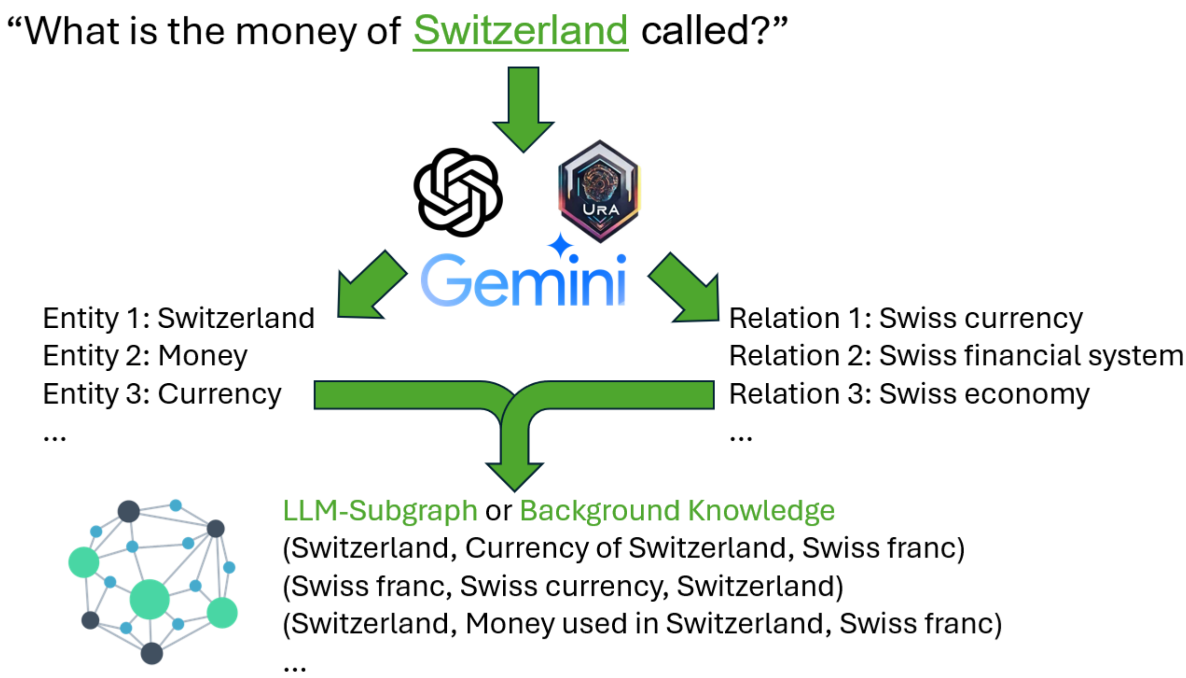

Fig. 4.3: Background LLM-Subgraph for the question "What is the money of Switzerland called?"

In order to retrieve background knowledge i.e. LLM-background subgraph from the LLM, we have devised a set of 3 prompts. The first prompt in our series is dedicated to the identification and generation of all entities that are related to the user's question. This includes the topic entity, which is the primary focus of the query. The prompt is structured as follows:

```
'''
Given a question, generate all entities related to the question.

Question: {query}
Entities:
'''
```

Moving on to the second prompt, its purpose is to extract all relations that are associated with the original question, provided that such relations exist. This step is crucial in understanding the context and the connections between different entities. Here is how the second prompt is formatted:

```
'''
Given a question, generate all relations related to the question.

Question: {query}
Relations:
'''
```

The final prompt in our sequence is responsible for bringing together the generated entities and relations, and combining them with the user's question to create a set of triples. These triples, in the format of (subject, relation, object), will form the structure of a sub-graph. Here's the format of the final prompt:

```
'''
Given a question along with entities and relations, assemble triples
(subject, relation, object) for a knowledge graph.

Question: {query}
Entities: {entities}
Relations: {relations}

Triples:
'''
```

The resulting triples from the final prompt will serve as the foundation of the background knowledge for the knowledge injection component.

Using the example from before, running the query "what is the money of switzerland called" through the first prompt gives us these entities:

```
- Switzerland
- Money
- Currency
```

Putting the question through the second prompt yields us these relations:

```
- Swiss currency
- Swiss financial system
- Swiss economy
- Swiss banking
- Swiss financial markets
- Swiss central bank
<truncated>
```

And finally combining the user query with background entities and relations gives us this knowledge graph via the third prompt:

```
- (Switzerland, Currency of Switzerland, Swiss franc)
- (Swiss franc, Swiss currency, Switzerland)
- (Switzerland, Money used in Switzerland, Swiss franc)
- (Swiss franc, Legal tender in Switzerland, Switzerland)
- (Switzerland, Monetary unit of Switzerland, Swiss franc)
```

4.2.4 External Knowledge Retrieval

In order to effectively retrieve external knowledge i.e. explicit subgraph from the vast expanse of information available, we will employ a methodical approach that leverages the topic entity or entities obtained from the preceding entity linking step. This process involves querying the comprehensive Freebase KG by utilizing the SPARQL query language or querying Wikidata KG via their API [121]. As the process of querying Wikidata KG is simple, we will mainly focus on querying the Freebase KG.

Firstly, we transform the entities derived from the entity linking step into their corresponding Freebase identifiers. This conversion is facilitated by the Wikidata API, a service that provides a mapping between entity names and their respective unique identifiers in the Freebase KG.

Once we have obtained these Freebase IDs, we incorporate them into a predefined

SPARQL query template. This template is designed to retrieve all relevant triples associated with the given entity. The template is as follows:

```
'''
PREFIX ns: <http://rdf.freebase.com/ns/>
SELECT ?subject ?predicate ?object
WHERE {
{ ?subject ?predicate ns:{entity} } UNION
{ ns:{entity} ?predicate ?object }
}
'''
```

Subsequently, we examine all of the Freebase IDs that appear within the retrieved triples and convert these IDs back into their respective entity names via the API. After that, we transform all of the triples into a predefined text format (subject, predicate, object) to prepare for the last step. Running the topic entity "Switzerland" as an example from the entity linking step returns these triples from Freebase KG (after converting IDs to entity names):

```
(Swiss franc, replaced by, WIR franc),
(Switzerland, member of, Confederation),
(Old Swiss Confederacy, replaced by, Switzerland),
<truncated>
```

4.2.5 Knowledge Injection or Re-ranking

In the final step of our process, we concatenate the two sub-graphs (LLM-based and External-based) and then utilize MPNet, a sophisticated text embedding model outlined in the study by Song et al. in 2020 [94], to convert both the triples and the user's question into a shared embedding space. This transformation allows us to compute the similarities between the triples and the user's question, thereby enabling us to identify the most relevant triples for answering the user's query.

Once we have pinpointed the top-K triples, we integrate them with the user's question into a crafted prompt. The structure of this prompt is as follows:

```
"""
Below are facts in the form of the triple meaningful to answer the questions
(Short answer, explanations not needed, output N/A if you can't provide an answer)
```

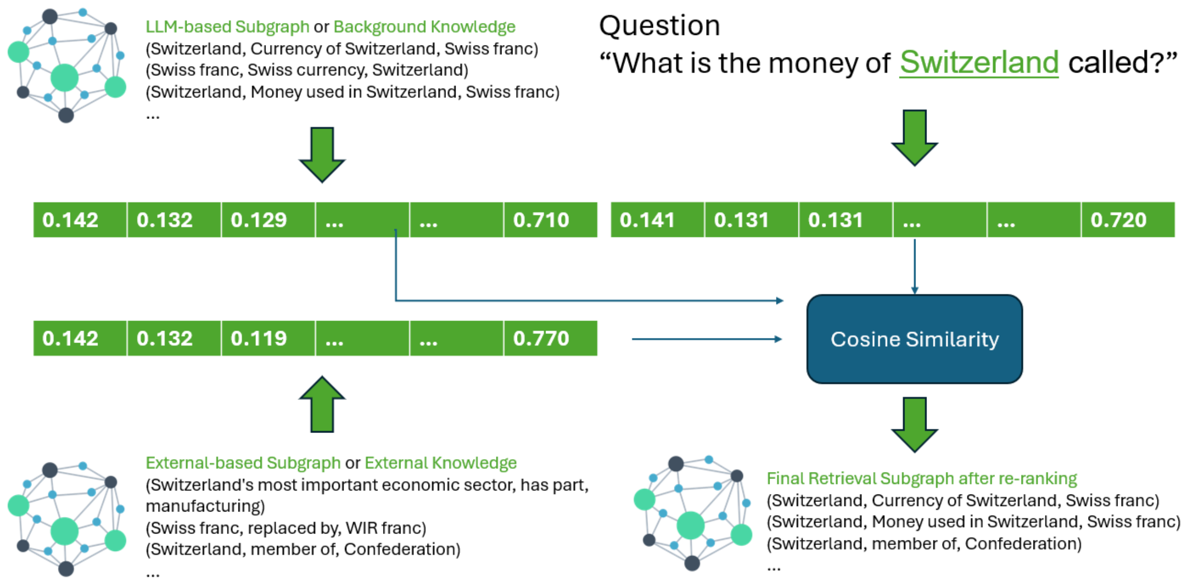

Fig. 4.4: Knowledge Injection the Reasoner with the Final Retrieval Subgraph i.e re-ranking subgraph/triple facts

```

{reranked_triples}


Question: {question}
Answer:
"""
```

Subsequently, we feed this carefully constructed prompt into a large language model (LLM) to generate a response to the user's question. This approach ensures that our system not only comprehends the user's question but also leverages the most relevant information from the triples and background knowledge to get an accurate answer. Combining all intermediate results for the example "what is the money of switzerland called" gives us the final prompt in this format:

```
Below are facts in the form of the triple meaningful to answer the questions
(Short answer, explanations not needed, output N/A if you can't provide an answer)

(Switzerland, Currency of Switzerland, Swiss franc),
(Swiss franc, Swiss currency, Switzerland),
(Switzerland, Money used in Switzerland, Swiss franc),
(Swiss franc, Legal tender in Switzerland, Switzerland),
(Switzerland, Monetary unit of Switzerland, Swiss franc),
(Swiss Federal budget, country, Swiss Confederation)
```

```
Question: what is the money of switzerland called
Answer:
```

5

Experiments

Having outlined our proposed method for an improved IR-based Knowledge Base Question Answering (KBQA) system in the previous chapter, we now transition to the experimental phase of our research. This chapter is dedicated to the rigorous evaluation of our approach, where we will detail the experimental setup, including the datasets, evaluation metrics, and baseline models used for comparison. By conducting comprehensive experiments, we aim to validate the effectiveness of our proposed methodologies and demonstrate their superiority over existing systems.

Through a series of carefully designed experiments, we will showcase how our synergistic approach of combining external retrieval sources with parametric knowledge retrieval from Large Language Models (LLMs) enhances the system's performance. We will provide a thorough analysis of the results, focusing on key performance indicators such as Accuracy, Hits@1, and F1 scores, to substantiate our claims of improvement. This chapter will also include an ablation study to identify the contributions of different components of our system and to understand the underlying reasons behind our method's success.

The insights gained from these experiments will not only confirm the viability of our proposed methods but also highlight potential areas for further refinement and development. Through this meticulous evaluation, we aim to pave the way for future innovations in the field of KBQA.

5.1 Experimental Setups

We explain datasets, models, metrics, and implementations. For additional details, see Appendix.

5.1.1 Datasets

Following the work on KGQA (KV-MeM [76], EmbedKGQA [88], KAPING [6], Li et al. [65]) We evaluate our IR-based KBQA method on two Knowledge Graph Question Answering (KGQA) datasets, namely WebQuestionsSP and Mintaka.

- **WebQuestionSP (WebQSP)**: The WebQuestionsSP (WebQuestions Semantic Parses) dataset is a collection of question-answer pairs designed for evaluating the performance of systems in answering questions about the world using a structured knowledge base, introduced by Yih et al. in 2016 [128]. The dataset comprises 4,737 questions for which full semantic parses in SPARQL queries are available, along with "partial" annotations for another 1,073 questions. For this experiment, we will use 1215 questions from the test set.

- **Mintaka**: The Mintaka dataset is a comprehensive resource for developing and evaluating end-to-end question-answering models, particularly those designed to handle complex, natural, and multilingual queries. Introduced by Sen et al. in 2022 [90], Mintaka comprises 20,000 question-answer pairs, meticulously selected and annotated with Wikidata entities. For this experiment, we will sample 1000 English questions from the test set.

It is important to note that when working with the WebQSP dataset, we will be utilizing a smaller version of the Freebase Knowledge Graph (KG). This is following the work of EmbedKGQA [88] due to resource limitations.

5.1.2 Evaluation Metrics

Generation

As for the metrics we will be using to evaluate our model generation, we have chosen Exact Match Accuracy as our primary measure, following the work of Sen et al. [90], Mavi et al. [73], and Baek et al. [6]. This metric will be used to determine if the answer provided by the LLM contains one of the answer entities. In other words, the model will be considered accurate if its response includes the correct entity, even if it also includes additional, potentially incorrect information. This approach allows us to assess the model's ability to identify and include the correct entities in its responses, while also acknowledging that it may sometimes provide additional, unnecessary information.

We also need to take into consideration the alias of the answer entity when choosing the reasoner/reader/generator model as LLMs like ChatGPT [14]. This is unlike the fine-tuned seq2seq model over the test split like most variants of T5. The LLMs offered through APIs like ChatGPT, which will not focus on generating the answer that is concise and matched string-by-string with the answer entity on the datasets, will likely elaborate on and extend the answer verbosely to encompass better user experience.

Retrieval

Aside from accuracy, we also use F1 and Hits@1 metrics to measure the performance of our retriever/sub-graphing process following the work of Baek et al. [6].

5.1.3 Baseline Methods for Comparison

For the first baseline, we use the I/O Prompt Technique, which prompts the LLM with no external or internal knowledge, to test the sole ability of the models. In this approach, the LLM will not be provided with any context or information related to the knowledge graph or the specific domain of the questions. This baseline method is designed to evaluate the LLM's inherent ability to generate responses based solely on its pre-existing training, without any additional input. The results from this baseline will provide a benchmark to compare the performance of more sophisticated methods that incorporate external or internal knowledge.

As for the second baseline, we will re-implement the works of KAPING [6] in our own environment. By re-implementing KAPING in our own environment, we aim to establish a comparative baseline that leverages external knowledge from the knowledge graph. The details of implementation will be discussed in the following section.

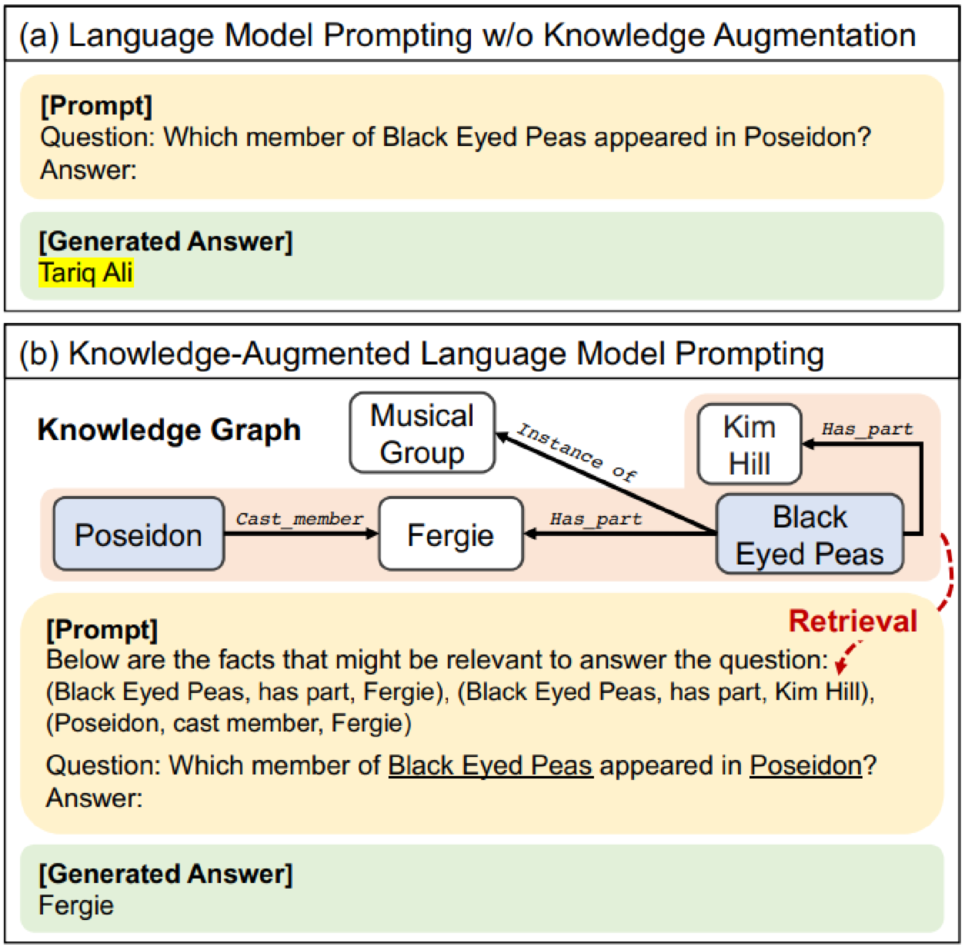

Fig. 5.1: Baseline methods from the work of Baek et al. [6]

5.1.4 Large Language Models

For this experiment, we use 3 large language models and 1 smaller language models. These include Gemini 1.0 Pro [104], GPT-3.5-Turbo [82] and GemSUra-7B [108]. GemSUra-7B is a fine-tuned large language model specifically designed for Vietnamese language processing, developed collaboratively by researchers from Ho Chi Minh University of Technology (HCMUT) - Vietnam National University HCMC, and Stanford University. This model leverages Vietnamese articles from Wikipedia and other sources, aiming to enhance the quality of Vietnamese language models. And our small language model is a fine-tuned variant of XLM-RoBERTa[1] on MRCQuestionAnswering task [21]. For the XLM-RoBERTa, because it's not designed for seq2seq generation or retrieval-main task but rather an extractive question-answering, we augmented the Gemini' background knowledge into the pipeline using XLM-RoBERTa as a reasoner and report the

[1]The detail is available at https://huggingface.co/nguyenvulebinh/vi-mrc-base

result.

5.1.5 Implementation Details

We implemented the proposed method with Python 3.10.11, PyTorch 2.3, and CUDA 12.1. The results are obtained on a gaming laptop with AMD Ryzen 5 2600 CPUs and NVIDIA RTX 4050 GPUs[2]. We set all of the LLMs' temperatures to default (usually 0.7).

5.2 Evaluation Results and Further Analysis

5.2.1 Main Results (RQ1)

Dataset And Method	Gemini 1.0 Pro	GPT-3.5-Turbo	GemSUra-7B	XLM-RoBERTa
Mintaka W/ No knowledge	53.4	37.6	30.8	0
Mintaka W/ KAPING	14.0	24.3	21.0	4.4
Mintaka W/ Ours	**29.0**	**27.4**	**29.0**	**14.8**
WebQSP W/ No knowledge	54.1	35.0	39.0	0
WebQSP W/ KAPING	32.2	32.5	37.3	14.6
WebQSP W/ Ours	**43.7**	**33.7**	**45.6**	**21.8**

Table 5.1: Benchmark of our method versus the baseline methods for generated answer. "W/ KAPING" is the original pipeline re-implementation.

Dataset And Retriever Type	Hits@1	F1
KAPING (No background) w/ Mintaka	10.5	7.9
Gemini (KAPING + Background) w/ Mintaka	**17.3**	**9.1**
GPT-3.5-Turbo (KAPING + Background) w/ Mintaka	15.2	8.5
GemSUra-7B (KAPING + Background) w/ Mintaka	9.8	7.1
KAPING (No background) w/ WebQSP	18.2	11.4
Gemini (KAPING + Background) w/ WebQSP	**25.2**	**17.3**
GPT-3.5-Turbo (KAPING + Background) w/ WebQSP	19.6	14.1
GemSUra-7B (KAPING + Background) w/ WebQSP	18.3	11.51

Table 5.2: Benchmark of our method versus the baseline methods for retriever/sub-graph performance

The results from the tables indicate that there is significant room for improvement in all of the implementations. However, it's important to note that these are complex tasks

[2]The implementation is available at https://github.com/DerpLEL/knowledge-base-project

and the scores achieved are not insignificant. A note that for our XLM-RoBERTa result, we opt to use Gemini's background knowledge when applying our method. This is due to the fact that XLM-RoBERTa and other smaller-sized models are not reliable when it comes to generating background, parametric knowledge.

It is very noticeable that the no-knowledge baseline consistently outperforms both KAPING and our proposed method. We noticed during our experiments that when insufficient triples are presented along with our given prompts, models tend to answer "N/A" for not enough information. This is most likely due to our 1-hop implementation, which might not be enough for 2-hop or 3-hop questions. The retriever/sub-graphing results highlight this fact quite prominently.

Our method, when applied to the Gemini 1.0 Pro model, has shown some improvements, particularly on the WebQSP dataset, where it achieved a score of 43.7. This is a substantial improvement over KAPING, despite not reaching the levels of the no-knowledge baseline methods. For GPT-3.5-Turbo however, our method does not translate to any significant improvement over the KAPING method, as GPT-3.5-Turbo has been trained to reject certain questions and prompts to avoid hallucinations, including our set of background knowledge retrieval prompts. The effects of our method applied to the GemSUra-7B model similarly to Gemini 1.0 Pro. As for our XLM-RoBERTa's result, it is observed that without context, the model is incapable of providing the answer, and shows improvements when applied with our method.

On the Mintaka dataset, our method follows a similar trend to the WebQSP dataset, with our method showing gains over KAPING. Note that the accuracy scores for Mintaka are overall lower compared to WebQSP, due to the various types of questions contained within the dataset.

We deduce several flaws from our experiments:

- **One-hop retrieval constraint**: Since our retrieval step only does one-hop retrieval, the triples we obtain may not be sufficient for answering questions requiring 2-hop or 3-hop retrieval, which are abundant in the 2 datasets.

- **Cut down version of Freebase KG**: In the case of the WebQSP dataset, as stated, we are using a smaller version of the Freebase KG. Hence, the knowledge graph may not contain all of the triples relevant to the user question, leading to subpar performance.

- **Limited dataset choices**: Our experiments only involve two datasets. This may not accurately reflect the pipeline's performance when applied to other knowledge graphs or question domains.

Considering these flaws, we devise some recommendations to improve this system:

- **Use multi-hop retrieval**: To address the limitation of one-hop retrieval, the system should be improved to support multi-hop retrieval. This will allow the system to retrieve a chain of related entities and relations, which can be beneficial for answering complex questions that require 2 or more retrieval hops.

- **Run the full Freebase KG**: To improve the performance, we can gather resources to query on the original Freebase KG dump, provided by Google API i.e. Wikidata end-point. This will increase the chances of the knowledge graph containing the relevant information to the user's question.

- **Consider Semantic Parser-based Approach**: As mentioned in the chapter of Related Works, the current SOTA methods on many datasets that we explored i.e. WebQSP, Mintaka, usually are Semantic Parser (SP-based) methods [70, 131, 92]. Some methods [131] claim that their approach is a fusion of both IR-based and SP-based methods but usually, it is a weak semi-IR-based component coupled with a strong core SP-based mechanism.

- **Experiment on more datasets**: Expanding the scope of our experiments to include a broader range of datasets is necessary for comprehensive evaluation and generalization of our method's performance. Incorporating datasets with varying characteristics, such as different domains, question types, and levels of complexity, will provide a more thorough assessment of our method's efficacy across diverse scenarios. Additionally, considering datasets with larger and more diverse knowledge graphs will facilitate a better understanding of our method's scalability and adaptability.

5.2.2 Ablation Study (RQ2)

For our ablation study, we will consider one case with Gemini 1.0 Pro as our chosen large language model, and the chosen dataset is the WebQSP dataset. We will cover four cases: No knowledge, KAPING (No background), Background knowledge only, and Full (KAPING + background).

Cases	Overall accuracy
No knowledge	**54.1**
KAPING (No background)	32.2
Background only	48.2
Full (KAPING + background)	43.7

Table 5.3: Ablation study table on reducing different components

From the results in Table 5.3, it's evident that the model's performance varies significantly across the four cases:

- **No knowledge**: Achieves an accuracy of 54.1%, serving as a baseline for our study.
- **KAPING (No background)**: i.e. IR-based Subgraph only experiences a significant drop to 32.2%, indicating that the absence of background knowledge adversely affects the model's accuracy when relying solely on KAPING.
- **Background knowledge only**: Shows an improvement with an accuracy of 48.2%, demonstrating the positive impact of background knowledge.
- **Full (KAPING + background)**: Results in an accuracy of 43.7%, which, while not surpassing the baseline or the background-only condition, is notably higher than KAPING (No background).

The main sticking point here is that the Full case (KAPING + background) outperforms the KAPING (No background) case by a substantial margin, achieving an accuracy of 43.7% compared to 32.2%. This indicates that the integration of background knowledge significantly enhances the performance of the model when KAPING alone falls short. While background knowledge on its own contributes positively to the model's performance, the combination of KAPING and background knowledge is crucial for leveraging the strengths of both approaches, thereby mitigating the limitations observed when either component is used in isolation. This highlights the importance of a comprehensive approach that incorporates both external knowledge and LLM's background knowledge to optimize the model's accuracy on the WebQSP dataset.

5.2.3 Comparison with different LLMs in generation phase (RQ3)

To demonstrate the effectiveness of using various large language models (LLMs) as reasoners in an information retrieval-based knowledge base question answering (KBQA) system, we replaced the LLMs in the Reasoner module (right after receiving the final retrieval subgraph) and evaluated Gemini 1.0 Pro, GPT-3.5-Turbo, GemSUra-7B (recorded on 20/May/24), and a fine-tuned version of XLM-Roberta. The XLM-Roberta model was fine-tuned on the MRC Question Answering task using datasets such as SQuAD 2.0, mailong25, UIT-ViQuAD, and MultiLingual Question Answering. We compared their performance using Exact Match Accuracy, following the methodology of Baek et al. [6]. Our results show that Gemini achieved an accuracy of 43.7, GPT-3.5-Turbo 33.7, GemSUra-7B 45.6, and XLM-Roberta (augmented with the final subgraph retrieved from Gemini for simplicity) 21.8. These findings highlight the varied performance across different models and the potential benefits of fine-tuning and model augmentation for improving accuracy in KBQA tasks.

Furthermore, we conducted a comparative analysis on two datasets: WebQSP and Mintaka. All results were reported using the full pipeline (both IR-based retrieval subgraph and LLM-base background subgraph). The results, as detailed in Table 5.4, show that GemSUra-7B performed the best overall on WebQSP with an Exact Match Accuracy of 45.6, while Gemini excelled on the Mintaka dataset with an accuracy of 44.1. Notably, XLM-Roberta demonstrated lower performance across both datasets even though it's fine-tuned on the MRCQuestionAnswering task.

Reasoner	WebQSP	Mintaka
Gemini	43.7	**44.1**
ChatGPT	33.7	27.4
GemSURA	**45.6**	34.5
XLM-Roberta	21.8	14.8

Table 5.4: Comparison (Exact Match Accuracy) of different LLMs on full-pipeline

These evaluations underscore the importance of selecting appropriate LLMs for KBQA systems and suggest that certain models, particularly those like GemSUra-7B and Gemini 1.0 Pro, may offer substantial advantages depending on the dataset and specific requirements of the task.

In addition to the evaluations on the WebQSP dataset, we extended our analysis to the Mintaka dataset to assess the performance of the full pipeline. The results on the Mintaka dataset further corroborate the findings from WebQSP, indicating consistent performance trends across different types of KBQA tasks.

5.2.4 Is the subgraph retrieved from LLMs 1-hop or n-hop? (RQ4)

We further investigate whether the subgraphs retrieved from Large Language Models (LLMs) are 1-hop or n-hop. The distinction between 1-hop and n-hop subgraphs is crucial for understanding the efficiency and effectiveness of our Information Retrieval-based Knowledge-Based Question Answering (IR-based KBQA) system. A 1-hop subgraph refers to the immediate connections from a given node, whereas an n-hop subgraph extends to connections that are n steps away. As laid down in the previous section, multi-hop subgraph retrieval is computationally inefficient and costly for most ideal IR-based KBQA runnable systems. By conducting an ablation study, we aim to determine how the depth of the retrieved subgraph by LLMs impacts the performance of our KBQA system.

Specifically on 5.2, we compare the Hits@1 and F1 of the retrieved subgraph using 1-hop subgraphs (i.e. only KAPING) versus those generated using $(n \geq 1)$-hop subgraphs (we denoted it as $(n \geq 1)$-hop because apart from the IR-based external subgraph $S_G^* \subseteq G$ achieved from following the work of Baek et al. [6], we also generated and augmented, from the LLMs, an additional subgraph $S_G^{'} \subseteq \text{Think-first}(q, Reasoner)$ for question q, importantly noted that this is totally independent of the knowledge base $G \subseteq K$ and only depend on LLMs powering the Reasoner module i.e. Gemini, GemSURA. This analysis helps us to identify the optimal retrieval strategy, ensuring that our system can effectively leverage the knowledge encoded in LLMs to provide accurate and comprehensive answers.

As shown in Table 5.2, Hits@1 and F1 scores indicate that 1-hop subgraphs (i.e., only KAPING) generally still offer a more computationally efficient solution while maintaining robust performance. In contrast, $(n \geq 1)$-hop subgraphs retrieval can potentially provide richer context (by a whooping 64.76% in Mintaka and 38.46% on WebQSP reported on comparing Hits@1 of only KAPING (Baek et al.[6] original work) and Gemini (KAPING + background); for F1 it is an increase of 15.18% for WebQSP and 51.75%) with only an additional fraction of cost by incorporating extended connections from the probabilistic model that is independent of the closed Knowledge Base i.e. Freebase, Wikidata, etc. and probabilistically closely to the question q. In conclusion, because LLMs is typically a probabilistic black box, we still need to do more extensive research to test out this hypothesis

5.2.5 Comparison with different IR-based KBQA method (RQ5)

Method	Hits@1	F1	Accuracy
KAPING[6]	40.27	43.15	*73.89*
ChatKBQA[70]	**86.4**	**83.5**	**77.8**
Tiara[92]	*75.2*	*78.9*	-
GMT-KBQA[38]	-	76.6	73.1
Ours (Gemini + KAPING + Background)	10.5	7.9	43.7

Table 5.5: WebQSP

Method	Hits@1	F1	Accuracy
KAPING[6]	7.5	**41.83**	-
Mintaka baseline (T5 for CBQA)[90]	38.0	-	-
Li et al.[65]	**61.7**	-	-
Ours (Gemini + KAPING + Background)	17.3	9.1	**44.1**

Table 5.6: Mintaka

It is evident from Table 5.5 that our method, combining Gemini, KAPING, and background information, currently lags behind the SOTA methods in the field. Specifically, ChatKBQA outperforms other methods significantly, achieving the highest Hits@1, F1, and Accuracy. Our method achieved a Hits@1 of 10.5, an F1 score of 7.9 (due to our limitation of exploring only 1-hop subgraph), and an Accuracy of 43.7, which are considerably lower than the leading results (we don't use reasoner fine-tune on the train split of respect datasets). Note that our method still surpasses the original pipeline KAPING in Hits@1 in Mintaka by a whooping 130.66%.

Despite these results, it is important to consider several factors that demonstrate the potential of our approach and justify its inclusion in this comparative study:

Innovation and Integration: Our method uniquely integrates the Gemini model with the KAPING framework and additional background knowledge. This approach, while still in its early stages, showcases a novel direction that could be further refined and optimized.

Scope for Improvement: The current implementation represents a preliminary attempt. With additional fine-tuning, optimization, and the incorporation of more advanced retrieval techniques, there is substantial room for improvement. For instance, enhancing the training datasets and using more sophisticated model architectures could significantly boost performance.

Flexibility and Extensibility: Our method's modular design allows for easy updates and integration with newer models and techniques as they become available. This flexibility is a crucial advantage, as it ensures that the system can continuously evolve and improve.

Comparative Baseline: By providing a comparative baseline, our research contributes to the broader understanding of the strengths and weaknesses of various approaches. This baseline can be valuable for future researchers aiming to build upon or improve our methods.

In conclusion, while our current results fall short of the SOTA, the innovative aspects of our method, combined with its potential for future enhancement, underline its value and relevance in the ongoing development of KBQA systems. We believe that with further research and development, our approach can achieve competitive performance and contribute significantly to the field.

6

Conclusion

In this concluding chapter, we encapsulate the key findings and insights drawn from the preceding discussions on knowledge graph integration. Building upon the foundation laid in the preceding chapters, this section synthesizes the core outcomes and implications of the research endeavors. Our aim is to distill the collective significance of the explored methodologies and insights, thereby provide a coherent understanding about our work. We also aim to map out plausible directions for further inquiry and development.

6.1 Possible improvements

There are several potential enhancements and refinements of our proposed knowledge graph-based question-answering approach. To begin with, since our retriever are limited to one-hop only, delving into more research regarding multi-hop reasoning, especially with LLMs, is a path with much potential as it could improve our retriever's performance greatly.

Secondly, because of our limited dataset choices, we can't conclude much from our experiments despite the improvements over the baseline. As such, expanding our dataset and conducting more extensive experiments would provide a stronger foundation for evaluating the effectiveness of our approach. By incorporating a larger and more diverse set of queries, we can better understand the strengths and limitations of our system under various conditions.

Lastly, we could also consider incorporating a more sophisticated ranking mechanism for the retrieved answers. Currently, our system relies on a simple similarity score between the query and the triples. However, by leveraging machine learning techniques, we could potentially train a model to more accurately predict the relevance of a triple to a given query.

6.2 Future plans

The forthcoming objectives within our research entail an expansion of methodologies to further refine our framework for question-answering on knowledge graph with large language models. A primary focus will involve a comparative analysis of diverse approaches, aiming to implement and assess various methods conducive to maintaining a knowledge graph efficiently.

A pivotal aspect of our future endeavors encompasses the development of a specialized case study centered on academic affairs. This focal study will entail the meticulous curation and processing of academic documents and pertinent data, utilizing our refined pipeline to construct a specialized knowledge graph. The resultant knowledge graph will be seamlessly integrated into a designated KG DBMS for storage and accessibility. Subsequently, the integration with a chat interface will be implemented and rigorously evaluated to gauge its efficacy and usability.

This comprehensive approach will not only advance our understanding of knowledge graph integration with large language models but also validate the practical applications of our research through tangible experimentation and evaluation within the academic domain.

References

[1] Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. "On the surprising behavior of distance metrics in high dimensional space". In: *Database Theory—ICDT 2001: 8th International Conference London, UK, January 4–6, 2001 Proceedings 8*. Springer. 2001, pp. 420–434.

[2] Garima Agrawal et al. "Building Knowledge Graphs from Unstructured Texts: Applications and Impact Analyses in Cybersecurity Education". In: *Information* 13 (Nov. 2022), p. 526. DOI: 10.3390/info13110526.

[3] Garima Agrawal et al. *Can Knowledge Graphs Reduce Hallucinations in LLMs? : A Survey*. 2024. arXiv: 2311.07914 [cs.CL].

[4] Ismail Aliyu, A.F. Kana, and Salisu Aliyu. "Development of Knowledge Graph for University Courses Management". In: *International Journal of Education and Management Engineering* 10 (Apr. 2020), pp. 1–10. DOI: 10.5815/ijeme.2020.02.01.

[5] Tom Ayoola et al. "ReFinED: An Efficient Zero-shot-capable Approach to End-to-End Entity Linking". In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*. Ed. by Anastassia Loukina, Rashmi Gangadharaiah, and Bonan Min. Hybrid: Seattle, Washington + Online: Association for Computational Linguistics, July 2022, pp. 209–220. DOI: 10.18653/v1/2022.naacl-industry.24. URL: https://aclanthology.org/2022.naacl-industry.24.

[6] Jinheon Baek, Alham Fikri Aji, and Amir Saffari. "Knowledge-Augmented Language Model Prompting for Zero-Shot Knowledge Graph Question Answering". In: *Proceedings of the 1st Workshop on Natural Language Reasoning and*

Structured Explanations (NLRSE). Ed. by Bhavana Dalvi Mishra et al. Toronto, Canada: Association for Computational Linguistics, June 2023, pp. 78–106. DOI: 10.18653/v1/2023.nlrse-1.7. URL: https://aclanthology.org/2023.nlrse-1.7.

[7] Yuntao Bai et al. *Constitutional AI: Harmlessness from AI Feedback*. 2022. arXiv: 2212.08073 [cs.CL].

[8] Maciej Besta et al. "Graph of Thoughts: Solving Elaborate Problems with Large Language Models". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 38.16 (Mar. 2024), pp. 17682–17690. ISSN: 2159-5399. DOI: 10.1609/aaai.v38i16.29720. URL: http://dx.doi.org/10.1609/aaai.v38i16.29720.

[9] Nikita Bhutani, Xinyi Zheng, and H V Jagadish. "Learning to Answer Complex Questions over Knowledge Bases with Query Composition". In: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. CIKM '19. Beijing, China: Association for Computing Machinery, 2019, pp. 739–748. ISBN: 9781450369763. DOI: 10.1145/3357384.3358033. URL: https://doi.org/10.1145/3357384.3358033.

[10] Kurt Bollacker, Robert Cook, and Patrick Tufts. "Freebase: A shared database of structured general human knowledge". In: *AAAI*. Vol. 7. 2007, pp. 1962–1963.

[11] Kurt Bollacker et al. "Freebase: a collaboratively created graph database for structuring human knowledge". In: *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*. SIGMOD '08. Vancouver, Canada: Association for Computing Machinery, 2008, pp. 1247–1250. ISBN: 9781605581026. DOI: 10.1145/1376616.1376746. URL: https://doi.org/10.1145/1376616.1376746.

[12] Antoine Bordes et al. "Translating Embeddings for Modeling Multi-relational Data". In: *Advances in Neural Information Processing Systems*. Ed. by C.J. Burges et al. Vol. 26. Curran Associates, Inc., 2013. URL: https://proceedings.neurips.cc/paper_files/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf.

[13] Tomaz Bratanic. *Knowledge Graphs LLMs: Multi-hop Question Answering*. Published on Medium, 10 min read. June 2023. URL: https://medium.com/neo4j/knowledge-graphs-llms-multi-hop-question-answering-322113f53f51.

[14] Tom Brown et al. "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[15] Shulin Cao et al. "Program Transfer for Answering Complex Questions over Knowledge Bases". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 8128–8140. DOI: 10.18653/v1/2022.acl-long.559. URL: https://aclanthology.org/2022.acl-long.559.

[16] Hung-Ting Chen, Michael Zhang, and Eunsol Choi. "Rich Knowledge Sources Bring Complex Knowledge Conflicts: Recalibrating Models to Reflect Conflicting Evidence". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 2292–2307. DOI: 10.18653/v1/2022.emnlp-main.146. URL: https://aclanthology.org/2022.emnlp-main.146.

[17] Shuang Chen et al. "ReTraCk: A Flexible and Efficient Framework for Knowledge Base Question Answering". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*. Ed. by Heng Ji, Jong C. Park, and Rui Xia. Online: Association for Computational Linguistics, Aug. 2021, pp. 325–336. DOI: 10.18653/v1/2021.acl-demo.39. URL: https://aclanthology.org/2021.acl-demo.39.

[18] Yongrui Chen et al. *Outlining and Filling: Hierarchical Query Graph Generation for Answering Complex Questions over Knowledge Graphs*. 2022. arXiv: 2111.00732 [cs.AI].

[19] Zi-Yuan Chen et al. "UHop: An Unrestricted-Hop Relation Extraction Framework for Knowledge-Based Question Answering". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 345–356. DOI: 10.18653/v1/N19-1031. URL: https://aclanthology.org/N19-1031.

[20] Zhoujun Cheng et al. *Binding Language Models in Symbolic Languages*. 2023. arXiv: 2210.02875 [cs.CL].

[21] Alexis Conneau et al. "Unsupervised Cross-lingual Representation Learning at Scale". In: *CoRR* abs/1911.02116 (2019). arXiv: 1911.02116. URL: http://arxiv.org/abs/1911.02116.

[22] Rajarshi Das et al. "Case-based Reasoning for Natural Language Queries over Knowledge Bases". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Ed. by Marie-Francine Moens et al. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 9594–9611. DOI: 10.18653/v1/2021.emnlp-main.755. URL: https://aclanthology.org/2021.emnlp-main.755.

[23] Tim Dettmers et al. "Convolutional 2D Knowledge Graph Embeddings". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32.1 (Apr. 2018). DOI: 10.1609/aaai.v32i1.11573. URL: https://ojs.aaai.org/index.php/AAAI/article/view/11573.

[24] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for

Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423.

[25] Guanting Dong et al. *Bridging the KB-Text Gap: Leveraging Structured Knowledge-aware Pre-training for KBQA*. 2023. arXiv: 2308.14436 [cs.CL].

[26] Andrew Drozdov et al. "You can't pick your neighbors, or can you? When and How to Rely on Retrieval in the kNN-LM". In: *Findings of the Association for Computational Linguistics: EMNLP 2022*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 2997–3007. DOI: 10.18653/v1/2022.findings-emnlp.218. URL: https://aclanthology.org/2022.findings-emnlp.218.

[27] Zhengxiao Du et al. "GLM: General Language Model Pretraining with Autoregressive Blank Infilling". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 320–335. DOI: 10.18653/v1/2022.acl-long.26. URL: https://aclanthology.org/2022.acl-long.26.

[28] Alexander Dunn et al. *Structured information extraction from complex scientific text with fine-tuned large language models*. 2022. arXiv: 2212.05238 [cs.CL].

[29] Yu Feng et al. "A Pretraining Numerical Reasoning Model for Ordinal Constrained Question Answering on Knowledge Base". In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Ed. by Marie-Francine Moens et al. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 1852–1861. DOI: 10.18653/v1/2021.findings-emnlp.159. URL: https://aclanthology.org/2021.findings-emnlp.159.

[30] Jun Gao et al. *Exploring the Feasibility of ChatGPT for Event Extraction*. 2023. arXiv: 2303.03836 [cs.CL].

[31] Junyang Gao et al. "Efficient Knowledge Graph Accuracy Evaluation". In: *VLDB 2019*. 2019. URL: https://www.amazon.science/publications/efficient-knowledge-graph-accuracy-evaluation.

[32] Tianyu Gao, Xingcheng Yao, and Danqi Chen. "SimCSE: Simple Contrastive Learning of Sentence Embeddings". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Ed. by Marie-Francine Moens et al. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 6894–6910. DOI: 10.18653/v1/2021.emnlp-main.552. URL: https://aclanthology.org/2021.emnlp-main.552.

[33] Yu Gu and Yu Su. "ArcaneQA: Dynamic Program Induction and Contextualized Encoding for Knowledge Base Question Answering". In: *Proceedings of the 29th International Conference on Computational Linguistics*. Ed. by Nicoletta Calzolari et al. Gyeongju, Republic of Korea: International Committee on Computational Linguistics, Oct. 2022, pp. 1718–1731. URL: https://aclanthology.org/2022.coling-1.148.

[34] Yu Gu et al. "Beyond I.I.D.: Three Levels of Generalization for Question Answering on Knowledge Bases". In: *Proceedings of the Web Conference 2021*. WWW '21. ACM, Apr. 2021. DOI: 10.1145/3442381.3449992. URL: http://dx.doi.org/10.1145/3442381.3449992.

[35] Gaole He et al. "Improving Multi-hop Knowledge Base Question Answering by Learning Intermediate Supervision Signals". In: *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. WSDM '21. ACM, Mar. 2021. DOI: 10.1145/3437963.3441753. URL: http://dx.doi.org/10.1145/3437963.3441753.

[36] Junxian He, Graham Neubig, and Taylor Berg-Kirkpatrick. "Efficient Nearest Neighbor Language Models". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Ed. by Marie-Francine Moens et al. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 5703–5714. DOI: 10.18653/v1/2021.emnlp-main.461. URL: https://aclanthology.org/2021.emnlp-main.461.

[37] Aidan Hogan et al. "Knowledge Graphs". In: *ACM Computing Surveys* 54.4 (July 2021), pp. 1–37. ISSN: 1557-7341. DOI: 10.1145/3447772. URL: http://dx.doi.org/10.1145/3447772.

[38] Xixin Hu et al. "Logical Form Generation via Multi-task Learning for Complex Question Answering over Knowledge Bases". In: *Proceedings of the 29th International Conference on Computational Linguistics*. Ed. by Nicoletta Calzolari et al. Gyeongju, Republic of Korea: International Committee on Computational Linguistics, Oct. 2022, pp. 1687–1696. URL: https://aclanthology.org/2022.coling-1.145.

[39] Lei Huang et al. *A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions*. 2023. arXiv: 2311.05232 [cs.CL].

[40] Xin Huang, Jung-Jae Kim, and Bowei Zou. "Unseen Entity Handling in Complex Question Answering over Knowledge Base via Language Generation". In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Ed. by Marie-Francine Moens et al. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 547–557. DOI: 10.18653/v1/2021.findings-emnlp.50. URL: https://aclanthology.org/2021.findings-emnlp.50.

[41] Gautier Izacard and Edouard Grave. "Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Ed. by Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty. Online: Association for Computational Linguistics, Apr. 2021, pp. 874–880. DOI: 10.18653/v1/2021.eacl-main.74. URL: https://aclanthology.org/2021.eacl-main.74.

[42] Gautier Izacard et al. *Atlas: Few-shot Learning with Retrieval Augmented Language Models*. 2022. arXiv: 2208.03299 [cs.CL].

[43] Joel Jang et al. "TemporalWiki: A Lifelong Benchmark for Training and Evaluating Ever-Evolving Language Models". In: *Proceedings of the 2022 Conference*

on Empirical Methods in Natural Language Processing. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 6237–6250. DOI: 10.18653/v1/2022.emnlp-main.418. URL: https://aclanthology.org/2022.emnlp-main.418.

[44] Shaoxiong Ji et al. "A Survey on Knowledge Graphs: Representation, Acquisition, and Applications". In: *IEEE Transactions on Neural Networks and Learning Systems* 33.2 (Feb. 2022), pp. 494–514. ISSN: 2162-2388. DOI: 10.1109/tnnls.2021.3070843. URL: http://dx.doi.org/10.1109/TNNLS.2021.3070843.

[45] Ziwei Ji et al. "Survey of Hallucination in Natural Language Generation". In: *ACM Computing Surveys* 55.12 (Mar. 2023), pp. 1–38. ISSN: 1557-7341. DOI: 10.1145/3571730. URL: http://dx.doi.org/10.1145/3571730.

[46] Jinhao Jiang et al. "StructGPT: A General Framework for Large Language Model to Reason over Structured Data". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 9237–9251. DOI: 10.18653/v1/2023.emnlp-main.574. URL: https://aclanthology.org/2023.emnlp-main.574.

[47] Jinhao Jiang et al. *UniKGQA: Unified Retrieval and Reasoning for Solving Multi-hop Question Answering Over Knowledge Graph*. 2023. arXiv: 2212.00959 [cs.CL].

[48] Saurav Kadavath et al. *Language Models (Mostly) Know What They Know*. 2022. arXiv: 2207.05221 [cs.CL].

[49] Nikhil Kandpal et al. *Large Language Models Struggle to Learn Long-Tail Knowledge*. 2023. arXiv: 2211.08411 [cs.CL].

[50] Minki Kang et al. *Knowledge Graph-Augmented Language Models for Knowledge-Grounded Dialogue Generation*. 2023. arXiv: 2305.18846 [cs.CL].

[51] Jared Kaplan et al. *Scaling Laws for Neural Language Models*. 2020. arXiv: 2001.08361 [cs.LG].

[52] Vladimir Karpukhin et al. "Dense Passage Retrieval for Open-Domain Question Answering". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Bonnie Webber et al. Online: Association for Computational Linguistics, Nov. 2020, pp. 6769–6781. DOI: 10.18653/v1/2020.emnlp-main.550. URL: https://aclanthology.org/2020.emnlp-main.550.

[53] Jungo Kasai et al. *RealTime QA: What's the Answer Right Now?* 2024. arXiv: 2207.13332 [cs.CL].

[54] Wahab Khan et al. "Exploring the frontiers of deep learning and natural language processing: A comprehensive overview of key challenges and emerging trends". In: *Natural Language Processing Journal* 4 (2023), p. 100026. ISSN: 2949-7191. DOI: https://doi.org/10.1016/j.nlp.2023.100026. URL: https://www.sciencedirect.com/science/article/pii/S2949719123000237.

[55] Andrew Lampinen et al. "Can language models learn from explanations in context?" In: *Findings of the Association for Computational Linguistics: EMNLP 2022*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 537–563. DOI: 10.18653/v1/2022.findings-emnlp.38. URL: https://aclanthology.org/2022.findings-emnlp.38.

[56] Yunshi Lan and Jing Jiang. "Query Graph Generation for Answering Multi-hop Complex Questions from Knowledge Bases". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky et al. Online: Association for Computational Linguistics, July 2020, pp. 969–974. DOI: 10.18653/v1/2020.acl-main.91. URL: https://aclanthology.org/2020.acl-main.91.

[57] Yunshi Lan, Shuohang Wang, and Jing Jiang. "Knowledge Base Question Answering with Topic Units". In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, July 2019, pp. 5046–5052. DOI:

10.24963/ijcai.2019/701. URL: https://doi.org/10.24963/ijcai.2019/701.

[58] Yunshi Lan et al. *A Survey on Complex Knowledge Base Question Answering: Methods, Challenges and Solutions*. 2021. arXiv: 2105.11644 [cs.CL].

[59] Yunshi Lan et al. *Complex Knowledge Base Question Answering: A Survey*. 2022. arXiv: 2108.06688 [cs.CL].

[60] Mike Lewis et al. "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky et al. Online: Association for Computational Linguistics, July 2020, pp. 7871–7880. DOI: 10.18653/v1/2020.acl-main.703. URL: https://aclanthology.org/2020.acl-main.703.

[61] Patrick Lewis et al. *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. 2021. arXiv: 2005.11401 [cs.CL].

[62] Aitor Lewkowycz et al. *Solving Quantitative Reasoning Problems with Language Models*. 2022. arXiv: 2206.14858 [cs.CL].

[63] Tianle Li et al. "Few-shot In-context Learning on Knowledge Base Question Answering". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 6966–6980. DOI: 10.18653/v1/2023.acl-long.385. URL: https://aclanthology.org/2023.acl-long.385.

[64] Xingxuan Li et al. *Chain-of-Knowledge: Grounding Large Language Models via Dynamic Knowledge Adapting over Heterogeneous Sources*. 2024. arXiv: 2305.13269 [cs.CL].

[65] Yihao Li et al. *An Enhanced Prompt-Based LLM Reasoning Scheme via Knowledge Graph-Integrated Collaboration*. 2024. arXiv: 2402.04978 [cs.CL].

[66] Jiachang Liu et al. "What Makes Good In-Context Examples for GPT-3?" In: *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*. Ed. by Eneko Agirre, Marianna Apidianaki, and Ivan Vulić. Dublin, Ireland and Online: Association for Computational Linguistics, May 2022, pp. 100–114. DOI: `10.18653/v1/2022.deelio-1.10`. URL: `https://aclanthology.org/2022.deelio-1.10`.

[67] Nelson F. Liu et al. *Lost in the Middle: How Language Models Use Long Contexts*. 2023. arXiv: `2307.03172 [cs.CL]`.

[68] Ye Liu et al. "Uni-Parser: Unified Semantic Parser for Question Answering on Knowledge Base and Database". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 8858–8869. DOI: `10.18653/v1/2022.emnlp-main.605`. URL: `https://aclanthology.org/2022.emnlp-main.605`.

[69] Zemin Liu et al. *GraphPrompt: Unifying Pre-Training and Downstream Tasks for Graph Neural Networks*. 2023. arXiv: `2302.08043 [cs.LG]`.

[70] Haoran Luo et al. *ChatKBQA: A Generate-then-Retrieve Framework for Knowledge Base Question Answering with Fine-tuned Large Language Models*. 2023. arXiv: `2310.08975 [cs.CL]`.

[71] Yubo Ma et al. *Large Language Model Is Not a Good Few-shot Information Extractor, but a Good Reranker for Hard Samples!* 2023. arXiv: `2303.08559 [cs.CL]`.

[72] Alex Mallen et al. "When Not to Trust Language Models: Investigating Effectiveness of Parametric and Non-Parametric Memories". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, July 2023,

pp. 9802–9822. DOI: 10.18653/v1/2023.acl-long.546. URL: https://aclanthology.org/2023.acl-long.546.

[73] Vaibhav Mavi, Anubhav Jangra, and Adam Jatowt. *A Survey on Multi-hop Question Answering and Generation*. 2022. arXiv: 2204.09140 [cs.CL].

[74] James P. McCusker and Deborah L. McGuiness. *What is a Knowledge Graph?* Retrieved 21 March 2017. 2017. URL: https://www.authorea.com.

[75] Tomas Mikolov et al. "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* (2013).

[76] Alexander Miller et al. "Key-Value Memory Networks for Directly Reading Documents". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Ed. by Jian Su, Kevin Duh, and Xavier Carreras. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 1400–1409. DOI: 10.18653/v1/D16-1147. URL: https://aclanthology.org/D16-1147.

[77] Alexander Miller et al. "Key-value memory networks for directly reading documents". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 2016, pp. 1400–1409.

[78] Sewon Min et al. "Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?" In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 11048–11064. DOI: 10.18653/v1/2022.emnlp-main.759. URL: https://aclanthology.org/2022.emnlp-main.759.

[79] Sameh K Mohamed, Aayah Nounu, and Vít Nováček. "Biological applications of knowledge graph embedding models". In: *Briefings in Bioinformatics* 22.2 (Feb. 2020), pp. 1679–1693. ISSN: 1477-4054. DOI: 10.1093/bib/bbaa012. eprint: https://academic.oup.com/bib/article-pdf/22/2/1679/36654277/bbaa012.pdf. URL: https://doi.org/10.1093/bib/bbaa012.

[80] Neo4j-NaLLM. *NaLLM*. GitHub repository. 2023. URL: `https://github.com/neo4j/NaLLM`.

[81] Ontotext. *What is a Knowledge Graph?* Accessed on 1 December 2023. 2018. URL: `https://www.ontotext.com/knowledgehub/fundamentals/what-is-a-knowledge-graph/`.

[82] OpenAI. *(2022)*. Received from. URL: `https://openai.com/research/gpt-3-5-turbo`.

[83] Long Ouyang et al. *Training language models to follow instructions with human feedback*. 2022. arXiv: `2203.02155 [cs.CL]`.

[84] Karl Pearson. "LIII. On lines and planes of closest fit to systems of points in space". In: *The London, Edinburgh, and Dublin philosophical magazine and journal of science* 2.11 (1901), pp. 559–572.

[85] Thomas Pellissier Tanon et al. "From freebase to wikidata: The great migration". In: *Proceedings of the 25th international conference on world wide web*. 2016, pp. 1419–1428.

[86] Shuofei Qiao et al. *Reasoning with Language Model Prompting: A Survey*. 2023. arXiv: `2212.09597 [cs.CL]`.

[87] Stephen Robertson et al. "Okapi at TREC-3." In: Jan. 1994, pp. 0-.

[88] Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. "Improving Multi-hop Question Answering over Knowledge Graphs using Knowledge Base Embeddings". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky et al. Online: Association for Computational Linguistics, July 2020, pp. 4498–4507. DOI: `10.18653/v1/2020.acl-main.412`. URL: `https://aclanthology.org/2020.acl-main.412`.

[89] E. W. Schneider. "Course Modularization Applied: The Interface System and Its Implications For Sequence Control and Data Analysis". In: *Association for the Development of Instructional Systems (ADIS)* (Apr. 1972), pp. 1–21.

[90] Priyanka Sen, Alham Fikri Aji, and Amir Saffari. “Mintaka: A Complex, Natural, and Multilingual Dataset for End-to-End Question Answering”. In: *Proceedings of the 29th International Conference on Computational Linguistics*. Ed. by Nicoletta Calzolari et al. Gyeongju, Republic of Korea: International Committee on Computational Linguistics, Oct. 2022, pp. 1604–1619. URL: https://aclanthology.org/2022.coling-1.138.

[91] Jiaxin Shi et al. “TransferNet: An Effective and Transparent Framework for Multi-hop Question Answering over Relation Graph”. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Ed. by Marie-Francine Moens et al. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 4149–4158. DOI: 10.18653/v1/2021.emnlp-main.341. URL: https://aclanthology.org/2021.emnlp-main.341.

[92] Yiheng Shu et al. “TIARA: Multi-grained Retrieval for Robust Question Answering over Large Knowledge Base”. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 8108–8121. DOI: 10.18653/v1/2022.emnlp-main.555. URL: https://aclanthology.org/2022.emnlp-main.555.

[93] Amit Singhal. *Introducing the Knowledge Graph: things, not strings*. Retrieved 21 March 2017. Official Google Blog. May 2012. URL: https://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html.

[94] Kaitao Song et al. “MPNet: Masked and Permuted Pre-training for Language Understanding”. In: *CoRR* abs/2004.09297 (2020). arXiv: 2004.09297. URL: https://arxiv.org/abs/2004.09297.

[95] Karen Sparck Jones. “A statistical interpretation of term specificity and its application in retrieval”. In: *Document Retrieval Systems*. GBR: Taylor Graham Publishing, 1988, pp. 132–142. ISBN: 0947568212.

[96] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. "Yago: a core of semantic knowledge". In: *Proceedings of the 16th international conference on World Wide Web*. 2007, pp. 697–706.

[97] Haitian Sun, Tania Bedrax-Weiss, and William Cohen. "PullNet: Open Domain Question Answering with Iterative Retrieval on Knowledge Bases and Text". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Ed. by Kentaro Inui et al. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 2380–2390. DOI: 10.18653/v1/D19-1242. URL: https://aclanthology.org/D19-1242.

[98] Haitian Sun et al. "Open Domain Question Answering Using Early Fusion of Knowledge Bases and Text". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Ed. by Ellen Riloff et al. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 4231–4242. DOI: 10.18653/v1/D18-1455. URL: https://aclanthology.org/D18-1455.

[99] Jiashuo Sun et al. *Think-on-Graph: Deep and Responsible Reasoning of Large Language Model on Knowledge Graph*. 2024. arXiv: 2307.07697 [cs.CL].

[100] Yu Sun et al. *ERNIE 3.0: Large-scale Knowledge Enhanced Pre-training for Language Understanding and Generation*. 2021. arXiv: 2107.02137 [cs.CL].

[101] Zhiqing Sun et al. *RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space*. 2019. arXiv: 1902.10197 [cs.LG].

[102] Beth M Sundheim. "The message understanding conferences". In: *TIPSTER TEXT PROGRAM PHASE II: Proceedings of a Workshop held at Vienna, Virginia, May 6-8, 1996*. 1996, pp. 35–37.

[103] Gytė Tamaauskaitė and Paul Groth. "Defining a Knowledge Graph Development Process Through a Systematic Review". In: *ACM Transactions on Software Engineering and Methodology* 32 (2022), pp. 1–40. URL: https://api.semanticscholar.org/CorpusID:248435579.

[104] Gemini Team et al. *Gemini: A Family of Highly Capable Multimodal Models*. 2024. arXiv: 2312.11805 [cs.CL].

[105] David Thulke et al. *Efficient Retrieval Augmented Generation from Unstructured Knowledge for Task-Oriented Dialog*. 2021. arXiv: 2102.04643 [cs.CL].

[106] Hugo Touvron et al. *Llama 2: Open Foundation and Fine-Tuned Chat Models*. 2023. arXiv: 2307.09288 [cs.CL].

[107] Hugo Touvron et al. *LLaMA: Open and Efficient Foundation Language Models*. 2023. arXiv: 2302.13971 [cs.CL].

[108] Sang T. Truong et al. "Crossing Linguistic Horizons: Finetuning and Comprehensive Evaluation of Vietnamese Large Language Models". In: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Seattle, Washington: Association for Computational Linguistics, 2024.

[109] USC Information Sciences Institute. *usc-isi-i2/kgtk: KGTK: Knowledge Graph Toolkit*. 2023. URL: https://github.com/usc-isi-i2/kgtk.

[110] Shikhar Vashishth et al. "InteractE: Improving Convolution-Based Knowledge Graph Embeddings by Increasing Feature Interactions". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.03 (Apr. 2020), pp. 3009–3016. DOI: 10.1609/aaai.v34i03.5694. URL: https://ojs.aaai.org/index.php/AAAI/article/view/5694.

[111] Ashish Vaswani et al. "Attention Is All You Need". In: *CoRR* abs/1706.03762 (2017). arXiv: 1706.03762. URL: http://arxiv.org/abs/1706.03762.

[112] Denny Vrandečić and Markus Krötzsch. "Wikidata: a free collaborative knowledgebase". In: *Commun. ACM* 57.10 (Sept. 2014), pp. 78–85. ISSN: 0001-0782. DOI: 10.1145/2629489. URL: https://doi.org/10.1145/2629489.

[113] Jianing Wang et al. *Boosting Language Models Reasoning with Chain-of-Knowledge Prompting*. 2023. arXiv: 2306.06427 [cs.CL].

[114] Ruize Wang et al. "K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters". In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Ed. by Chengqing Zong et al. Online: Association for Computational Linguistics, Aug. 2021, pp. 1405–1418. DOI: 10.18653/v1/2021.

findings-acl.121. URL: https://aclanthology.org/2021.findings-acl.121.

[115] Xiaozhi Wang et al. "KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation". In: *Transactions of the Association for Computational Linguistics* 9 (2021). Ed. by Brian Roark and Ani Nenkova, pp. 176–194. DOI: 10.1162/tacl_a_00360. URL: https://aclanthology.org/2021.tacl-1.11.

[116] Xuezhi Wang et al. *Self-Consistency Improves Chain of Thought Reasoning in Language Models*. 2023. arXiv: 2203.11171 [cs.CL].

[117] Jason Wei et al. "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models". In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo et al. Vol. 35. Curran Associates, Inc., 2022, pp. 24824–24837. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

[118] Jason Wei et al. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. 2023. arXiv: 2201.11903 [cs.CL].

[119] Jason Wei et al. *Finetuned Language Models Are Zero-Shot Learners*. 2022. arXiv: 2109.01652 [cs.CL].

[120] Xiang Wei et al. *Zero-Shot Information Extraction via Chatting with ChatGPT*. 2023. arXiv: 2302.10205 [cs.CL].

[121] Wikidata. *MediaWiki API*. API Documentation. 2024. URL: https://www.wikidata.org/w/api.php.

[122] Wikipedia contributors. *Knowledge graph — Wikipedia, The Free Encyclopedia*. [Online; accessed 1-December-2023]. 2023. URL: https://en.wikipedia.org/w/index.php?title=Knowledge_graph&oldid=1187111076.

[123] Yike Wu et al. *Retrieve-Rewrite-Answer: A KG-to-Text Enhanced LLMs Framework for Knowledge Graph Question Answering*. 2023. arXiv: 2309.11206 [cs.CL].

[124] Tianbao Xie et al. "UnifiedSKG: Unifying and Multi-Tasking Structured Knowledge Grounding with Text-to-Text Language Models". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 602–631. DOI: 10.18653/v1/2022.emnlp-main.39. URL: https://aclanthology.org/2022.emnlp-main.39.

[125] Yuanmeng Yan et al. "Large-Scale Relation Learning for Question Answering over Knowledge Bases with Pre-trained Language Models". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Ed. by Marie-Francine Moens et al. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 3653–3660. DOI: 10.18653/v1/2021.emnlp-main.296. URL: https://aclanthology.org/2021.emnlp-main.296.

[126] Shunyu Yao et al. *ReAct: Synergizing Reasoning and Acting in Language Models*. 2023. arXiv: 2210.03629 [cs.CL].

[127] Xi Ye et al. "RNG-KBQA: Generation Augmented Iterative Ranking for Knowledge Base Question Answering". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 6032–6043. DOI: 10.18653/v1/2022.acl-long.417. URL: https://aclanthology.org/2022.acl-long.417.

[128] Wen-tau Yih et al. "The Value of Semantic Parse Labeling for Knowledge Base Question Answering". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Ed. by Katrin Erk and Noah A. Smith. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 201–206. DOI: 10.18653/v1/P16-2033. URL: https://aclanthology.org/P16-2033.

[129] Wen-tau Yih et al. "The Value of Semantic Parse Labeling for Knowledge Base Question Answering". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Ed. by Katrin Erk and Noah A. Smith. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 201–206. DOI: 10.18653/v1/P16-2033. URL: https://aclanthology.org/P16-2033.

[130] Diego Lopez Yse. *Knowledge Graphs from scratch with Python*. Accessed on 23 November 2023. 2023. URL: https://lopezyse.medium.com/knowledge-graphs-from-scratch-with-python-f3c2a05914cc.

[131] Donghan Yu et al. *DecAF: Joint Decoding of Answers and Logical Forms for Question Answering over Knowledge Bases*. 2023. arXiv: 2210.00063 [cs.CL].

[132] Wenhao Yu et al. *Generate rather than Retrieve: Large Language Models are Strong Context Generators*. 2023. arXiv: 2209.10063 [cs.CL].

[133] Jing Zhang et al. "Subgraph Retrieval Enhanced Model for Multi-hop Knowledge Base Question Answering". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 5773–5784. DOI: 10.18653/v1/2022.acl-long.396. URL: https://aclanthology.org/2022.acl-long.396.

[134] Lingxi Zhang et al. "FC-KBQA: A Fine-to-Coarse Composition Framework for Knowledge Base Question Answering". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 1002–1017. DOI: 10.18653/v1/2023.acl-long.57. URL: https://aclanthology.org/2023.acl-long.57.

[135] Zhengyan Zhang et al. "ERNIE: Enhanced Language Representation with Informative Entities". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Ed. by Anna Korhonen, David Traum, and Lluís

Màrquez. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 1441–1451. DOI: 10.18653/v1/P19-1139. URL: https://aclanthology.org/P19-1139.

[136] Ruochen Zhao et al. "Verify-and-Edit: A Knowledge-Enhanced Chain-of-Thought Framework". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 5823–5840. DOI: 10.18653/v1/2023.acl-long.320. URL: https://aclanthology.org/2023.acl-long.320.

[137] Lingfeng Zhong et al. *A Comprehensive Survey on Automatic Knowledge Graph Construction*. 2023. arXiv: 2302.05019 [cs.IR].

Appendix

Appendix A: Task assignment

Nguyen Viet Phuong

- Research the tech-stack and popular framework to work with LM prompting i.e. Langchain, HuggingFace.
- Build wrapper for URA-LLaMa API of the URA research team.
- Craft the initial prompt for Entity Extraction from the text. Later adopt the ReFinED model for the Entity Linking process and the overall Question Analysis.
- Research the Related Works.
- Compare the result of work in Related Works with the result in Our Proposed Method i.e. fact-checking.
- Re-run the experiments of the proposed methods for additional attempts.
- Write the report.

Ho Nguyen Ngoc Bao

- Assists in research regarding KG-Augmented LLMs and KGQA
- Research relevant datasets for question-answering benchmark
- Explore the KAPING framework and reimplementing the work in the local environment
- Implement the proposed method
- Run experiments for the proposed methods

- Setting up Github commitments conventions
- Write the report.

Appendix B: Source code

Our source code can be found at this github link:

`https://github.com/DerpLEL/knowledge-base-project`