Doctoral Dissertation

TOWARDS EVIDENCE-BASED REASONING VIA UNSUPERVISED KNOWLEDGE EXTRACTION AND SEMANTIC RETRIEVAL

VO Thien Trung

Supervisor NGUYEN Le Minh

Division of Advanced Science and Technology
Japan Advanced Institute of Science and Technology
Information Science

March 2026

Abstract

Large language models (LLMs) have demonstrated strong capabilities across a wide range of reasoning tasks, especially when supported by prompting or external knowledge. Among the most common ways to incorporate such knowledge is retrieval-augmented generation (RAG), which grounds model outputs in retrieved evidence. In scientific and biomedical settings, however, users need more than a correct answer: they must be able to (i) identify which facts the system relies on, (ii) assess whether the evidence genuinely supports the conclusion, and (iii) verify the reasoning through explicit, fine-grained references to the source material. Accordingly, this thesis argues that evidence-based reasoning with LLMs requires inspectable evidence selection and verifiable reasoning traces, not only fluent answers—especially in biomedical and scientific settings where decisions must be audited. In this thesis, *evidence-based reasoning* means that each prediction is grounded in explicitly selected evidence and accompanied by a reasoning trace that can be checked at the appropriate granularity (sentence-level in text, path-level in graphs, and cell-level in tables).

In practice, many LLMs paired with RAG pipelines still fall short for three structural reasons. First, much of an LLM's knowledge remains implicit in its parameters, making it difficult to inspect, reuse, or maintain as explicit facts. Second, retrieval is frequently driven by lexical matching and vector similarity, which can capture topical relatedness but still miss the precise evidence required for multi-hop inference. Third, reasoning traces are often not audit-friendly—especially for tables, where verification requires localized, cell-level citations rather than generic explanations.

A common solution is to train task-specific retrievers and reasoning models. However, in biomedical and scientific applications, labeled data is often limited and expensive, and the evidence format changes across tasks, including text, graphs, and tables. As a result, supervised systems can be hard to transfer and maintain: they usually need new labels or fine-tuning when the domain or task changes. This motivates an unsupervised and explainable framework that can work with minimal annotation. We therefore organize the thesis around an *unsupervised* pipeline that (i) makes knowledge explicit, (ii) performs *semantic retrieval*, meaning unsupervised *evidence selection* across data formats (sentences in text, columns/cells in tables, and salient paths in graphs), and (iii) produces reasoning traces that can be directly checked. These goals align with three pillars of the dissertation:

unsupervised knowledge extraction makes latent model knowledge explicit, *semantic retrieval* selects the right evidence precisely, and *evidence-based reasoning* exposes verifiable traces for auditing.

This thesis develops an unsupervised and explainable evidence-based reasoning framework that works across tabular and textual settings with minimal annotation. We first study unsupervised semantic retrieval and verifiable reasoning without external knowledge, focusing on tabular evidence. For tabular reasoning and fact verification, UCRET selects claim-relevant columns via spherical k-means and produces label-conditioned counterfactual reasoning with concise cell-level citations. To strengthen evidence selection in tables, UCRET-JS extends this idea with distribution-aware retrieval, using Jensen–Shannon divergence to compare contextual token distributions rather than relying on a single embedding.

While reasoning only from the provided input improves transparency and reduces supervision needs, it also exposes a key limitation: the evidence available in a table or a passage can be implicit, incomplete, or insufficient for multi-hop inference. To address this limitation, the thesis moves to unsupervised semantic retrieval with external knowledge, where knowledge graphs (KGs) provide additional context, enable multi-hop reasoning, and improve answer reliability. K-Bloom converts latent knowledge in pretrained language models into high-precision KG tuples using an Optimal Transport formulation, making model knowledge explicit and reusable. USCRaKe performs unsupervised text retrieval with Optimal Transport and Jensen–Shannon divergence by comparing distributions of contextual tokens rather than single-vector similarity, improving evidence selection for compositional queries. Building on the biomedical KG produced by K-Bloom, UGAT-MedQA applies unsupervised graph attention to identify salient multi-hop paths, which are verbalized into step-by-step evidence-linked reasoning for medical question answering.

Across general, biomedical, and scientific benchmarks, the results show that an unsupervised and explainable pipeline can deliver higher quality evidence selection and more verifiable reasoning without task-specific labels or fine-tuning while remaining adaptable to different evidence formats (tables, text, and graphs). Future work will focus on reducing hallucinations, making retrieval more intent- and verification-oriented, maintaining domain knowledge graphs over time, and extending unsupervised table reasoning to noisier real-world and clinical settings.

Keywords: Large Language Models, Knowledge Graphs, Optimal Transport, Unsupervised Learning, Question Answering, Table-based Fact Verification

Acknowledgment

First of all, I would like to express my best sincerest gratitude to my principal advisor, Professor Nguyen Le Minh of Japan Advanced Institute of Science and Technology (JAIST), for his exceptional guidance, intellectual insight, and unwavering support throughout the course of my doctoral research. His profound expertise and scholarly rigor have greatly shaped my academic development and been indispensable to the completion of this dissertation. I am particularly grateful for his constructive feedback, inspiring discussions, and constant encouragement, which have enriched both my research and personal growth. Without his consistent support, I could not finish the work in this dissertation.

I wish to acknowledge the Japan Advanced Institute of Science and Technology for providing an excellent academic environment and the necessary infrastructure that enabled me to pursue my research objectives effectively. I am thankful to the faculty members, administrative staff, and technical personnel whose professionalism and support have greatly facilitated my academic journey. I also extend my sincere appreciation to all JAIST staff for creating a wonderful environment for both research and life. I would love to devote my sincere thanks and appreciation to all members of Nguyen's laboratory. Being a member of Nguyen's lab and JAIST is a wonderful time of my research life.

I am deeply thankful to my family, friends, and colleagues for their encouragement, patience, and understanding, which have sustained me through the challenges of this academic endeavor. Without their support, I would never have completed this work.

Finally, I would like to express my sincere appreciation to all those who have contributed, directly or indirectly, to the completion of this dissertation. Their intellectual, moral, and practical support has been indispensable, and I remain profoundly grateful for their presence throughout this long and rewarding journey.

List of Abbreviations

Abbreviation	**Meaning**
LLM	Large Language Model
PLM	Pre-trained Language Model
RAG	Retrieval-Augmented Generation
NLP	Natural Language Processing
KG	Knowledge Graph
KB	Knowledge Base
DPR	Dense Passage Retrieval
OT	Optimal Transport
EMD	Earth Mover's Distance
WMD	Word Mover's Distance
KL	Kullback-Leibler divergence
JSD	Jensen-Shannon Divergence
GNN	Graph Neural Network
GCN	Graph Convolutional Network
GAT	Graph Attention Network
QA	Question Answering
BQA	Biomedical Question Answering
MCQA	Multiple-Choice Question Answering
FV	Fact Verification
TFV	Table-based Fact Verification
TQA	Table-based Question Answering
CCoE	Counterfactual Chain of Explanation
CCoV	Counterfactual Chain of Verification

Table 1: List of abbreviations used in this thesis.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 Research Overview and Motivation

Recent years have seen rapid progress in pre-trained foundation models, especially large language models (LLMs), which achieve strong performance across a broad range of tasks [3,4], including recommender systems [5], molecular discovery [6], and report generation [7]. These models are trained on massive corpora and often exhibit emergent capabilities [5, 8], such as robust language understanding, instruction following, and in-context learning.

To improve the faithfulness and usefulness of generated outputs, Retrieval-Augmented Generation (RAG) [9] has become a representative technique in generative AI. RAG conditions the model on retrieved evidence from external sources [6, 9, 10], and is often effective with minimal or no additional training [11]. This design has shown promise in knowledge-intensive tasks such as open-domain question answering [12–14], as well as in broader language tasks and downstream applications [15–19].

In scientific and biomedical settings, however, users need more than a correct answer: they must be able to (i) identify which facts the system relies on, (ii) assess whether the evidence genuinely supports the conclusion, and (iii) verify the reasoning through explicit, fine-grained references to the source material. Accordingly, this dissertation argues that evidence-based reasoning with LLMs requires *inspectable evidence selection* and *verifiable reasoning traces*, not only fluent answers—especially in biomedical and scientific settings where decisions must be audited.

Despite these successes, both LLMs and RAG still face challenges that affect their reliability and interpretability in complex domains such as biomedicine and science. In practice, many LLMs paired with RAG pipelines still fall short for three structural reasons.

Limitation 1: Knowledge is implicit and hard to reuse. Much of an LLM's knowledge remains implicit in its parameters, making it difficult to inspect, reuse, or maintain as explicit facts. Language models such as

BERT [20], GPT-3 [21], and T5 [22] are trained on large, diverse datasets and store this knowledge implicitly. This raises a fundamental question: what specific facts reside inside these models, and how can we extract and represent them systematically? A promising approach involves knowledge graphs (KGs), which represent information as triples (h, r, t) and make relationships within a domain accessible to both humans and machines [23]. Integrating pretrained language models (PLMs) with KGs has improved applications such as search engines [24], recommendation [25], and question answering [26, 27]. However, existing extraction methods can be noisy. For example, BertNet [28] extracts relational triples from masked sentences, but the resulting graphs may contain inconsistent entity pairs, which reduces reliability and precision. This motivates an **unsupervised knowledge extraction** component, which turns PLM knowledge into precise, reusable triples without labels.

Limitation 2: Retrieval is often pointwise and misses multi-hop evidence. Retrieval is frequently driven by lexical matching and vector similarity (e.g., BM25, cosine, Euclidean). These methods capture topical relatedness, but they can still miss the precise evidence required for multi-hop inference. While RAG helps by retrieving external context, its reliance on Euclidean or cosine similarity [29–33] is often insufficient for capturing fine-grained semantic relationships in contextualized embeddings, especially in complex multi-hop reasoning scenarios. Moreover, although BM25 [34] and Dense Passage Retrieval (DPR) [35] enable efficient retrieval from large corpora, they frequently lack semantic depth [36]. This motivates our **unsupervised semantic retrieval** component, using Optimal Transport with Jensen–Shannon divergence to capture distribution-level semantics beyond surface overlap and single-vector matching.

Limitation 3: Reasoning is hard to verify, especially for graphs and tables. Reasoning traces are often difficult to verify, particularly when the evidence is structured (e.g., graphs and tables). For tables, verification is inherently *localized*: users need precise, cell-level support rather than generic natural-language rationales. However, most LLMs are primarily trained on unstructured text, which limits robust generalization to semi-structured inputs and multi-step operations over tabular data [37–41]. While recent progress in table-based fact verification and table reasoning (e.g., TART [42], Table-LLaVA [43], and others [44–46]) has shown promise, many approaches rely on fine-tuning, domain-specific training, or closed models, which reduces transferability and complicates auditability in practice. This

motivates our **unsupervised column selection** mechanism and citation-centric reasoning, designed to expose exactly *which* table cells support each decision.

For knowledge graphs, the challenge shifts from citing cells to isolating a *salient subgraph* for multi-hop inference. Large KGs contain millions of facts, and naive retrieval can introduce substantial noise, which confuses downstream LLM reasoning. Existing approaches [47–50] either rely on generic retrievers, heuristic traversal, or costly LLM-driven graph exploration; supervised node selection further requires labeled data or hand-crafted rules, limiting scalability across biomedical domains. These issues underscore the need for an unsupervised yet structured mechanism to estimate relevance and surface *evidence-linked* multi-hop paths that users can inspect.

Finally, LLMs still face inherent limitations such as hallucinations and outdated internal knowledge [5, 8]. This further strengthens the case for evidence-based pipelines where intermediate evidence and reasoning steps are explicitly surfaced for inspection and post-hoc verification [51].

Why unsupervised. A common solution is to train task-specific retrievers and reasoning models. In biomedical and scientific applications, however, this is often impractical: labeled data is often limited and expensive, and the evidence format changes across tasks (tables, text, and graphs). As a result, supervised systems are harder to transfer and maintain: they usually need new labels or fine-tuning when the domain or task changes. This motivates an unsupervised and explainable framework that can work with minimal annotation. We therefore organize the thesis around an *unsupervised* pipeline that (i) makes knowledge explicit, (ii) performs *semantic retrieval*, meaning unsupervised *evidence selection* across data formats (sentences in text, columns/cells in tables, and salient paths in graphs), and (iii) produces reasoning traces that can be directly checked. These objectives align with the three core pillars of our dissertation: unsupervised knowledge extraction, which transforms latent model knowledge into explicit facts; semantic retrieval, which ensures the most relevant evidence is accurately selected; and evidence-based reasoning, which generates verifiable traces for rigorous auditing.

The research is organized into two complementary scenarios:

- **Unsupervised semantic retrieval without external knowledge:** The system first performs evidence selection from the given input alone (e.g., identifying claim-relevant table columns) and supports the output with *checkable citations*.
- **Unsupervised semantic retrieval with external knowledge:**

When the input evidence is implicit or incomplete, the system first **retrieves additional evidence** from external resources (e.g., text/KGs), then **performs multi-step reasoning** using evidence-linked traces that remain inspectable.

Overview of our unified pipeline. Across both settings, we develop a unified pipeline that works with different evidence formats (tables, text, and graphs) under unsupervised manner. The core principle is to use a specific selector for each evidence type: identifying relevant columns and cells in tables, selecting sentences in text, and finding salient paths in knowledge graphs. Regardless of the input format, the system ensures that all reasoning remains checkable through explicit citations or evidence-linked traces.

Part I: Unsupervised semantic retrieval and verifiable reasoning without external knowledge. We first study verifiable reasoning when the system must rely only on the given input, with a focus on tables. For table-based fact verification, **UCRET** selects claim-relevant columns using spherical k-means, forms a compact evidence table, and generates a predicted answer and label-conditioned explanations with concise, cell-level citations. To further strengthen table evidence selection, **UCRET-JS** extends this idea with distribution-aware scoring: it uses Jensen–Shannon divergence over contextual token distributions rather than relying on a single embedding.

Reasoning only from the provided table improves transparency because the supporting evidence is directly observable and can be cited at the cell level. However, it also exposes a key limitation: for multi-hop questions, the necessary facts are not explicitly stated in a single input. Crucial background knowledge may be unstated, absent, or fragmented across multiple locations, making input-only reasoning frequently insufficient. This motivates our second setting, where external knowledge is introduced to support multi-hop inference while still keeping evidence selection and reasoning traces explicit and inspectable.

Part II: Unsupervised semantic retrieval and verifiable reasoning with external knowledge. To address this limitation, we move to unsupervised semantic retrieval with external knowledge. **K-Bloom** converts latent knowledge in pretrained language models into high-precision KG tuples using an Optimal Transport formulation, making model knowledge explicit and reusable. Building on the biomedical KG produced by K-Bloom, **UGAT-MedQA** applies unsupervised graph attention to identify salient multi-hop paths, which are verbalized into step-by-step, evidence-linked reasoning for medical question answering. In addition, **USCRaKe** performs unsupervised text retrieval with Optimal Transport and Jensen–

Shannon divergence by comparing distributions of contextual tokens rather than single-vector similarity, improving evidence selection for multi-hop and compositional questions

Key novelty. This dissertation proposes a unified, unsupervised pipeline that **extracts knowledge**, **selects evidence**, and **produces verifiable reasoning** across tables, knowledge graphs, and text. Its novelty lies in a *family* of unsupervised evidence selectors tailored to each format: Optimal Transport-based scoring for knowledge extraction, distribution-aware Optimal Transport retrieval for text, unsupervised graph attention for identifying salient multi-hop paths, and clustering-based selection for claim-relevant table columns. Because these selectors do not rely on labeled training data, the pipeline can be reused across tasks while still producing transparent, verifiable reasoning traces.

1.2 Research Objective and Contributions

Research objective. The objective of this research is to develop an effective, **unsupervised**, and **explainable** pipeline that **extracts knowledge**, **retrieves evidence**, and **reasons transparently** across text, knowledge graphs, and tables, while achieving **competitive accuracy** with **minimal annotation**. The central research question is: *How can we develop fully unsupervised and explainable methods that extract knowledge from pretrained language models, identify and retrieve relevant evidence across text, knowledge graphs, and tables, and provide transparent reasoning for question answering and fact verification, all without supervised training?*

Main contributions. To answer this, the dissertation proposes and evaluates the following contributions:

- **UCRET (Unsupervised Column Extraction for Table-based Fact Verification without Knowledge Graphs):** An interpretable framework for table-based fact verification that uses spherical k-means as unsupervised column selection to form a concise evidence table and apply label-conditioned counterfactual reasoning (SUPPORTS, REFUTES, NOT ENOUGH INFO) to deliver precise decisions with clear explanations, achieving strong performance without using a knowledge graph.
- **UCRET-JS (Distribution-Aware Table Evidence Selection):** An extension of UCRET that strengthens table evidence selection

using Jensen–Shannon divergence over contextual token distributions, improving robustness for table QA tasks without requiring labeled training data.

- **K-Bloom (Unsupervised Knowledge Extraction):** A method for extracting high-quality knowledge graphs from pretrained language models using OT-based distance scoring, expanding from minimal seeds to produce a reusable symbolic resource for downstream reasoning.
- **UGAT-MedQA (Unsupervised Graph Attention Network for Biomedical QA):** We construct a biomedical knowledge graph using K-Bloom, then apply an unsupervised Graph Attention Network to select salient multi-hop paths, verbalize those paths, and supply them to the LLM as evidence, yielding step-by-step and readable explanations for medical QA without task-specific labels.
- **USCRaKE (Unsupervised semantic chunk retrieval):** Formulate sentence-level retrieval as Optimal Transport with Jensen–Shannon divergence as the ground cost, enriched with lightweight one-hop ConceptNet verbalization, yielding robust evidence selection without labels.
- **Generality and Impact:** Extensive experiments across general, biomedical, and scientific datasets show competitive accuracy and clear explanations, narrowing the gap with supervised approaches while improving transparency and flexibility.

1.3 Overview of the Dissertation

Figure 1.1 illustrates the overall structure of this dissertation. We propose a unified, unsupervised framework that integrates knowledge extraction, semantic retrieval, and structured reasoning with large language models to address diverse natural language processing tasks. The primary goal is to develop modular and interpretable systems that function effectively without labeled data or task-specific fine-tuning. To achieve this, the thesis is organized into two parts: (i) unsupervised semantic retrieval and verifiable reasoning without external knowledge, and (ii) unsupervised semantic retrieval with external knowledge.

In summary, this thesis delivers a practical, transparent, and flexible framework. It shows how to discover the knowledge hidden inside the language models, retrieve semantically precise evidence from text and tables, and reason step by step—without labeled data. Ultimately, this work establishes a modular, interpretable, and scalable paradigm for building trustworthy AI systems capable of accurate and explainable inference across

diverse real-world language understanding tasks.

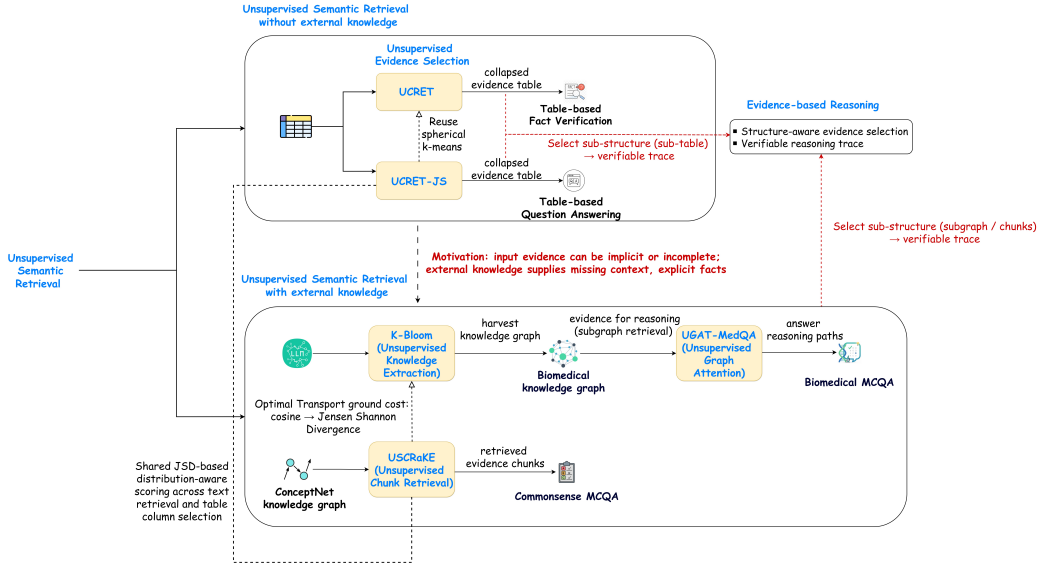


Figure 1.1: Dissertation Outline.

The remainder of the thesis is organized as follows.

- **Chapter 2: Background** reviews foundations in knowledge extraction, semantic retrieval, explainable reasoning, KG-augmented QA, and table-based verification, and summarizes key limitations that motivate the proposed methods.
- **Chapter 3: UCRET** presents an unsupervised approach for table-based fact verification via spherical k-means column selection and checkable, label-conditioned explanations with cell-level citations.
- **Chapter 4: UCRET-JS - Unsupervised Column Relevance Extraction for Table-Based Question Answering via Structured and Explainable Large Language Model Reasoning** presents a KG-free method for table QA. It selects relevant columns using spherical k-means and explains decisions with label-conditioned counterfactual reasoning.
- **Chapter 4: UCRET-JS** focuses on **table question answering** and strengthens column and evidence selection with Jensen–Shannon divergence over contextual token distributions.
- **Chapter 5: K-Bloom** introduces an unsupervised method to extract high-precision KG triples from pretrained language models using Optimal Transport scoring.

- **Chapter 6: UGAT-MedQA** uses the extracted biomedical KG from K-Bloom and unsupervised graph attention to select salient multi-hop paths and verbalize them into evidence-linked medical QA reasoning.
- **Chapter 7: USCRaKe** proposes unsupervised semantic chunk retrieval for text using Optimal Transport with Jensen–Shannon divergence, and shows how retrieved evidence supports interpretable reasoning for multiple-choice QA.
- **Chapter 8: Conclusion and Future Work** summarizes the thesis and outlines future directions for unsupervised evidence selection and verifiable reasoning across formats.

Note on chapter order. Although **UCRET-JS** appears before **USCRaKE** in this dissertation, the reuse of Jensen–Shannon divergence should be understood as a shared design principle rather than a dependency on later chapters. UCRET-JS adopts the same *distribution-aware* motivation in the table setting to improve robustness in column selection for option-driven table QA. The general formulation and analysis of Jensen–Shannon divergence within an Optimal Transport retrieval framework is then presented in detail in the USCRaKE chapter.

Chapter 2

Background

2.1 Unsupervised Knowledge Extraction from PLMs to Knowledge Graphs

2.1.1 Knowledge Graph Construction and Extraction

2.1.1.1 From Text to Graphs

Knowledge Graph Construction (KGC) traditionally follows pipelines for entity, relation, and attribute extraction from text, producing triples that can be integrated into large KGs [52–54]. Human-curated resources such as Freebase, WordNet, and Wikidata [55–57] offer broad coverage but are costly to maintain. Automated approaches (OpenIE, YAGO) [58–61] reduce annotation cost but may introduce noise or require significant post-processing. This motivates methods that can extract higher-precision triples with minimal supervision, which is the focus of this dissertation.

2.1.1.2 From PLMs to Graphs

A complementary line of work aims to *extract* graph-structured knowledge *from* PLMs themselves, using prompts to elicit relations and entities (e.g., LAMA; prompt-engineering and automated prompt discovery). Compared to text-only extraction, PLM-based extraction can quickly surface factual associations learned during pre-training. However, prior approaches often predict missing *objects* for pre-specified $(\text{subject}, \text{relation}, \_)$ templates, offering limited support for discovering *new* entity pairs or expanding around seed knowledge. This gap motivates seed-guided, unsupervised strategies that (i) search beyond single triples, (ii) score candidate pairs robustly, and (iii) assemble outputs into a coherent KG suitable for downstream reasoning.

2.1.2 Probing PLMs for Factual Knowledge

PLMs such as BERT [20] encode broad linguistic and factual regularities learned from large corpora. A central question is how much of this knowledge is retrievable in a controlled way. Early factual probing frameworks (e.g., LAMA) treat PLMs as implicit knowledge bases by filling cloze prompts to recover $(\text{subject}, \text{relation}, \text{object})$ triples [51]. Subsequent work explores prompt design, paraphrasing, and prompt automation to improve recall and stability of retrieved facts (e.g., conditional masked language modeling [62], prompt tuning and AutoPrompt).

Despite progress, factual probing exposes several limitations relevant to this dissertation: (i) outputs may be *prompt-sensitive* and inconsistent across paraphrases, (ii) coverage remains biased toward frequent entities while long-tail facts are underrepresented, and (iii) retrieved snippets are not immediately *structured* for downstream reasoning. These issues motivate methods that transform latent parametric knowledge into explicit, reusable symbolic structures and are less sensitive to prompt wording.

2.2 Semantic Retrieval for Question Answering

Retrieval-augmented generation (RAG) couples a parametric generator with a non-parametric memory of external documents to improve factual grounding [9]. Typical systems retrieve passages (e.g., via FAISS [63]) and condition a generator such as BART [64] to produce answers, yielding gains on open-domain QA benchmarks like Natural Questions [65] and reducing hallucinations in dialogue [66].

However, standard dense retrieval pipelines often depend on basic similarity (cosine or Euclidean) over chunked text and may return long, loosely related passages. This is problematic for multi-step reasoning or fine-grained alignment at the sentence level. Moreover, many RAG variants benefit from supervised retriever–generator tuning, which can diminish cross-domain portability. These observations motivate *unsupervised* retrieval mechanisms that (i) operate at finer granularity, (ii) capture subtle semantic relations beyond cosine distance, and (iii) remain effective without task-specific labels. Therefore, in this dissertation, we propose unsupervised text retrieval using Optimal Transport with Jensen–Shannon divergence to score sentence-level alignments while preserving an unsupervised pipeline robustly.

2.3 Explainable Reasoning with Graph-Augmented LLMs

2.3.1 Retrieval-Augmented Question Answering Systems

Retrieval-Augmented Generation (RAG) [9] has emerged as a prominent architecture for open-domain question answering and knowledge-intensive tasks due to its explainability, scalability, and reduced hallucination compared to purely parametric models. RAG seamlessly integrates parametric memory, represented by a pre-trained seq2seq generator BART [64], with non-parametric memory consisting of dense vector representations of external documents indexed by FAISS [63]. Given a query, RAG encodes it into dense embeddings, retrieves relevant passages from an external knowledge base, and conditions the generator on these retrieved documents to produce the final output. This retrieval-augmented design not only grounds the generation process in external evidence but also enables end-to-end training to jointly optimize both the retriever and generator components. Empirical results demonstrate RAG's effectiveness on benchmarks such as Natural Questions [65], outperforming parametric-only models by leveraging external knowledge to improve factual accuracy. Furthermore, RAG has been shown to reduce hallucination in knowledge-grounded tasks such as dialogue generation [66], where grounding responses in retrieved evidence enhances informativeness and faithfulness to source content. Despite their success, RAG pipelines [67, 68] depend on retrievers that use traditional metrics such as cosine or Euclidean distance to compute similarity over dense embeddings. In addition, though traditional techniques such as BM25 [34] and Dense Passage Retrieval [35] are highly effective at efficiently locating relevant documents across massive datasets, they frequently fall short when it comes to capturing the deeper, underlying semantic meaning of the content. This dependency reduces their effectiveness on complex reasoning tasks that require subtle semantic alignment, as it often results in the retrieval of long passages with loosely related content. Specifically, these methods remain limited by the chunking style of knowledge organization and struggle to effectively capture information in complex queries. In addition, many RAG variants [69–72] jointly fine-tune the retriever and generator in a supervised manner, which improves in-domain accuracy but reduces portability to unseen domains. These issues highlight the need for a retrieval mechanism that can capture fine-grained semantic alignment while remaining unsupervised. Unlike prior RAG pipelines that depend on cosine or Euclidean similarity and often require supervised fine-tuning of retriever–generator pairs, our framework

introduces a theoretically grounded OT–JSD retrieval mechanism. Optimal Transport is a paradigm for transforming one probability distribution into another while minimizing the cost of the transformation [73, 74]. Its strength lies in optimizing a transportation cost matrix, which effectively captures various modalities and constraints, such as text-text alignment and image-image alignment in a fine-grained manner. This unsupervised design captures fine-grained semantic alignment without task-specific supervision, thereby improving cross-domain generalization while reducing reliance on large annotated datasets.

2.3.2 Knowledge Graph-Augmented Question Answering

In our framework, structured knowledge from ConceptNet is leveraged as an auxiliary evidence source to support reasoning in MCQA. This approach aligns with the broader trend of incorporating structured knowledge into QA systems to enhance factual consistency and reasoning transparency. Existing approaches in incorporating knowledge graphs into question answering can be broadly grouped into three categories: embedding-based methods, semantic parsing methods, and retrieval-augmented methods. Embedding-based approaches embed entities and relations from KGs into continuous vector spaces and employ various architectures—such as key-value memory networks [75], sequence modeling [76], or graph neural networks [2]—to learn reasoning paths that connect the question with relevant entities and relations. On the other hand, semantic parsing-based methods convert natural language questions into structured query languages (e.g., SPARQL) using semantic parsers. These queries are then executed over the KG to retrieve answers [77–80]. However, such methods often treat reasoning as an external process executed by the KG engine, thereby underutilizing the reasoning capabilities of the model itself. Retrieval-augmented methods aim to bridge the gap between the structured nature of knowledge graphs and the reasoning capabilities of neural models. These approaches retrieve relevant knowledge triples or subgraphs from the KG and integrate them into the model's inference process to improve factual grounding. For instance, GraftNet [81] performs entity linking to extract local subgraphs, while PullNet [82], SR [83], DiFar [84], and UniKGQA [85] leverage dense retrievers to access more semantically relevant KG content. Building on this foundation, the emergence of large language models has given rise to RAG [9,86–88], a paradigm that integrates retrieved external knowledge—structured or unstructured—into generative pipelines. In this setting, structured KG triples are often verbalized into natural

language to guide generation or validate reasoning chains. For example, [89] uses KG-derived facts to correct hallucinated reasoning steps, while DECAF [90] jointly retrieves knowledge and generates both structured queries and natural language answers, unifying the strengths of semantic parsing and generation. Although graphs serve as effective knowledge carriers, their utility is constrained by several limitations. First, most publicly available KGs remain incomplete and inconsistent, as constructing high-quality graphs is resource-intensive and hampered by the lack of a unified ontology. Second, corpus-based KG construction faces a trade-off between efficiency and effectiveness: fine-grained graphs preserve detail but are computationally costly, while compact graphs risk losing critical information. These limitations motivate the need for a framework that can exploit KGs in a lightweight yet context-sensitive way, while complementing them with more robust semantic retrieval.

2.3.3 Question Answering over Knowledge Graphs

Question answering over a knowledge graph aims to retrieve and apply relevant facts from the knowledge graph to answer natural language questions. Given a natural language question q, a list of options $\mathcal{A}_q$, and a KG $\mathcal{G}$, the task aims to design a function f to reason answers $a \in \mathcal{A}_q$ based on knowledge from $\mathcal{G}$, i.e., $a = f(q, \mathcal{G})$. Recent advancements in question answering over a knowledge graph can be broadly categorized into two paradigms: semantic parsing-based and retrieval-based approaches. Semantic parsing (SP)-based methods utilize LLMs to convert natural language queries into structured logical forms such as S-expressions or SPARQL queries, which can be directly executed on a knowledge graph to retrieve precise answers [91], [92], [93]. These methods exploit the structured nature of KGs to enable interpretable and deterministic reasoning. In contrast, retrieval-based approaches aim to enhance response generation by extracting relevant entities, relations, or relational paths from the knowledge graph and conditioning large language models on this contextual information [94, 95]. Recently, approaches based on LLMs have utilized the reasoning abilities of these models to generate answers in a step-by-step process without requiring additional training [50], [96], [97]. The study by He et al. [48] focuses on retrieving additional contextual information relevant to a given question and using it as extra input to enhance the performance of LLMs, aiming to improve answer accuracy. This highlights the benefit of augmenting language models with external knowledge to improve performance in specialized domains.

2.3.4 Graph-Augmented Language Models for Question Answering

Integrating language models (LMs) with graphs containing natural language information has emerged as a significant research area [98]. Existing methods generally fall into two main categories: (i) leveraging latent graph-based features—often extracted with graph neural networks—to enhance language models [99, 100], and (ii) explicitly feeding verbalized graph information as part of the language model's input [96, 101]. The first category struggles with the inherent differences between graph structures and natural language, which can lead to limited effectiveness on knowledge-intensive tasks [102]. The second category, meanwhile, often suffers from the inclusion of noisy or irrelevant information retrieved from large graphs, which can hinder the reasoning performance of language models [48, 103]. To overcome these challenges, our approach integrates GAT-based retrieval with RAG for multiple-choice question answering, yielding superior results compared to previous methods.

2.4 Table-Based Fact Verification and Question Answering with Reasoning

Table-based reasoning tasks, including question answering, fact verification, and summary generation, require models to interpret structured tabular data and perform precise inference. Early work addressed these challenges through symbolic approaches, such as translating natural language into executable queries (e.g., SQL, SPARQL) [104, 105], or using graph neural networks to model table structure [106, 107]. However, these approaches often struggled to generalize due to their dependence on schema-specific formats and handcrafted operations. Pretrained neural methods such as TAPAS [108] extend BERT with table-aware embeddings and achieve strong results on cell selection and aggregation, typically under weak supervision, and TaBERT [109] jointly pretrains over text–table pairs to improve semantic parsing and QA via task-specific fine-tuning. However, both lines generally rely on large-scale pretraining and labeled adaptation, limiting plug-and-play transfer to new domains. These limitations highlight the motivation for a new approach that enhances flexibility and transparency across diverse domains.

With the emergence of large language models, recent methods have explored language-driven table reasoning. Several studies introduced table pretraining strategies that align textual and tabular representations to improve LLMs' ability to reason over semi-structured inputs [39, 110, 111].

Building on this, Ye et al. [111] proposed a decomposition-based strategy that simplifies reasoning by breaking down complex tables and questions into subcomponents. Chain-of-Table [112] incorporates table structures into chain-of-thought prompting, while ReAcTable [113] integrates reactive and proactive planning to handle complex multi-step reasoning.

Despite these advances, many existing methods rely primarily on textual reasoning, which can be insufficient for tasks that require accurate numerical comparison or structural table understanding. To improve flexibility and performance, hybrid systems have emerged that dynamically assign subtasks to symbolic (e.g., SQL) or textual reasoning modules depending on the query [113, 114]. However, these systems still face challenges in identifying the most relevant table columns and providing interpretable justifications for their decisions. On the other hand, Liu et al. [115] proposed Multi-aspect Adversarial Contrastive Learning (Macol), which leverages adversarial examples generated from multi-aspect reasoning to train a tabular-textual encoder via contrastive learning. Although Macol achieved promising results on three Wikipedia-based table fact-checking datasets, it requires substantial computational resources and struggles to generalize to diverse datasets due to its reliance on multi-aspect reasoning rules.

To address these limitations, we propose a zero-shot, unsupervised pipeline that uses spherical k-means for claim-driven column filtering and open-source LLMs for interpretable reasoning, providing flexibility and transparency without dependence on task-specific fine-tuning. For fact verification, we apply label-conditioned counterfactual reasoning to generate structured explanations for SUPPORTS, REFUTES, and NOT ENOUGH INFO in fact verification, or to derive precise answers in table-based question answering. The result is a compact, auditable pipeline for table reasoning—covering both fact verification and question answering without relying on KGs. Our method explicitly targets the dual goals of precision and interpretability, aiming to bridge the gap between symbolic and textual reasoning in table-based fact verification.

2.5 Positioning and Identified Gaps

The surveyed literature points to four gaps this dissertation targets:

1. **From probing to usable structure:** Factual probing retrieves token completions, but downstream reasoning benefits from explicit, reusable KGs. We bridge this with seed-guided, unsupervised extraction that scores and assembles entity–relation knowledge from PLMs.

2. **Fine-grained, unsupervised retrieval:** Dense retrieval with cosine similarity over long passages is coarse and often supervised. We propose an Optimal Transport + Jensen–Shannon Divergence mechanism for *sentence-level*, unsupervised alignment.
3. **Selective graph grounding:** KG integration boosts transparency but can inject noise. We introduce learned graph attention for *targeted* node/path selection before verbalization into LLM inputs.
4. **Interpretable table verification:** Many table methods lack explicit column relevance and contrastive rationales. We combine unsupervised column filtering with counterfactual, label-specific explanations to improve auditability.

Chapter Summary

This chapter surveyed core areas relevant to the dissertation: probing PLMs for factual knowledge, constructing and exploiting KGs, semantic retrieval and RAG, graph-augmented QA, and table-centric reasoning. Common limitations—prompt sensitivity, incomplete KGs, coarse similarity measures, and unclear explanations—shape the design choices in subsequent chapters. The dissertation responds with unsupervised, explainable methods for (i) extracting KGs from PLMs, (ii) fine-grained semantic retrieval, (iii) selective graph grounding for medical QA, and (iv) interpretable fact verification over tables without relying on KGs.

Chapter 3

UCRET: Unsupervised Column Relevance Extraction for Table-Based Fact Verification via Structured and Explainable Large Language Model Reasoning

3.1 Introduction

Tables are a fundamental data format widely used in diverse domains such as healthcare, finance, and scientific research due to their ability to organize complex information systematically [116]. Unlike unstructured text, tables encode rich information through the interaction of rows and columns, enhancing their data capacity and introducing significant challenges for automated reasoning [108]. The tasks involving tabular data, such as table-based fact verification (TFV) and table question answering, require models to not only comprehend the structural relationships within tables but also perform precise numerical reasoning and multi-step inference [39, 117].

However, most large language models (LLMs) are primarily trained on unstructured text, limiting their ability to generalize to semi-structured inputs like tables or hybrid formats combining tables and text, as seen in [37–40]. These models often struggle with multi-step reasoning tasks, especially when dealing with complex mathematical operations involving tables, as presented in [41]. Although recent work [118] has demonstrated the capacity of LLMs to process and reason with semi-structured tables, this study remains limited and does not directly assess the mathematical reasoning abilities of the models.

One prominent direction in recent work focuses on planning-based approaches that generate step-wise plans to guide the reasoning process [112, 113, 119, 120]. Recent works in this direction either fine-tune open

LLMs [42, 44] or rely on prompting closed-source models [112, 113, 119]. However, fine-tuning relies on high-quality data, which is often costly to acquire [42, 121]. While prompt-based methods using commercial LLMs can be costly and raise concerns about reproducibility and accessibility. In addition, these approaches [112, 113] primarily utilize textual reasoning techniques, such as chain-of-thought, which often lack the precision needed for accurate table interpretation and numerical computation, both essential to effective table-based reasoning.

Recent advancements in table-based fact verification have been driven by state-of-the-art (SOTA) methods, including TART [42], Table-LLaVA [43], PROTRIX [44], MACT [45], and SynTab-LLaVA [46]. Despite their progress, these approaches face significant challenges. TART relies on multiple LLMs to parse and reason over tables, which raises computational complexity. Similarly, MACT leverages multiple LLMs as multi-agent collaboration framework with tool use, resulting in increased computational complexity. To improve multi-step reasoning over tables, PROTRIX generates instruction-tuning data using a Plan-then-Reason pipeline to fine-tune LLMs, further increasing model complexity and training cost.

To address these challenges, we propose **UCRET** (**U**nsupervised **C**olumn **R**elevance **E**xtraction for **T**able-Based Fact Verification via Structured and Explainable Large Language Model Reasoning), a fully interpretable, zero-shot framework for table-based fact verification, requiring neither fine-tuning nor pretraining, and enabling robust, explainable reasoning with open-source LLMs. UCRET consists of three primary stages: (1) **Column Filtering**, which constructs a condensed evidence table by selecting semantically relevant columns using unsupervised clustering; (2) **Counterfactual Chain-of-Explanation**, which prompts an LLM to generate structured explanations for each veracity label assumption; and (3) **Answer Generation**, which compares the generated explanations to determine the most plausible veracity label through the verification process. UCRET is easy to deploy, works with any open-source LLM, and is readily adaptable to new domains or downstream tasks. The three components of UCRET form a coherent, modular pipeline, where each stage explicitly improves the reasoning quality of the next. Column Filtering selects only relevant evidence, Counterfactual Explanation considers all possible outcomes, and Answer Generation reflects on the alternatives to ensure interpretability and transparency. UCRET operates fully zero-shot, without any fine-tuning, making it flexible and accessible across different domains. To the best of our knowledge, UCRET is the first framework to employ spherical k-means for unsupervised, claim-driven column selection. This approach precisely identifies the most relevant evidence columns without requiring labels or fine-tuning before reasoning.

Extensive experiments on PubHealthTab and SCITAB demonstrate that UCRET consistently surpasses both open-source and closed-source models such as GPT-3.5-turbo, with an accuracy improvement of up to 17.8%, while narrowing the performance gap with GPT-4 to less than 2.7%. Furthermore, ablation studies underscore the critical role of both Column Filtering and Counterfactual Explanation, confirming their essential contribution to achieving high accuracy and interpretability.

In summary, the main contributions of this chapter are as follows:

- We introduce UCRET, a novel, fully modular, zero-shot framework for table-based fact verification with structured reasoning that combines unsupervised, claim-driven column selection with multi-perspective counterfactual explanation, all without fine-tuning, pretraining, or closed-source models.
- UCRET works with any open-source LLM, requires no fine-tuning or proprietary models, and can be easily adapted to new domains or downstream applications.
- Extensive experiments on PubHealthTab and SCITAB, along with ablation studies, demonstrate that both Column Filtering and Counterfactual Explanation are essential, complementary components for achieving high accuracy and interpretability.

The rest of this section has been organized as follows. Section 3.2 provides a brief overview of previous works on tabular information processing and the use of large language models (LLMs) for table reasoning. Section 3.3 describes the UCRET framework in detail. Section 3.4 presents empirical results and comparisons with other state-of-the-art methods. Section 3.5 discusses error analysis, and Section 3.6 concludes the work and outlines future directions.

3.2 Related Works

Table-based reasoning tasks, including question answering, fact verification, and summary generation, require models to interpret structured tabular data and perform precise inference. Early work addressed these challenges through symbolic approaches, such as translating natural language into executable queries (e.g., SQL, SPARQL) [104, 105], or using graph neural networks to model table structure [106, 107]. However, these approaches often struggled to generalize due to their dependence on schema-specific formats and handcrafted operations. Pretrained neural methods such as TAPAS [108] extend BERT with table-aware embeddings and achieve strong

results on cell selection and aggregation, typically under weak supervision, and TaBERT [109] jointly pretrains over text–table pairs to improve semantic parsing and QA via task-specific fine-tuning. However, both lines generally rely on large-scale pretraining and labeled adaptation, limiting plug-and-play transfer to new domains. These limitations highlight the motivation for a new approach that enhances flexibility and transparency across diverse domains.

With the emergence of large language models, recent methods have explored language-driven table reasoning. Several studies introduced table pretraining strategies that align textual and tabular representations to improve LLMs' ability to reason over semi-structured inputs [39, 110, 111]. Building on this, Ye et al. [111] proposed a decomposition-based strategy that simplifies reasoning by breaking down complex tables and questions into subcomponents. Chain-of-Table [112] incorporates table structures into chain-of-thought prompting, while ReAcTable [113] integrates reactive and proactive planning to handle complex multi-step reasoning.

Despite these advances, many existing methods rely primarily on textual reasoning, which can be insufficient for tasks that require accurate numerical comparison or structural table understanding. To improve flexibility and performance, hybrid systems have emerged that dynamically assign subtasks to symbolic (e.g., SQL) or textual reasoning modules depending on the query [113, 114]. However, these systems still face challenges in identifying the most relevant table columns and providing interpretable justifications for their decisions. On the other hand, Liu et al. [115] proposed Multi-aspect Adversarial Contrastive Learning (Macol), which leverages adversarial examples generated from multi-aspect reasoning to train a tabular-textual encoder via contrastive learning. Although Macol achieved promising results on three Wikipedia-based table fact-checking datasets, it requires substantial computational resources and struggles to generalize to diverse datasets due to its reliance on multi-aspect reasoning rules. To address these limitations, we propose UCRET, a zero-shot, unsupervised pipeline that uses spherical k-means for claim-driven column filtering and open-source LLMs for interpretable reasoning, providing flexibility and transparency without dependence on task-specific fine-tuning. Our method explicitly targets the dual goals of precision and interpretability, aiming to bridge the gap between symbolic and textual reasoning in table-based fact verification.

3.3 Methodology

3.3.1 Problem Formalization

We define the task of table-based fact verification as a structured reasoning problem where, given a natural language claim C and a table $\mathcal{D}$, the goal is to predict a label $V \in \{\texttt{SUPPORTS}, \texttt{REFUTES}, \texttt{NOT ENOUGH INFORMATION}\}$. Formally, we denote our framework as a function $g_\phi(\cdot)$ with parameters ϕ as follows:

$$V = g_\phi(C, \mathcal{D}) \tag{3.1}$$

where $\mathcal{D}$ consists of a descriptive caption S and its content $\{D_{i,j} \mid 1 \leq i \leq R_\mathcal{D},\ 1 \leq j \leq C_\mathcal{D}\}$, where $R_\mathcal{D}$ and $C_\mathcal{D}$ correspond to the total number of rows and columns, respectively. Each cell (i, j) contains a data value $D_{i,j}$. Figure 3.1 illustrates the input and output structure of our framework.

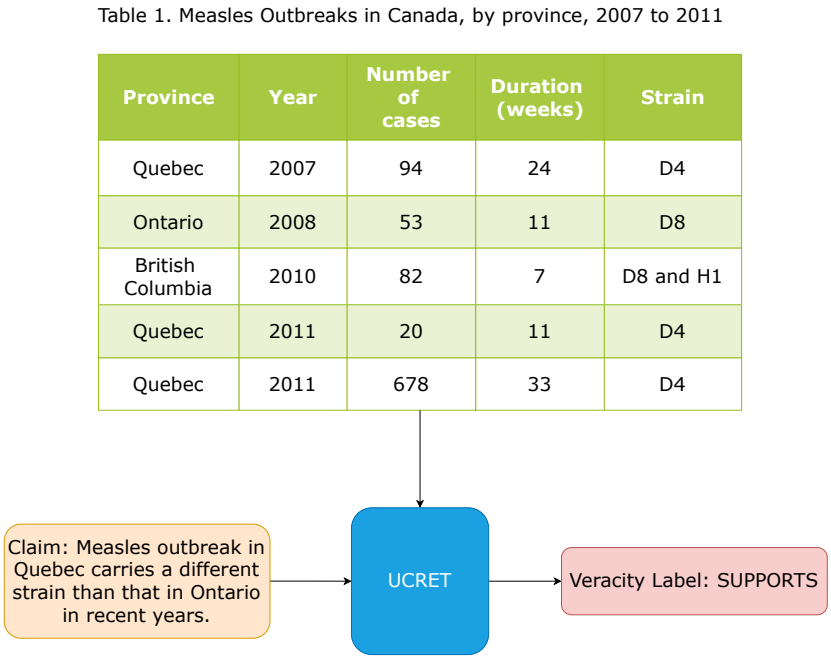


Figure 3.1: An example of the input and output for UCRET.

Despite recent progress, most existing methods for table-based fact verification still face significant challenges in real-world scenarios. These models often struggle to identify and focus on the most relevant evidence

within complex, noisy tables, and their reasoning is frequently limited to a single perspective or label. This limitation can lead to incomplete, biased decision processes, especially when evaluating complex or unclear claims that need a deeper, more detailed examination of the table contents. To systematically address the problems of extracting relevant evidence and enabling faithful, multi-perspective reasoning over tabular data, we propose **UCRET**, the zero-shot framework that operates without fine-tuning or supervision. UCRET is designed to solve two critical problems: (i) selecting claim-relevant evidence from noisy tables and (ii) constructing structured, contrastive reasoning paths that enable the model to reason systematically across multiple veracity hypotheses.

As illustrated in Figure 3.2, UCRET decomposes the verification task into three tightly connected stages: Column Filtering, Counterfactual Chain-of-Explanation, and Answer Generation. This decomposition allows UCRET to isolate the evidence selection, explanation generation, and decision-making stages, enabling interpretable and generalizable reasoning over tabular data. Unlike prior work that relies on closed-source models or fine-tuning on curated datasets, UCRET operates entirely in a zero-shot setting and demonstrates robustness across both medical and scientific domains.

The combination of these three stages is critical for effective table-based reasoning. Column Filtering focuses the model's attention on semantically relevant evidence, reducing noise and irrelevant distractions. Counterfactual Chain-of-Explanation (CCoE) prompts the model to systematically explore all veracity possibilities, encouraging structured, faithful reasoning rather than biased or shallow inference. Finally, Answer Generation introduces a reflective verification step, comparing which of the generated explanations is most likely, leading to the final inference of the veracity label V. This modular decomposition enables UCRET to isolate and address the core challenges of table-based verification—evidence selection, reasoning generation, and prediction validation—leading to more accurate, interpretable, and robust fact verification across domains.

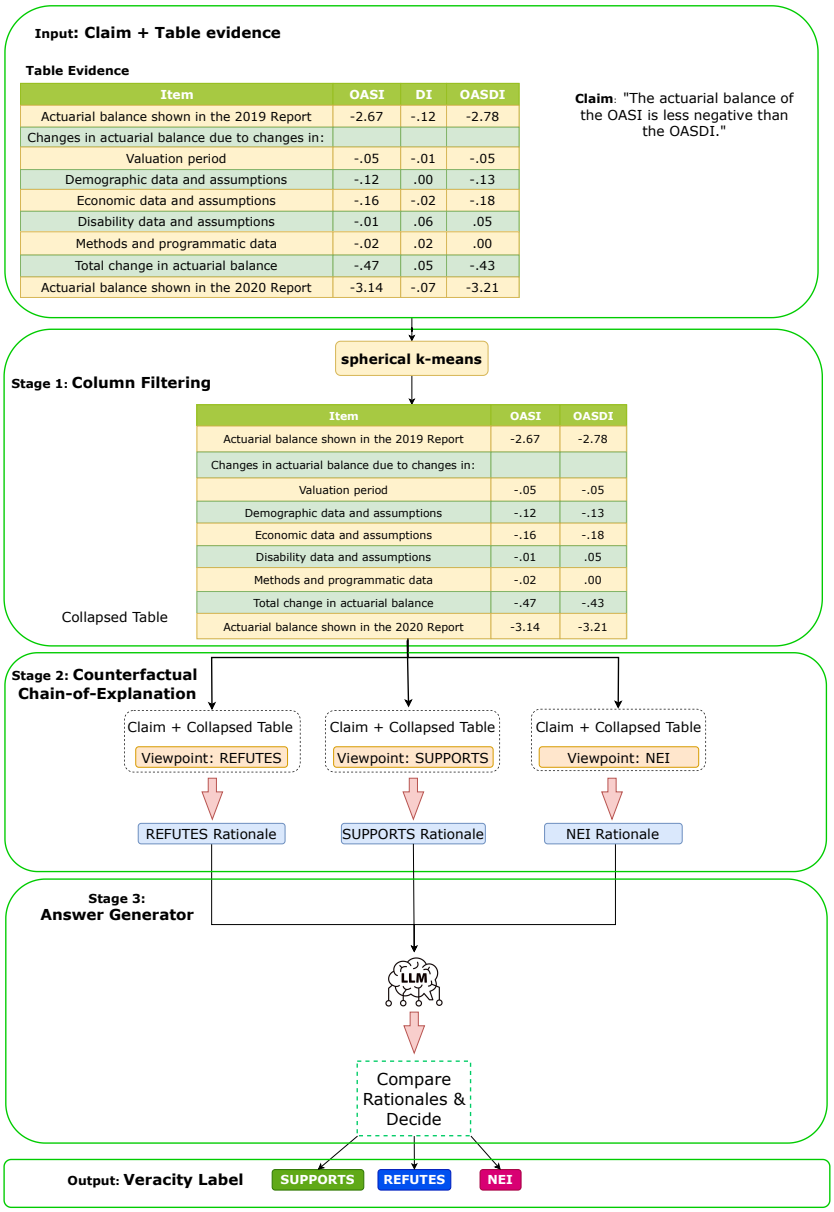


Figure 3.2: An overall framework of UCRET, which contains three main modules: (i) Column Filtering, which selects columns that are relevant to the query to make a collapsed evidence table; (ii) Counterfactual Chain-of-Explanation, which offers detailed reasoning, evaluating how effectively each explanation aligns with the claim by highlighting both its strengths and limitations; and (iii) Answer Generator, which produces the final decision of the claim.

Stage 1: Column Filtering Tables often contain irrelevant or distracting information that is unrelated to the specific claim, which can hinder precise reasoning. To address this challenge, we introduce an unsupervised column filtering module that applies spherical k-means clustering to automatically select only the columns most semantically aligned with the claim. By narrowing the context to the most informative columns and their corresponding row values, this stage minimizes the risk of introducing irrelevant evidence that could mislead the model. The output is a collapsed evidence table $D_{\text{collapsed}}$ that preserves only the most informative content needed for claim verification. This automated filtering not only enhances interpretability and accuracy but also makes the framework inherently adaptable to diverse and previously unseen table formats.

Stage 2: Counterfactual Chain-of-Explanation In this stage, the Counterfactual Chain-of-Explanation module evaluates the relationship between the input claim and the collapsed evidence table $D_{\text{collapsed}}$ generated in Stage 1. The language model is prompted to generate three distinct explanations—one each under the assumptions that the evidence supports, refutes, or does not provide enough information for the claim. Each explanation directly corresponds to a specific veracity label: `SUPPORTS`, `REFUTES`, or `NOT ENOUGH INFORMATION`. For every scenario, the model is instructed to reason explicitly about the strengths and weaknesses of the connection between the evidence and the claim.

Unlike traditional chain-of-thought prompting, which typically produces a single explanation for the most likely label, our counterfactual approach systematically requires the model to consider all possible outcomes. By examining the evidence from multiple perspectives, the model reduces the risk of bias or premature conclusions. This comprehensive, multi-scenario analysis results in more robust predictions and generates explanations that are clearer and more transparent, greatly improving interpretability and trustworthiness. Building on the claim-focused evidence selection in Stage 1, this structured reasoning approach enables careful, interpretable evaluation for each veracity scenario, helping to ensure that no relevant perspective is overlooked.

Stage 3: Answer Generator In the final stage, the Answer Generator module plays a pivotal role in the claim verification process. At this stage, an LLM is instructed to analyze the strengths and weaknesses of each explanation concerning the claim and the collapsed evidence table. By performing this introspective verification, the model selects the label

whose supporting explanation is most coherent and plausible. This reflective deliberation enforces consistency between the evidence, the explanation, and the final prediction, improving robustness and transparency in the claim verification.

3.3.2 The UCRET framework

We next detail UCRET's components and how they interact within the pipeline. Key novelty of UCRET in the Column Filtering stage: we use claim-driven spherical k-means to collapse the table into a compact evidence view before any LLM reasoning, which improves both accuracy and auditability. Figure 3.2 illustrates the full overview of the **UCRET** framework.

STAGE 1: Column Filtering

Tables often contain irrelevant or distracting information that is unrelated to the specific claim, which can hinder precise reasoning. To address this challenge, we propose an unsupervised column filtering module that leverages spherical k-means clustering on embeddings. This approach captures the underlying semantic structure of columns by grouping them based on their similarity in embedding space. Our method applies a two-step process to automatically select columns most relevant to the claim: first, it applies lexical similarity for initial filtering, then refines the selection using semantic clustering with spherical k-means for precise alignment. By narrowing the context to the most informative columns and their corresponding row values, this stage minimizes the risk of introducing irrelevant evidence that could mislead the model. The output is a collapsed evidence table $\mathcal{D}_{\text{collapsed}}$ that preserves only the most informative content needed for claim verification. This automated filtering not only enhances interpretability and accuracy but also makes the framework inherently adaptable to diverse and previously unseen table formats. A complete specification of Stage 1 is presented in Algorithm 3.1.

Input notes. For each column $j \in \mathcal{J}$, the *column token set* T_j is the set of normalized tokens from (i) the header and (ii) non-empty cells. Let T_C be the claim token set. Column embeddings $\mathbf{x}_j$ are built from header and data cells; $\mathbf{u}_C$ is the claim embedding. Cosine is computed on ℓ_2-normalized vectors.

Algorithm 3.1 Lexical Pre-filter + Spherical k-means for Column Filtering

Input: claim C; table $\mathcal{D}$ with $C_{\mathcal{D}}$ columns; column index set $\mathcal{J} = \{1, \ldots, C_{\mathcal{D}}\}$; claim tokens T_C; column token sets $\{T_j\}_{j \in \mathcal{J}}$; column embeddings $\{\mathbf{x}_j\}_{j \in \mathcal{J}}$; claim embedding $\mathbf{u}_C$; Jaccard threshold τ; clusters k; cosine threshold θ

Step 1: Lexical pre-filter
$S_{\text{lex}} \leftarrow \left\{ j \in \mathcal{J} : J(j) = \frac{|T_C \cap T_j|}{|T_C \cup T_j|} \geq \tau \right\}$
$R \leftarrow \mathcal{J} \setminus S_{\text{lex}}$
if $R = \varnothing$ **then**
    return $\mathcal{D}_{\text{collapsed}}$ with columns S_{lex}
end if
Step 2: Spherical k-means on R
Normalize all vectors: $\tilde{\mathbf{x}}_j \leftarrow \mathbf{x}_j / \|\mathbf{x}_j\|_2 \ \forall j \in \mathcal{J}$; $\tilde{\mathbf{u}}_C \leftarrow \mathbf{u}_C / \|\mathbf{u}_C\|_2$
Work set: use $\{\tilde{\mathbf{x}}_j : j \in R\}$ for clustering
$k' \leftarrow \min(k, \ |R|)$
Initialize $\{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{k'}\}$ by spherical k-means++ on $\{\tilde{\mathbf{x}}_j : j \in R\}$
repeat
    Assignment: $a(j) \leftarrow \arg\max_{t \in \{1,\ldots,k'\}} \langle \tilde{\mathbf{x}}_j, \boldsymbol{\mu}_t \rangle \ \ \forall j \in R$
    Update: **for** $t = 1$ **to** k' **do**
        $X_t \leftarrow \{\tilde{\mathbf{x}}_j : a(j) = t\}$
    if $X_t = \varnothing$ **then**
            re-seed $\boldsymbol{\mu}_t \leftarrow$ spherical k-means++ sample from $\{\tilde{\mathbf{x}}_j : j \in R\}$
    else
            $\boldsymbol{\mu}_t \leftarrow \sum_{\mathbf{x} \in X_t} \mathbf{x}$; $\boldsymbol{\mu}_t \leftarrow \boldsymbol{\mu}_t / \|\boldsymbol{\mu}_t\|_2$
    end if
    end for
until no assignment changes
Pick claim-aligned cluster: $s_t \leftarrow \langle \boldsymbol{\mu}_t, \tilde{\mathbf{u}}_C \rangle$ for t=1..k'; $t^\star \leftarrow \arg\max_t s_t$
$S_{\text{sem}} \leftarrow \begin{cases} \{\, j \in R : a(j) = t^\star \,\}, & \text{if } s_{t^\star} \geq \theta \\ \varnothing, & \text{otherwise} \end{cases}$
Step 3: Merge
$S \leftarrow S_{\text{lex}} \cup S_{\text{sem}}$
if $|S| = 0$ **then**
    $S \leftarrow \textsc{Top2}\big(\{(\langle \tilde{\mathbf{u}}_C, \tilde{\mathbf{x}}_j \rangle, j) : j \in \mathcal{J}\}\big)$
end if
return $\mathcal{D}_{\text{collapsed}}$ with columns S

STAGE 2: Counterfactual Chain-of-Explanation

As stated in [122], counterfactual thoughts significantly influence cognition, emotion, and social perception. People frequently generate counterfactual scenarios about how past events might have unfolded differently, often imagining better outcomes rather than worse ones under different conditions. Counterfactual reasoning is considered "human-friendly" because its contrastive and selective nature mirrors the way people naturally reason about cause and effect when attempting to understand the causal structure of events [123, 124].

In this stage, we adopt a form of counterfactual reasoning tailored for table-based claim verification. Given a claim C and a collapsed evidence table $\mathcal{D}_{collapsed}$, the model generates a counterfactual chain of explanation to evaluate how the evidence may support, refute, or fail to verify the claim. Specifically, for each label, `SUPPORTS`, `REFUTES`, or `NOT ENOUGH INFORMATION`, the language model is asked to generate an explanation conditioned on that assumption: (1) e_S, assuming the table supports the claim; (2) e_R, assuming that the table refutes the claim; (3) e_{NEI}, assuming the table provides insufficient information. To be clear, each explanation is generated by a parameterized probabilistic model p_θ through a maximized likelihood process as:

$$p_\theta\big(e_{\mathrm{S}} \mid y = \mathrm{S},\, C,\, \mathcal{D}_{collapsed}\big) \;=\; \prod_{i=1}^{L} p_\theta\big(t_i^{(\mathrm{S})} \mid t_{<i}^{(\mathrm{S})},\, y = \mathrm{S},\, C,\, \mathcal{D}_{collapsed}\big) \quad (3.2)$$

$$p_\theta\big(e_{\mathrm{R}} \mid y = \mathrm{R},\, C,\, \mathcal{D}_{collapsed}\big) \;=\; \prod_{i=1}^{L} p_\theta\big(t_i^{(\mathrm{R})} \mid t_{<i}^{(\mathrm{R})},\, y = \mathrm{R},\, C,\, \mathcal{D}_{collapsed}\big) \quad (3.3)$$

$$p_\theta\big(e_{\mathrm{NEI}} \mid y = \mathrm{NEI},\, C,\, \mathcal{D}_{collapsed}\big) \;=\; \prod_{i=1}^{L} p_\theta\big(t_i^{(\mathrm{NEI})} \mid t_{<i}^{(\mathrm{NEI})},\, y = \mathrm{NEI},\, C,\, \mathcal{D}_{collapsed}\big) \quad (3.4)$$

where L is the length of an explanation.

To systematically generate these explanations, we design Chain-of-Explanation (CoE) prompts instructing the model to reason step-by-step explicitly. For instance, the CoE prompt for the `SUPPORTS` label is as follows:

```
You are an expert for fact verification.
Your task is to explain why the given claim is supported by the
    evidence in the table representation.
```

```
Claim: {claim}
Table Representation:
{table_representation}

Step-by-step, identify the relevant facts, data points, or rows in
    the table that directly confirm or strongly align with the claim
    .
Clearly explain how each piece of evidence contributes to verifying
    the claim.
Your explanation should be logical, precise, and grounded in
    specific table content.
```

In addition, for the `REFUTES` label, an LLM is instructed as follows:

```
You are an expert for fact verification.
Your task is to explain why the claim is refuted by the content of
    the given evidence table.

Claim: {claim}
Table Representation:
{table_representation}

Step-by-step, identify which parts of the table contradict or
    disprove the claim.
Point out any inconsistencies, conflicting data, or logical
    mismatches between the claim and the evidence.
Your explanation should be clear and rooted in specific rows or data
     from the table.
```

Lastly, for the `NOT ENOUGH INFORMATION (NEI)` label, an LLM receives the following prompt:

```
You are an expert for fact verification.
Your task is to explain why the evidence table does not provide
    enough information to verify or refute the claim.

Claim: {claim}
Table Representation:
{table_representation}

Analyze each row of the table step-by-step, explaining exactly how
    the statements and their contexts fail to provide sufficient
    data to either support or refute the claim.
```

This counterfactual approach enables the model to explore multiple plausible interpretations of the evidence for the claim and to reflect on the reasoning process under each scenario. These explanations serve as intermediate reasoning artifacts that enhance interpretability and guide the model toward a more accurate final prediction in downstream stages.

To illustrate Stage 2, we provide an instantiated prompt that is directly used as the input to the LLM for generating an explanation for `SUPPORT` label. The prompt explicitly includes (i) the natural-language claim and (ii) the collapsed evidence table produced by the column filtering stage, so that the model is constrained to reason over a compact, relevance-preserving representation rather than the full raw table. By instructing the LLM to identify concrete rows and values and to explain their contributions in a step-by-step manner, this example clarifies how our pipeline operationalizes evidence grounding and produces a verifiable rationale aligned with the selected evidence.

```
You are an expert for fact verification.
Your task is to explain why the given claim is supported by the
    evidence in the table representation.

Claim: "The actuarial balance of the OASI is less negative than the
    OASDI."

Table Representation:
[Collapsed Evidence Table: Actuarial Balance (OASI vs. OASDI)]
| Item | OASI | OASDI |
|-------------------------------------------------|-------|-------|
| Actuarial balance shown in the 2019 Report | -2.67 | -2.78 |
| Changes in actuarial balance due to changes in: | | |
| - Valuation period | -0.05 | -0.05 |
| - Demographic data and assumptions | -0.12 | -0.13 |
| - Economic data and assumptions | -0.16 | -0.18 |
| - Disability data and assumptions | -0.01 | 0.05 |
| - Methods and programmatic data | -0.02 | 0.00 |
| Total change in actuarial balance | -0.47 | -0.43 |
| Actuarial balance shown in the 2020 Report | -3.14 | -3.21 |

Step-by-step, identify the relevant facts, data points, or rows in
    the table that directly confirm or strongly align with the claim
    .
Clearly explain how each piece of evidence contributes to verifying
    the claim.
Your explanation should be logical, precise, and grounded in
```

```
   specific table content.
```

STAGE 3: Answer Generator

A function f_{AG} takes $C, \mathcal{D}_{collapsed}, e_S, e_R, e_{NEI}$, and generates a final answer veracity label V. Mathematically, the function f_{AG} representing the Answer Generator (AG) module, is expressed as follows:

$$V = f_{AG}(C, \mathcal{D}_{collapsed}, e_S, e_R, e_{NEI}) \tag{3.5}$$

Similar to the CCoE generation stage, the predicted answer V is generated by a parameterized probabilistic model p_θ through a maximized likelihood process as:

$$p_\theta(V|C, \mathcal{D}_{collapsed}, e_S, e_R, e_{NEI}) = \prod_{i=1}^{N} p_\theta(V_i|C, \mathcal{D}_{collapsed}, e_S, e_R, e_{NEI}, V_{<i}) \tag{3.6}$$

where N is the length of the predicted answer.

Our prompting strategy provides LLMs with the ability to deliberate on their responses, enabling them to identify and correct errors through introspection. Specifically, this stage draws inspiration from Chain of Verification (CoVe) prompting, introduced by [125]. Building on this idea, we introduce a variant called Counterfactual Chain of Verification (CCoV) prompting. In this approach, we use the counterfactual explanations e_S, e_R, and e_{NEI} generated in Stage 2 as baseline responses for verification. An LLM is instructed to examine the strengths and weaknesses of each explanation concerning the claim C and the collapsed evidence table $D_{\text{collapsed}}$, and to select the most rational label. By incorporating both structured tabular evidence and natural language reasoning, this method enhances decision quality and improves model interpretability in claim verification tasks. Notably, our CCoV enables LLMs to introspectively evaluate the counterfactual explanations they generate, refining their outputs through a process of iterative verification and correction. This not only strengthens the model's reasoning capabilities but also fosters greater transparency in its decision-making process.

To systematically generate the final answer, we design our prompt to introspect upon the counterfactual explanations generated with C, $\mathcal{D}_{collapsed}$, and e_S, e_R, e_{NEI} as follows:

```
You are an expert fact-checker.
```

```
You will be given:
    - A claim.
    - Evidence presented in a table format.
    - Three different explanations concerning how this evidence
    relates to the claim.
        - One assumes the evidence SUPPORTS the claim,
        - One assumes it REFUTES the claim,
        - One assumes it is NOT ENOUGH INFORMATION.

Your task is to evaluate whether the provided claim is supported,
    refuted, or lacks sufficient information based on the given
    table evidence and explanation.

Input:
    Claim: {claim}
    Evidence: {collapsed_evidence_table}
    Explanation (SUPPORTS): {e_S}
    Explanation (REFUTES): {e_R}
    Explanation (NEI): {e_NEI}

Task:
    1. Analyze the claim, evidence, and explanation.
    2. Determine if the explanation and evidence together:
        - Support the claim (confirm it as true).
        - Refute the claim (contradict or disprove it).
        - Lack sufficient information (do not provide enough data to
     confirm or refute the claim).
    3. Make a final decision about the claim's veracity based on
    this comparison.

Answer with only one label: NOT ENOUGH INFORMATION, REFUTES, or
    SUPPORTS.
```

To illustrate Stage 3, we provide an example prompt in which the LLM is asked to act as an expert fact-checker and output a single veracity label. The prompt consists of (i) a claim, (ii) a collapsed evidence table produced by the preceding column filtering stage, and (iii) three candidate explanations generated under the mutually exclusive hypotheses that the evidence SUPPORTS, REFUTES, or provides NOT ENOUGH INFORMATION for the claim. Given these inputs, the LLM is instructed to compare the claim against the tabular evidence while assessing the internal consistency and evidential grounding of each explanation and then return exactly one label among SUPPORTS, REFUTES, NOT ENOUGH INFORMATION. This

example demonstrates how Stage 3 consolidates evidence and counterfactual rationales into a final, decision-oriented prediction.

```
You are an expert fact-checker.

You will be given:
    - A claim.
    - Evidence presented in a table format.
    - Three different explanations concerning how this evidence
    relates to the claim.
        - One assumes the evidence SUPPORTS the claim,
        - One assumes it REFUTES the claim,
        - One assumes it is NOT ENOUGH INFORMATION.

Your task is to evaluate whether the provided claim is supported,
    refuted, or lacks sufficient information based on the given
    table evidence and explanation.

Input:
    Claim: "The actuarial balance of the OASI is less negative than
    the OASDI."

    Evidence (Collapsed Table):
    | Item | OASI | OASDI |
    | Actuarial balance shown in the 2019 Report | -2.67 | -2.78 |
    | Changes in actuarial balance due to changes in: | | |
    | Valuation period | -0.05 | -0.05 |
    | Demographic data and assumptions | -0.12 | -0.13 |
    | Economic data and assumptions | -0.16 | -0.18 |
    | Disability data and assumptions | -0.01 | 0.05 |
    | Methods and programmatic data | -0.02 | 0.00 |
    | Total change in actuarial balance | -0.47 | -0.43 |
    | Actuarial balance shown in the 2020 Report | -3.14 | -3.21 |

    Explanation (SUPPORTS):
    "The claim states that OASI is 'less negative' than OASDI. In
    the collapsed table, OASI has an actuarial balance of -2.67 in
    the 2019 Report, while OASDI is -2.78. Since -2.67 is closer to
    zero than -2.78, OASI is less negative in 2019. The same pattern
     holds in the 2020 Report: OASI is -3.14 and OASDI is -3.21, and
     -3.14 is closer to zero than -3.21, so OASI is again less
    negative. Therefore, the table evidence directly supports the
    claim."
```

```
    Explanation (REFUTES):
    "Although the 2019 and 2020 headline actuarial balances appear
    close, the table shows that the total change in actuarial
    balance is -0.47 for OASI versus -0.43 for OASDI. This indicates
     OASI deteriorates more than OASDI over the period (a larger
    negative change). Since OASI becomes worse relative to OASDI in
    the changes, the claim that OASI is less negative than OASDI is
    contradicted by the overall direction of change in the evidence.
    "

    Explanation (NEI):
    "The table provides numerical actuarial balances, but it does
    not define the precise evaluation criterion for 'less negative'
    (e.g., whether it refers to headline balances only, absolute
    magnitude, or a specific actuarial horizon). It also does not
    provide uncertainty estimates or methodological context to judge
     whether the differences between OASI and OASDI are
    substantively meaningful. Because the evidence lacks definitions
     and context needed to interpret 'less negative' rigorously,
    there is not enough information to definitively support or
    refute the claim."

Task:
    1. Analyze the claim, evidence, and explanation.
    2. Determine if the explanation and evidence together:
        - Support the claim (confirm it as true).
        - Refute the claim (contradict or disprove it).
        - Lack sufficient information (do not provide enough data to
     confirm or refute the claim).
    3. Make a final decision about the claim's veracity based on
    this comparison.

Answer with only one label: NOT ENOUGH INFORMATION, REFUTES, or
    SUPPORTS.
```

In conclusion, we explicitly position UCRET as a knowledge-based decision-support system: the collapsed evidence table $\mathcal{D}_{\text{collapsed}}$ serves as a structured knowledge slice, the per-label counterfactual chains e_S, e_R, e_{NEI} are explicit knowledge artifacts for explanation, and the answer generator applies a verification operator over these artifacts to produce a justified decision.

3.4 Experiment Results

3.4.1 Empirical Preparation

We tune the number of clusters k for the column-filtering stage over the range $k \in \{2, 3, 4\}$, while also searching the cosine similarity threshold $\theta \in \{0.55, 0.60, 0.65\}$ and the Jaccard threshold $\tau \in \{0.20, 0.25, 0.30\}$. Across our experiments, $k = 2$ yields the best performance for clustering in Column Filtering. For the thresholds, the optimal setting is $\theta = 0.60$ and $\tau = 0.25$. We use NVIDIA A6000 GPU with 48GB memory for all experiments. Semantic embeddings for the claim and headers are generated using the BGE model [126], which produces 1024-dimensional vectors.

We use fully open-source instruction-tuned LLMs as backbones for the counterfactual explanation and answer generation stages: Llama-3.3-70B-Instruct [127], Qwen2.5-14B-Instruct, and Qwen2.5-72B-Instruct [128]. This selection aligns with our zero-shot, open-model goal while spanning capacities from 14B to 72B, allowing us to probe robustness across model sizes. Instruction tuning improves adherence to our structured prompts, and all models are implemented with 4-bit quantization [129] and set the *do_ sample* to False to prevent unexpected generation from the LLMs. Table 3.1 details model configurations used in our empirical study on UCRET.

Table 3.1: Model configurations

Task	**Model name**	**Pre-trained**	**Configuration**
Text Embedding	Flag Embedding	BAAI/bge-m3	default configuration
CCoE	Llama3.3	Llama-3.3-70B-Instruct	do_sample=False, max_new_tokens=1024
	Qwen2.5	Qwen2.5-14B-Instruct Qwen2.5-72B-Instruct	
Answer Generation	Llama3.3	Llama-3.3-70B-Instruct	do_sample=False, max_new_tokens=32
	Qwen2.5	Qwen2.5-14B-Instruct Qwen2.5-72B-Instruct	

3.4.2 Datasets

We select two existing tabular benchmarks to evaluate the performance of our proposed framework: PubHealthTab [130] and SCITAB [131], which focus on tables from scientific papers and public health articles, respectively. To ensure fair comparison, we follow prior work by reporting only test-set results and using accuracy as the primary metric, since the baselines assess

both datasets solely with accuracy. As shown in Figure 3.3, both datasets have a majority of samples labeled as support, with the remaining examples more evenly split between NEI and refute. SCITAB contains more samples than PubHealthTab, but the overall label distributions are similar across both datasets.

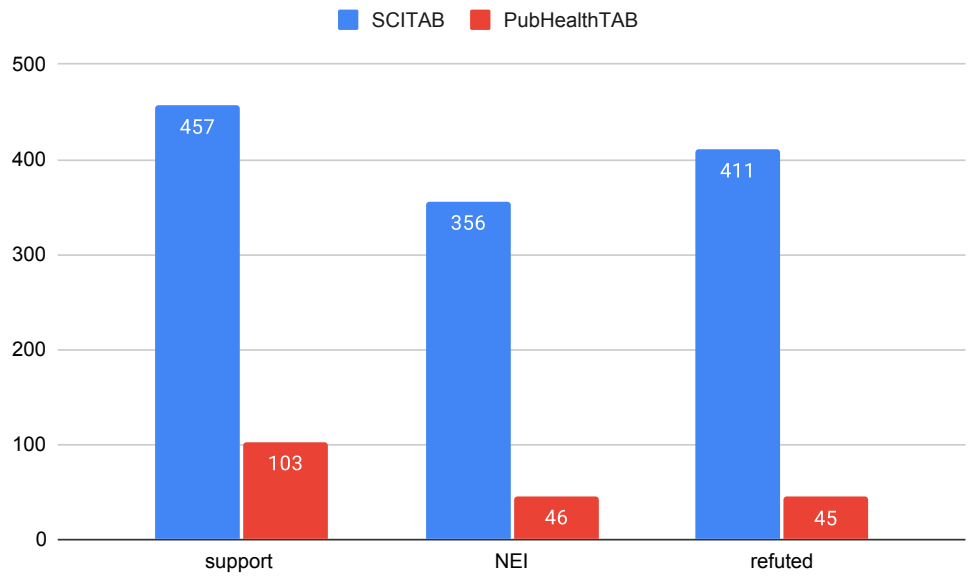


Figure 3.3: Label distribution in the SCITAB and PubHealthTab.

In addition, we analyze the table sizes in SCITAB and PubHealthTab based on the number of rows and columns. According to Table 3.2, the average table size in SCITAB is 7×6 (rows × columns), compared to 7×3 for PubHealthTab. The maximum table sizes reach up to 32×16 and 18×11 for SCITAB and PubHealthTab, respectively, demonstrating significant table complexity. Since LLMs process tables via serialization, these substantial table dimensions challenge their understanding and reasoning on tabular data. Therefore, SCITAB and PubHealthTab are suitable datasets for evaluating our proposed UCRET framework and demonstrating its efficiency in table-based fact verification tasks.

3.4.3 Experiment Results

We evaluate the performance of UCRET against several competitive baselines on two tabular fact verification benchmarks: PubHealthTab [130] and SCITAB [131]. For a baseline comparison, we evaluate our method against

Table 3.2: Statistics about the table size on SCITAB and PubHealthTab datasets.

	SCITAB	**PubHealthTab**
Domains	Scientific Articles	Wikipedia and medical sources
Num. samples	1,224	194
max column	16	11
max row	32	18
min column	2	0
min row	1	0
mode row	5	2
mode column	3	7
average column	6.11	3.43
average row	7.52	7.15

several recent state-of-the-art approaches. TART [42] is included due to its integration of external tool invocation within the chain-of-thought reasoning framework. We also consider Table-LLaVA [43] and PROTRIX [44], both of which employ hybrid reasoning strategies that combine unstructured text understanding with structured table parsing, making them particularly relevant for fact-checking and complex inference tasks. In addition, we compare with MACT [45], a multi-agent system equipped with tool-using capabilities that operates without reliance on closed-source large language models and does not require task-specific fine-tuning. Finally, we include SynTab-LLaVA [46], a hybrid multi-resolution multimodal model engineered to enhance the understanding of both textual content and structural details within table images.

As shown in Table 3.3, UCRET is competitive among prior open-source models and rivals closed-source methods, all while operating in a zero-shot setting with no fine-tuning. From the empirical results, UCRET surpasses strong baselines such as Table-LLaVA [43], ProTrix [44], and MACT [45], particularly on SCITAB, which presents more complex scientific reasoning. Notably, UCRET also outperforms TART [42] when it uses GPT-3.5-turbo, a much larger closed-source model. While GPT-4-based TART achieves the highest overall accuracy on PubHealthTab at 84.10%, our UCRET framework secures a commendable second-best score of 81.41% using only the open-source Llama-3.3-70B-Instruct model. These results highlight UCRET's strong performance, transparency, and cost efficiency, achieving competitive results without relying on heavy fine-tuning or proprietary systems. It is noteworthy that GPT-4 boasts approximately 1 trillion parameters [132], significantly outnumbering the parameters of our open-source model. Despite

this disparity, UCRET narrows the performance gap with GPT-4, underscoring the efficiency and effectiveness of our proposed framework. Furthermore, UCRET's reliance on open-source systems ensures transparency and cost-efficiency, making it a robust alternative for tabular fact-checking tasks.

Table 3.3: Accuracy comparison across baselines and UCRET variants using different LLM backbones on the PubHealthTab and SCITAB datasets. – denotes results not reported from prior publications. Bold indicates the best result; underline indicates the second best. Our reproduced results are denoted with $*$

Approach	Method	PubHealthTab	SCITAB
Baselines	TART (GPT-3.5-turbo) [42]	63.60	59.30
	TART (GPT-4) [42]	**84.10**	**63.60**
	Table-LLaVA [43]	50.23 $*$	–
	ProTrix [44]	–	45.00
	MACT [45]	–	57.40
	SynTab-LLaVA [46]	68.02	–
Ours	*Qwen2.5-14B-Instruct*		
	UCRET w/o Column Filtering	53.09	34.49
	UCRET w/o CCoE	54.12	38.32
	UCRET	58.27	40.85
	Qwen2.5-72B-Instruct		
	UCRET w/o Column Filtering	70.10	42.31
	UCRET w/o CCoE	71.61	45.16
	UCRET	75.77	53.42
	Llama-3.3-70B-Instruct		
	UCRET w/o Column Filtering	74.25	54.66
	UCRET w/o CCoE	77.32	56.12
	UCRET	<u>81.41</u>	<u>60.88</u>

The success of UCRET stems from its modular two-stage design. The Column Filtering module identifies claim-relevant columns through k-means clustering, producing a condensed and focused evidence table. The CCoE module then generates structured, label-specific explanations under different veracity assumptions. Together, these components enable more accurate, interpretable inference. We also observe consistent gains across different LLM backbones, including Qwen2.5-14B-Instruct, Qwen2.5-72B-Instruct, and Llama-3.3-70B-Instruct, demonstrating the robustness and generalizability of the approach.

3.4.4 Ablation Study

To evaluate the individual contributions of UCRET's two core modules: Column Filtering and Counterfactual Chain-of-Explanation, we conduct ablation studies. These experiments are designed to verify the effectiveness of our novel components, specifically unsupervised column filtering and label-conditioned counterfactual explanations, while demonstrating their complementary benefits. According to Table 3.3, removing either component results in consistent performance degradation across all model backbones and datasets. On PubHealthTab, omitting Column Filtering leads to drops of 5.18 points with Qwen2.5-14B-Instruct (from 58.27 to 53.09), 5.67 points with Qwen2.5-72B-Instruct (from 75.77 to 70.10), and 7.16 points with Llama-3.3-70B-Instruct (from 81.41 to 74.25). This confirms the importance of semantic column selection in reducing noise and isolating relevant evidence. The impact is even more pronounced on SCITAB, where table density and numerical complexity are higher. Accuracy drops by 6.36 points (Qwen2.5-14B-Instruct), 11.11 points (Qwen2.5-72B-Instruct), and 5.22 points (Llama-3.3-70B-Instruct) when Column Filtering is removed, highlighting its crucial role in challenging scientific settings.

On the other hand, the CCoE module also contributes substantial gains by enriching multi-label reasoning. Removing CCoE reduces performance by 2.53 points (Qwen2.5-14B-Instruct), 8.26 points (Qwen2.5-72B-Instruct) on SCITAB, and 3.76 points (Llama-3.3-70B-Instruct) on SCITAB, where detailed explanation chains are essential. On Llama-3.3-70B-Instruct, performance drops from 60.88 to 56.12 without CCoE.

In summary, our ablation study results demonstrate that Column Filtering and CCoE play complementary roles: the former sharpens evidence selection, while the latter enhances reasoning depth, enabling UCRET to maintain strong accuracy and interpretability across medical and scientific domains.

3.4.5 Effect of the number of clusters k to Column Filtering

As shown in Figure 3.4, we study how the cluster k in Stage 1 for column filtering affects performance on PubHealthTab. In experiments conducted on a random 100-instance subset using Qwen2.5-72B-Instruct, accuracy peaks at k=2 (85.60%), with modest declines for k=3 (84.20%) and k=4 (83.60%). This trend supports the intuition that smaller k avoids over-segmenting semantically coherent, claim-relevant columns, thereby retaining salient evidence while still removing distractors. Larger k produces finer

partitions but can split relevant columns across clusters, diluting selection scores and slightly reducing answer quality. Based on this observation, we adopt k=2 as the default for PubHealthTab in our main experiments.

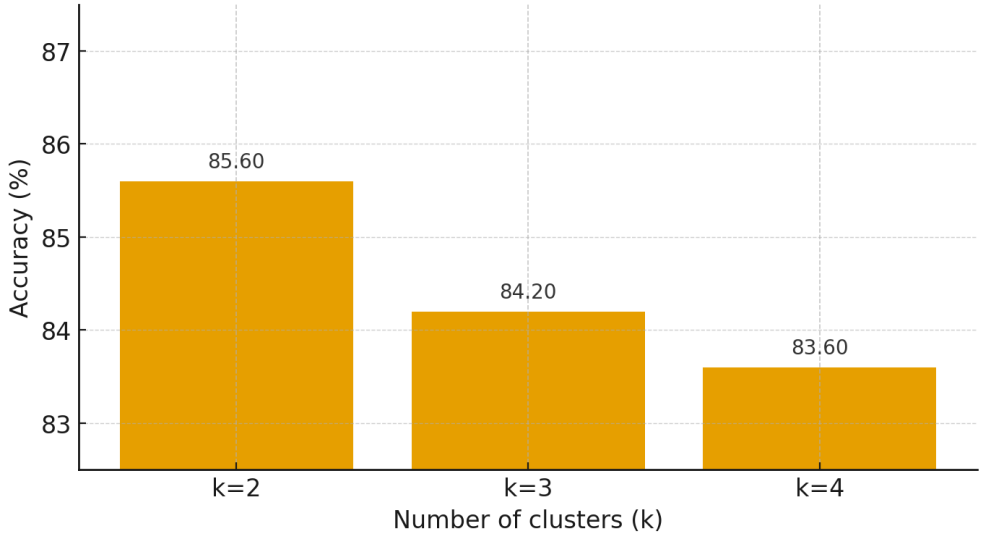


Figure 3.4: Sensitivity of column filtering to the number of clusters on PubHealthTab.

3.4.6 Effect of cosine threshold θ and Jaccard threshold τ to Column Filtering

In the experiment conducted on a random 100-instance subset using Qwen2.5-72B-Instruct, we assess nine (θ, τ) combinations with $k = 2$, where θ controls the cosine gate for the claim-aligned cluster, and τ is the Jaccard threshold in the lexical pre-filter. Accuracy peaks at θ=0.60, τ=0.25 with 85.8%. At θ=0.60, relaxing the lexical filter to τ=0.20 yields 85.4%, while tightening it to τ=0.30 gives 85.0%. Moving θ away from 0.60 degrades performance: with τ=0.25 accuracy is 85.2% at θ=0.55 and 85.6% at θ=0.65; with τ=0.20, 84.6% and 85.4%; with τ=0.30, 84.8% and 84.6%. Overall, a *moderate* cosine gate (θ=0.60) paired with a *mid-range* lexical threshold (τ=0.25) gives the best accuracy for the Column Filtering stage.

3.4.7 Computational Cost

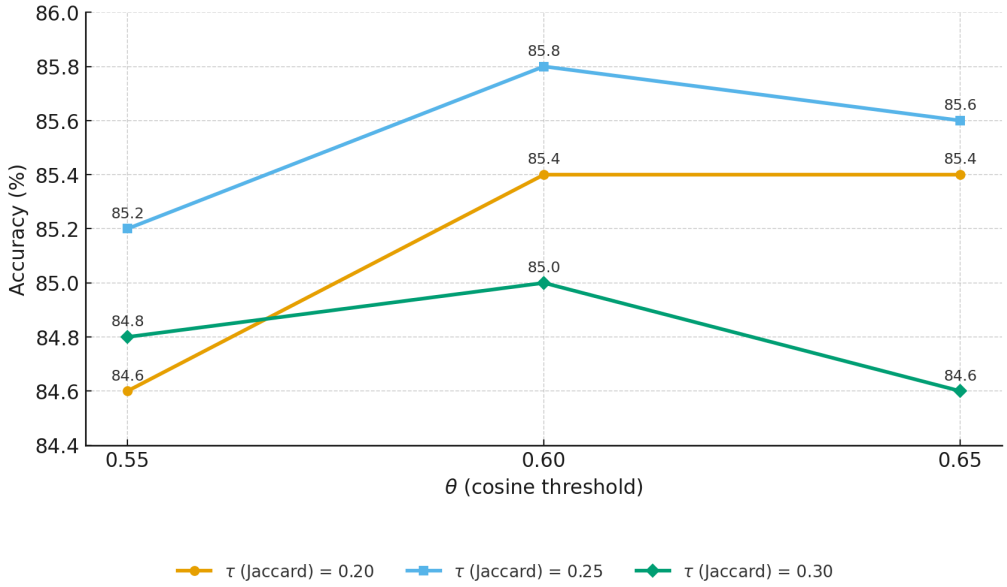


Figure 3.5: PubHealthTab sensitivity with fixed spherical k-means (k=2). X-axis: cosine threshold θ; lines: Jaccard threshold τ (lexical pre-filter).

To evaluate the computational efficiency of the UCRET framework across different large language models, this section analyzes the end-to-end runtime results presented in Table 3.4. The data reveals a clear trend: the Counterfactual Chain-of-Explanation and Answer Generation stages dominate the total runtime, accounting for the vast majority of the processing time. In contrast, the Column Filtering stage remains relatively quick and consistent across datasets. For instance, on PubHealthTab, the CCoE and Answer stages take between 73.30 and 236.00 minutes, depending on the model, compared to a fixed 2.90 minutes for Column Filtering. On the larger SCITAB dataset, this pattern holds, with CCoE and Answer times ranging from 462.80 to 1490.40 minutes, while Column Filtering still takes a steady 24.50 minutes. This disparity highlights that the runtime scales significantly with model size—larger models, such as Qwen2.5-72B-Instruct and Llama-3.3-70B-Instruct, require substantially more time than the smaller Qwen2.5-14B-Instruct, due to the intensive reasoning and explanation generation involved in the later stages. However, when normalized by minutes per question (min/Q), the runtime per question remains relatively stable, ranging from 0.39 to 1.24 min/Q across datasets, suggesting that the framework's efficiency scales reasonably with the number of questions. The consistent Column Filtering time across models indicates that this stage's computational cost is minimal and effectively independent of model size, scaling primarily with

the table size (the number of columns).

Table 3.4: End-to-end runtime per dataset. Values are reported in minutes (mins), with minutes per question (min/Q) shown in the right-most column.

Dataset	Model	Column Filtering (mins)	CCoE & Answer Generation (mins)	Total (mins)	min/Q
PubHealthTab	Qwen2.5 (14B-Instruct)	2.90	73.30	76.20	0.39
	Qwen2.5 (72B-Instruct)	2.90	236.00	238.90	1.23
	Llama-3.3 (70B-Instruct)	2.90	216.60	219.50	1.13
SCITAB	Qwen2.5 (14B-Instruct)	24.50	462.80	487.30	0.40
	Qwen2.5 (72B-Instruct)	24.50	1490.40	1514.90	1.24
	Llama-3.3 (70B-Instruct)	24.50	1367.90	1392.40	1.14

3.5 Error Analysis

Table 3.5 summarizes the performance of our UCRET framework and its ablated variants, showcasing the distinct yet complementary contributions of Column Filtering and Counterfactual Chain-of-Explanation generation. Across both PubHealthTab and SCITAB, the full framework consistently outperforms the ablations, underscoring the complementary effect of structured evidence selection and explanatory reasoning.

When we remove column filtering, the impact varies across labels and datasets. On SCITAB, the F1-score for the REFUTES class drops sharply by 26.5 points, falling from 59.84 to 33.33. This steep decline mainly stems from a significant drop in recall, from 46.23 to 20.44, even though precision slightly improves (from 84.82 to 90.32). This suggests that, without column filtering, the model struggles with the additional noise from irrelevant columns, which dilutes the refuting evidence. Interestingly, for the SUPPORTS class on SCITAB, we observe an opposite trend: F1-score increases from 47.34 to 54.46, largely driven by a substantial jump in recall (from 36.24 to 85.96), despite a drop in precision (from 68.25 to 42.03). This pattern implies that column filtering, while generally helpful, might sometimes inadvertently prune supporting evidence, especially in complex claims involving subtle numerical relationships. On PubHealthTab, the impact of removing column filtering is less dramatic but still noticeable: the F1-score for REFUTES declines by 10 points (from 76.74 to 66.67), reaffirming the value of column selection in preserving precision.

Table 3.5: Performance comparison of our claim verification model under ablation settings on the PubHealthTab and SCITAB datasets using LLaMA3.3-70B-Instruct. Results are reported in terms of precision, recall, and F1-score for each veracity label (SUPPORTS, REFUTES, NEI). We evaluate three setups: the full framework (UCRET), the framework without Column Filtering, and the framework without CCoE.

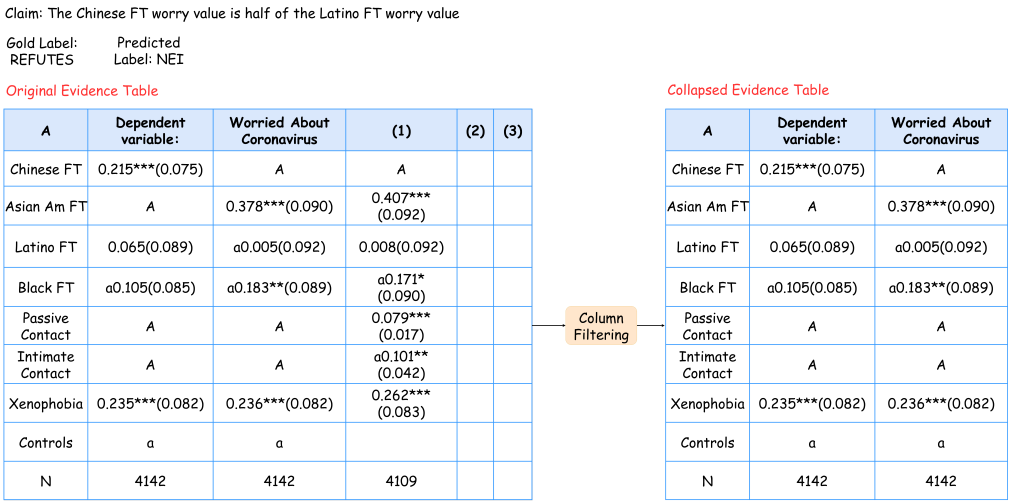
	PubHealthTab			**SCITAB**		
	precision	recall	F1	precision	recall	F1
Full framework						
SUPPORTS	61.97	97.78	75.86	68.25	36.24	47.34
REFUTES	82.50	71.74	76.74	84.82	46.23	59.84
NEI	95.18	76.7	85.15	50.92	90.37	65.14
w/o Column Filtering						
SUPPORTS	71.93	91.11	80.39	42.03	85.96	54.46
REFUTES	60.71	73.91	66.67	90.32	20.44	33.33
NEI	88.89	69.90	78.26	69.23	61.05	64.88
w/o CCoE						
SUPPORTS	64.62	93.33	76.36	61.68	28.93	39.39
REFUTES	70.21	71.74	70.97	58.81	62.53	60.61
NEI	91.46	72.82	81.08	52.90	71.77	60.91

Removing CCoE also degrades performance, though its severity depends on the dataset. On PubHealthTab, the decline is moderate REFUTES drops from 76.74 to 70.97, indicating that for more straightforward factual claims, explanatory reasoning is less critical. However, on SCITAB, where the claims tend to be more complex and ambiguous, the impact is much more pronounced. For instance, the F1-score for SUPPORTS falls from 47.34 to 39.39. This highlights CCoE's crucial role in handling scientific claims that require connecting incomplete or indirect pieces of evidence through hypothetical reasoning. Notably, the NEI class consistently benefits from the complete UCRET pipeline in both datasets. On PubHealthTab, the full framework achieves an F1-score of 85.15, with particularly strong precision (95.18) and solid recall (76.70), which reflects its ability to reliably abstain from incorrect judgments when evidence is lacking. Similarly, on SCITAB, the full model outperforms the ablated versions, achieving an F1-score of 65.14 compared to 64.88 without column filtering and 60.91 without CCoE. These results emphasize how the combination of structured column filtering and explanatory reasoning improves the model's ability to recognize situations with insufficient evidence. Overall, these findings illustrate the

complementary strengths of column filtering and CCoE. Column filtering plays a crucial role in reducing noise, while CCoE enriches the model's reasoning capability, especially in cases that demand nuanced numerical or compositional understanding. When combined, they enable our framework to perform robust and interpretable claim verification across datasets of varying complexity.

Although the results provide a clear picture of our framework's overall effectiveness, they do not fully explain why certain claims remain particularly challenging. To address this gap, we take a closer look at four representative errors that reveal the underlying causes of misclassification. These cases help illustrate how complex interactions between claim characteristics and table structures, including noisy numerical data, cells with noisy or mixed data, and incomplete tables, lead to incorrect predictions. Through this deeper examination, we identify a comprehensive understanding of the key limitations in our framework that lead to these errors. We divided the error cases into four categories as follows.

- **Case 1**: *Missing Critical Numeric Value.* As illustrated in Figure 3.6, the claim "The Chinese FT worry value is half of the Latino FT worry value" requires comparing numerical values for Chinese FT (0.215) and Latino FT (0.065). However, during the column filtering stage, the framework mistakenly selects a "Worried About Coronavirus" column that includes only a placeholder ("A") for Chinese FT, while excluding the original column that contains the actual value. This oversight likely stems from the table's complex regression-style structure, where ambiguous or overlapping column headers obscure the true source of numerical evidence. Without access to the correct value, the reasoning process breaks down, preventing the model from constructing a valid explanation. As a result, the framework erroneously predicts `NOT ENOUGH INFORMATION` instead of the correct label, `REFUTES`.



A	Dependent variable:	Worried About Coronavirus	(1)	(2)	(3)
Chinese FT	0.215***(0.075)	A	A		
Asian Am FT	A	0.378***(0.090)	0.407*** (0.092)		
Latino FT	0.065(0.089)	a0.005(0.092)	0.008(0.092)		
Black FT	a0.105(0.085)	a0.183**(0.089)	a0.171* (0.090)		
Passive Contact	A	A	0.079*** (0.017)		
Intimate Contact	A	A	a0.101** (0.042)		
Xenophobia	0.235***(0.082)	0.236***(0.082)	0.262*** (0.083)		
Controls	a	a			
N	4142	4142	4109		

A	Dependent variable:	Worried About Coronavirus
Chinese FT	0.215***(0.075)	A
Asian Am FT	A	0.378***(0.090)
Latino FT	0.065(0.089)	a0.005(0.092)
Black FT	a0.105(0.085)	a0.183**(0.089)
Passive Contact	A	A
Intimate Contact	A	A
Xenophobia	0.235***(0.082)	0.236***(0.082)
Controls	a	a
N	4142	4142

Figure 3.6: Missing Critical Numeric Value.

- **Case 2**: *Confusion from Noisy Cell Content.* As shown in Figure 3.7, the claim "The 1997 study had the largest sample size" requires comparing participant counts across studies, with Black et al. (1997) reporting 177 participants (59 cases and 118 controls). The k-means clustering step correctly selects the "Sample" column, but the entry for Makela et al. (2002) contains multiple values within a single cell: "535,544 children" (total population) and "161 children hospitalized for meningitis" (a clinical subgroup). Due to the ambiguous formatting and absence of structural delimiters, the model fails to isolate 535,544 as the relevant value during counterfactual reasoning. This confusion prevents the framework from recognizing that Makela et al.'s study involved a substantially larger population, ultimately leading to an incorrect prediction of `NOT ENOUGH INFORMATION` instead of the correct label `REFUTES`.
- **Case 3**: *Misalignment in Categorical Reasoning.* As presented in Figure 3.8, the claim "Foreign Policy is a domestic issue" requires distinguishing between categorical domains. The column filtering step correctly retains the "International Issues" and "Domestic Issues" columns, which are essential for assessing the claim. However, the table lacks vertical headers or clear row-level alignment, providing only horizontal labels without explicit relational cues. As a result, items such as "Foreign Policy" and "Gun Control" appear in the same row but under different columns, which misguides the reasoning process. The model interprets their co-occurrence as indicative of a shared category, conflating international and domestic issues. The lack of structural

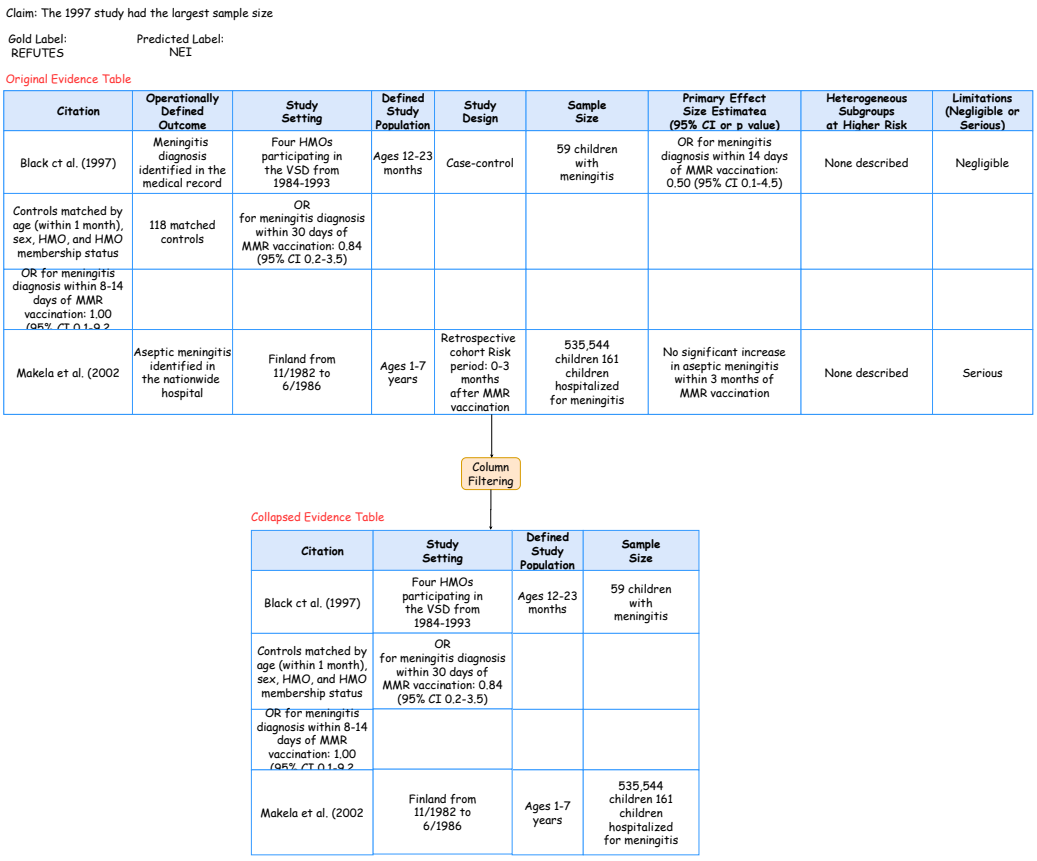
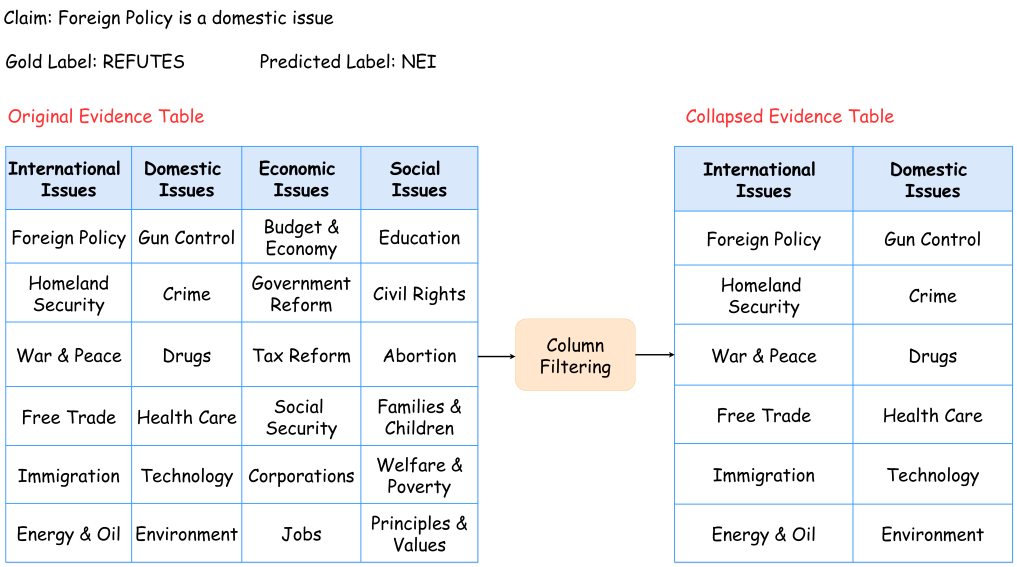


Figure 3.7: Confusion from Noisy Cell Content.

cues prevents the model from recognizing that Foreign Policy belongs under international concerns, resulting in a misclassification. This case shows how missing row organization or vertical guidance in the table can confuse the model when reasoning about categorical distinctions.



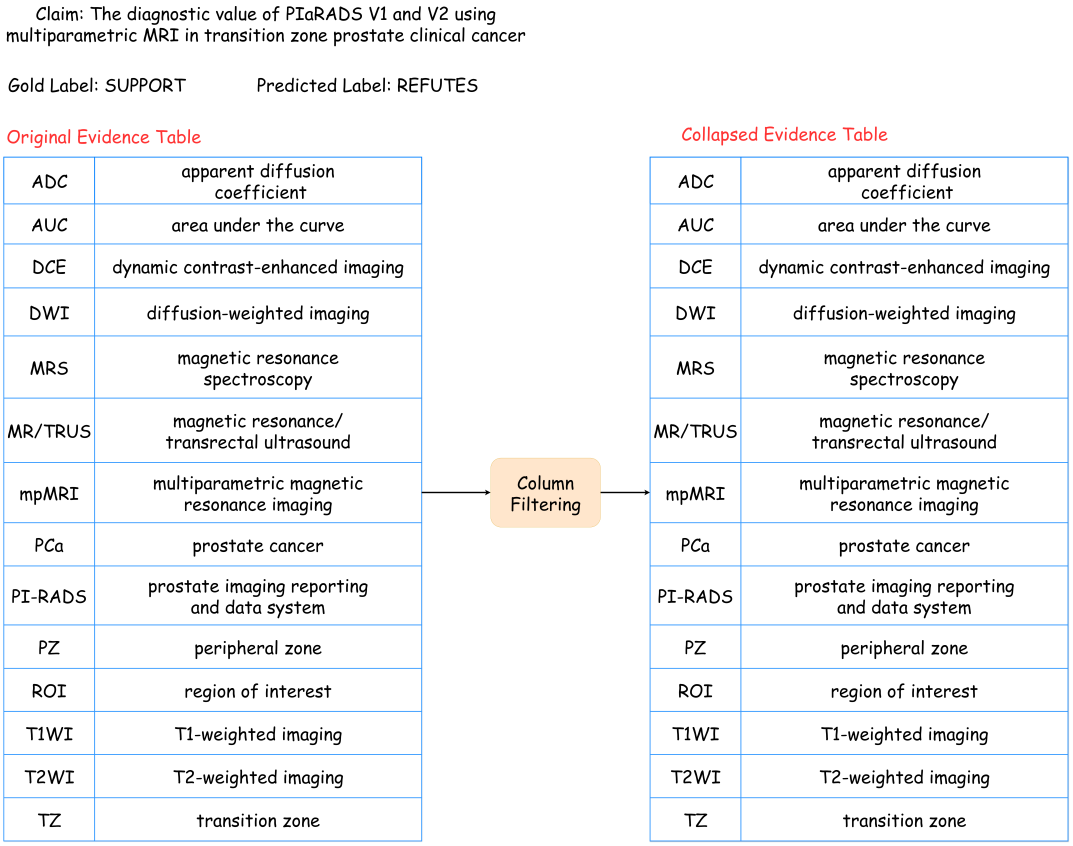
International Issues	Domestic Issues	Economic Issues	Social Issues
Foreign Policy	Gun Control	Budget & Economy	Education
Homeland Security	Crime	Government Reform	Civil Rights
War & Peace	Drugs	Tax Reform	Abortion
Free Trade	Health Care	Social Security	Families & Children
Immigration	Technology	Corporations	Welfare & Poverty
Energy & Oil	Environment	Jobs	Principles & Values

International Issues	Domestic Issues
Foreign Policy	Gun Control
Homeland Security	Crime
War & Peace	Drugs
Free Trade	Health Care
Immigration	Technology
Energy & Oil	Environment

Figure 3.8: Misalignment in Categorical Reasoning.

- **Case 4**: *Lack of Structure in Term-Definition Table.* As illustrated in Figure 3.9, the claim "The diagnostic value of PI-RADS V1 and V2 using multiparametric MRI in transition zone prostate clinical cancer" centers on assessing the relationship between diagnostic imaging protocols and prostate cancer localization. Although the column filtering step retains numerous relevant terms, such as "PI-RADS", "mpMRI", and "TZ", the structure of the table lacks horizontal headers that explicitly define relational semantics or align these concepts across rows. This absence of horizontal structure leads to an unstructured list of abbreviations and definitions, which prevents the model from establishing diagnostic relationships or comparative value between PI-RADS versions. As a result, the counterfactual reasoning process fails to construct a meaningful chain of explanation, leading to a misprediction. This case highlights a structural limitation of the column filtering process when applied to glossary-style tables without clearly defined headers or relational alignment.
  This case illustrates a **bridging external knowledge**: tables alone may lack sufficient information to fully answer certain questions. By retrieving relevant external knowledge from a reliable source, we can aid

language models in grasping the broader context of the query, enabling more informed and nuanced responses.



ADC	apparent diffusion coefficient
AUC	area under the curve
DCE	dynamic contrast-enhanced imaging
DWI	diffusion-weighted imaging
MRS	magnetic resonance spectroscopy
MR/TRUS	magnetic resonance/ transrectal ultrasound
mpMRI	multiparametric magnetic resonance imaging
PCa	prostate cancer
PI-RADS	prostate imaging reporting and data system
PZ	peripheral zone
ROI	region of interest
T1WI	T1-weighted imaging
T2WI	T2-weighted imaging
TZ	transition zone

ADC	apparent diffusion coefficient
AUC	area under the curve
DCE	dynamic contrast-enhanced imaging
DWI	diffusion-weighted imaging
MRS	magnetic resonance spectroscopy
MR/TRUS	magnetic resonance/ transrectal ultrasound
mpMRI	multiparametric magnetic resonance imaging
PCa	prostate cancer
PI-RADS	prostate imaging reporting and data system
PZ	peripheral zone
ROI	region of interest
T1WI	T1-weighted imaging
T2WI	T2-weighted imaging
TZ	transition zone

Figure 3.9: Lack of Structure in Term-Definition Table.

In general, these four cases reveal that, although our column filtering step often retains semantically relevant content, the counterfactual reasoning stage still faces several critical challenges. First, key numeric values required for accurate reasoning may be omitted when they are stored in auxiliary or non-retained columns, limiting the model's ability to perform quantitative comparisons. Second, cells containing noisy or mixed content can disrupt the reasoning process, as the model lacks fine-grained parsing mechanisms to isolate the relevant information. Third, category-based claims become difficult to verify when tables rely solely on horizontal headers and lack structural cues such as row alignment or vertical labels, which can lead to incorrect associations between unrelated items. Finally, glossary-style tables without any horizontal headers, as seen in Figure 3.9, introduce an additional structural limitation: although column filtering retains relevant entities,

the absence of relational structure prevents the model from constructing meaningful logical chains.

To address these issues, we plan to improve the robustness of our framework along multiple dimensions. First, we will introduce additional data quality checks after column filtering to detect and retain critical numeric information that may otherwise be missed. Second, we aim to improve the parsing of complex cells containing noisy or mixed content by incorporating more sophisticated content-aware splitting and normalization strategies. Third, to better capture the structural layout of diverse table formats, including glossary-style and row-aligned tables, we propose extending our use of k-means clustering to both column and row dimensions. This bidirectional clustering will allow the framework to identify not only the most relevant columns but also the most informative rows, thereby constructing a more contextually grounded and structured collapsed table. Together, these improvements are expected to provide the counterfactual reasoning component with cleaner, better-aligned evidence, mitigating noise-induced errors and increasing both the confidence and correctness of the model's final predictions.

3.6 Conclusion

We introduce **UCRET**, a zero-shot framework for interpretable table-based fact verification, requiring neither fine-tuning nor reliance on closed-source models. UCRET addresses key limitations of current large language models—particularly their difficulty in interpreting tabular structure and performing precise numerical reasoning—by explicitly decomposing the verification process into interpretable reasoning stages. The framework integrates two core components: (i) *Column Filtering*, which uses k-means clustering to construct a condensed, claim-relevant evidence table, and (ii) *Counterfactual Chain-of-Explanation*, which prompts an LLM to generate structured explanations under each possible veracity assumption. This design promotes both reasoning accuracy and transparency by disentangling evidence selection from multi-perspective inference. This separation of evidence selection from multi-hypothesis reasoning improves both accuracy and verifiability, especially when tables are noisy or heterogeneous.

In our experiments on PubHealthTab and SCITAB, UCRET achieves competitive results with open-source backbones and surpasses several strong baselines. Ablation studies indicate that Column Filtering and Counterfactual Chain-of-Explanation contribute complementary gains in both accuracy and interpretability. While GPT-4 remains a strong upper bound, our

findings suggest that careful decomposition and structure-aware reasoning can narrow the gap even with lighter models. Beyond table structure, a practical challenge is that **tables alone may not always contain sufficient information to fully verify or answer certain queries**, especially in domain settings where key context is implicit or missing. This motivates a **bridging external knowledge** direction: retrieving relevant external evidence from reliable sources can help language models grasp the broader context of a claim, leading to more informed and nuanced decisions when the table-only evidence is insufficient.

Future work will incorporate bidirectional clustering to jointly identify the most relevant columns and informative rows. In addition, integrating content-aware parsing and normalization strategies could further improve robustness to noisy or heterogeneous table formats. These enhancements aim to extend UCRET's applicability to a wider range of real-world tabular reasoning tasks across scientific, medical, and other high-stakes domains. Finally, we will explore knowledge-augmented extensions that retrieve external evidence when the collapsed table is not evidential enough, while still maintaining explicit, inspectable links between the final decision and its supporting sources.

Chapter Summary

This chapter introduced UCRET, an unsupervised framework for table-based fact verification that aims to make decisions both accurate and easy to verify. The pipeline starts with Column Filtering, where we use spherical k-means to select columns that are most relevant to the claim and to form a collapsed evidence table. By removing irrelevant columns, UCRET reduces noise and makes the evidence easier for a language model to read. On top of the collapsed table, UCRET performs label-conditioned counterfactual reasoning for the three veracity labels (SUPPORTS, REFUTES, and NOT ENOUGH INFO) and produces concise explanations with cell-level citations. This design encourages the model to justify each possible label using the same evidence, instead of giving one opaque explanation. Overall, UCRET shows that a fully unsupervised pipeline can support table verification while remaining transparent and inspectable.

Nevertheless, this table-only setting reveals an important boundary: **tables alone may not always provide enough information to fully verify a claim**. In such cases, a natural extension is **bridging external knowledge—triggering** retrieval from reliable sources to supply missing context, while keeping the final decision auditable through explicit evidence

links. More broadly, after completing the table-only setting (Scenario I), we move to **Scenario II**, where we study **unsupervised external retrieval** and **knowledge-graph reasoning** to bring in missing evidence with inspectable, evidence-linked traces.

In the next chapter, we extend this idea to table question answering, where the system must choose among multiple options, and the evidence differences can be small but important. In this setting, using cosine similarity inside spherical k-means can be less reliable, because it compares single embedding directions and may fail when the relevant signal is spread across multiple tokens or cells. Motivated by the distribution-aware retrieval results of USCRaKe, we introduce UCRET-JS, which replaces cosine-based matching with a Jensen–Shannon divergence (JSD) based measure to make column selection more robust for option-driven table QA. This extension keeps UCRET's unsupervised and explainable structure while adapting the similarity measure to better fit the tighter evidence selection required by TabMCQ-style questions.

Chapter 4

UCRET-JS: Unsupervised Column Relevance Extraction for Table-Based Question Answering via Structured and Explainable Large Language Model Reasoning

4.1 Introduction

Table-based question answering requires identifying which columns in a table are most relevant to a natural-language question and reasoning over them to select the correct answer from candidate options. Compared with table-based fact verification, this setting is often more fine-grained: multiple options can be semantically close, so small differences in the selected evidence may change the final choice.

In Chapter 3, we introduced an unsupervised column filtering method combining (i) a lexical pre-filter and (ii) spherical k-means clustering over dense embeddings using cosine similarity. Separately, our unsupervised chunk retrieval framework in Chapter 7 showed that *Jensen–Shannon divergence* (JSD) provides a stable, bounded, and symmetric way to compare representations in a distribution-aware manner.

In this chapter, we unify these ideas: we retain the lexical pre-filter, but replace cosine similarity in the semantic clustering step with a JSD-based criterion. To apply JSD to dense vectors, we map question and column embeddings onto the probability simplex using temperature-scaled softmax. The resulting selector remains *fully unsupervised* and integrates lexical and information-theoretic signals. We refer to this method as **UCRET-JS**.

4.2 Related Work

Recent research on table-based reasoning can be grouped into two main directions. The first direction improves reasoning precision by combining LLMs with explicit operations. Lu et al. [42] propose a tool-augmented framework that formats tables, builds task-specific tools for tabular operations, and generates explanations that integrate tool outputs, aiming to address weaknesses of pure chain-of-thought on tables. This line is effective for tasks that require reliable table manipulation and numerical computation, but it typically introduces additional tool design and training requirements.

The second direction improves table reasoning by training tabular LLMs with instruction tuning data, including large-scale curated or synthesized data. Zheng et al. [133] summarize this trend and propose a data synthesis pipeline tailored for table instruction tuning, emphasizing diversity and efficiency of synthetic instructions and tables. While training-centric approaches can deliver strong performance, they move the burden to data construction and fine-tuning, which can be costly to maintain when domains, table formats, or task requirements change.

As discussed in Chapter 3, UCRET focuses on a different bottleneck: unsupervised evidence selection inside the table. UCRET shows that even without task-specific fine-tuning, a pipeline can remain competitive by (i) filtering columns with a lexical step and spherical k-means over dense embeddings, and (ii) producing label-conditioned explanations grounded in a collapsed evidence table. This design targets transparency and portability, especially when labeled supervision is limited and when the evidence must be inspectable at the cell level.

Building on this motivation, UCRET-JS retains the same fully unsupervised pipeline—lexical pre-filtering followed by semantic column selection—but revises the similarity signal used in the semantic step. Specifically, we replace the cosine-based assignment in spherical k-means with a Jensen–Shannon divergence criterion by first mapping question and column embeddings onto the probability simplex (via temperature-scaled softmax). This design follows the distribution-aware perspective established in our USCRaKe chapter: unlike cosine similarity, which compares vector directions, JSD compares normalized distributions and is bounded and symmetric, which makes it a stable choice for unsupervised matching. This adjustment is particularly relevant for table question answering, where multiple answer options can satisfy similar semantic constraints and accurate answering depends on selecting columns that provide the most discriminative evidence. Importantly, UCRET-JS keeps the same explainable structure of UCRET;

it only changes the similarity measure used for column selection so that downstream reasoning is grounded in a more reliable collapsed evidence table.

4.3 Framework

Given a question Q, answer options $\mathcal{O} = \{o_1, \ldots, o_M\}$ (typically M=4), and a source table $\mathcal{D}$ with columns $\mathcal{J} = \{C_1, \ldots, C_{|\mathcal{D}|}\}$, our goal is to select a compact subset $S \subseteq \mathcal{J}$ of columns most relevant to Q and produce a collapsed table $\mathcal{D}_{\text{collapsed}}$ that preserves evidence necessary for answering Q. Formally, we denote our framework as a function $f_\phi(\cdot)$ with parameters ϕ as follows:

$$\hat{y} = f_\phi(Q, \mathcal{D}) \tag{4.1}$$

where $\mathcal{D}$ consists of a descriptive caption S and its content $\{D_{i,j} \mid 1 \leq i \leq R_\mathcal{D},\ 1 \leq j \leq C_\mathcal{D}\}$, where $R_\mathcal{D}$ and $C_\mathcal{D}$ correspond to the total number of rows and columns, respectively. Each cell (i, j) contains a data value $D_{i,j}$. Figure 4.1 illustrates the input and output structure of our framework.

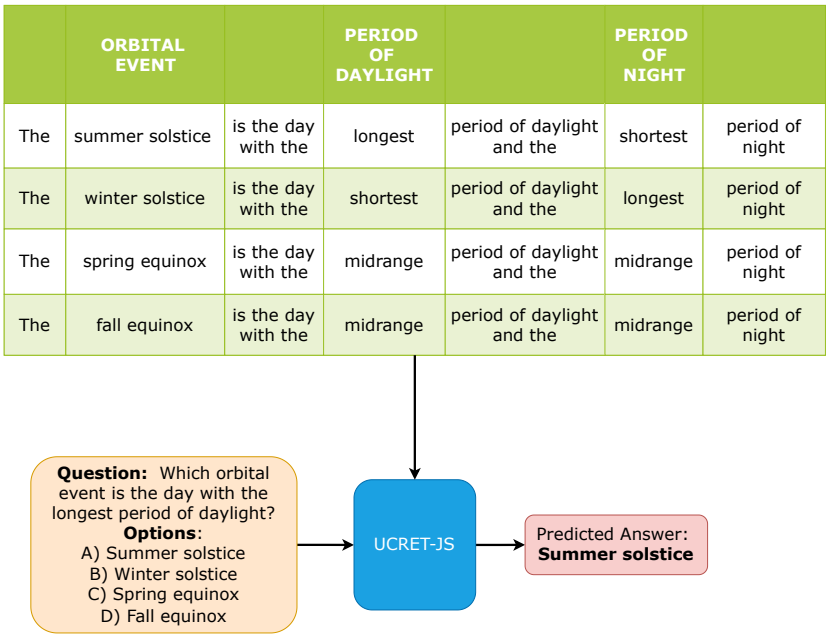


Figure 4.1: An example of the input and output for UCRET-JS.

As shown in Figure 4.2, UCRET-JS factorizes table-based queston answering into three tightly coupled stages: *Column Filtering*, *Counterfactual*

Chain-of-Explanation, and *Answer Generation*. This separation isolates evidence selection, explanation generation, and final decision-making, yielding interpretable and generalizable reasoning over tabular data. Unlike approaches that depend on closed-source models or task-specific fine-tuning, UCRET-JS operates fully in a zero-shot setting and exhibits robustness across both medical and scientific domains.

The three-stage design is essential for effective table reasoning. *Column Filtering* concentrates the model's attention on semantically relevant evidence, reducing noise. *Counterfactual Chain-of-Explanation* prompts the model to systematically explore all veracity possibilities, encouraging structured, faithful reasoning instead of shallow or biased inference. Finally, the *Answer Generation* stage performs a reflective verification, comparing the candidate explanations and selecting the option $\hat{y}$ most consistent with the evidence. This modular decomposition directly addresses the core challenges of table-based verification—evidence selection, rationale construction, and prediction validation—leading to more accurate, interpretable, and robust fact verification across domains.

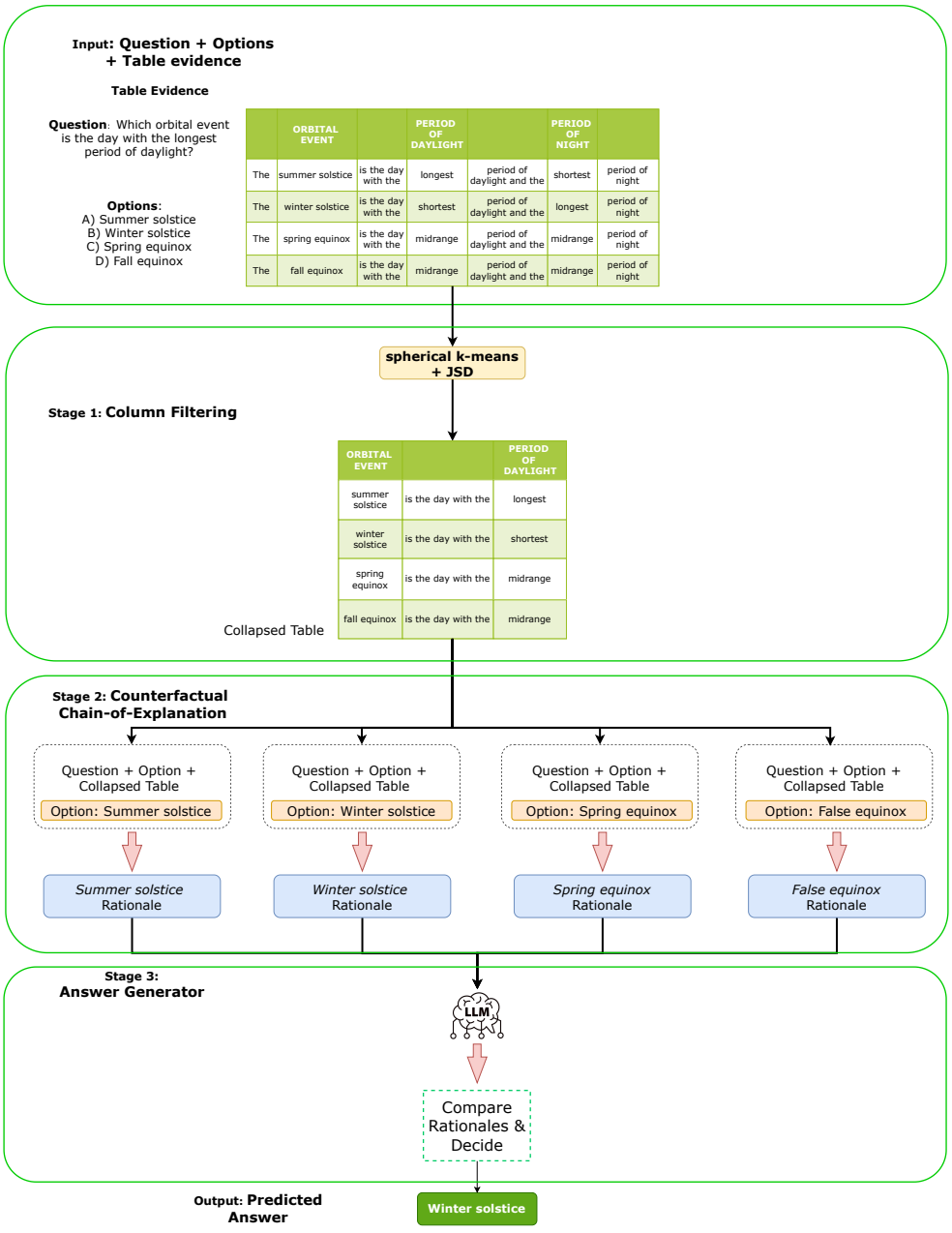


Figure 4.2: Overview of UCRET-JS. The framework has three modules: (i) **Column Filtering**, which selects query-relevant columns and forms a collapsed evidence table; (ii) **Counterfactual Chain-of-Explanation**, which generates option-specific rationales and assesses their alignment with the question; and (iii) **Answer Generation**, which verifies the competing explanations and outputs the final decision.

4.3.1 STAGE 1: Integrating JSD into the Column Selector

We retain the UCRET structure: (1) a **lexical pre-filter** selects columns with Jaccard overlap above a threshold τ; (2) a **semantic selection** step examines the remaining columns. The only change is the similarity used in step (2):

- *In chapter 3 UCRET:* rank or cluster columns by cosine similarity between L_2-normalized embeddings.
- **In UCRET-JS:** we measure the semantic similarity between the question Q and a candidate column C_j using the Jensen–Shannon divergence. Let $\boldsymbol{e}_Q \in \mathbb{R}^d$ denote the embedding of the question Q, and $\boldsymbol{e}_{C_j} \in \mathbb{R}^d$ denote the embedding of column C_j. To compute JSD, we first convert embeddings into probability distributions by applying a softmax over dimensions:

$$P_Q(l) = \frac{\exp(e_{Q,l})}{\sum_{t=1}^{d} \exp(e_{Q,t})}, \quad P_{C_j}(l) = \frac{\exp\big(e_{C_j,l}\big)}{\sum_{t=1}^{d} \exp\big(e_{C_j,t}\big)}, \quad l = 1, \ldots, d, \tag{4.2}$$

  where $e_{Q,l}$ and $e_{C_j,l}$ are the l-th components of $\boldsymbol{e}_Q$ and $\boldsymbol{e}_{C_j}$, respectively. This ensures $P_Q(l) \geq 0$, $P_{C_j}(l) \geq 0$, and $\sum_{l=1}^{d} P_Q(l) = \sum_{l=1}^{d} P_{C_j}(l) = 1$. We then define the mixture distribution at dimension l as:

$$M(l) = \frac{1}{2}\left(P_Q(l) + P_{C_j}(l)\right). \tag{4.3}$$

  Finally, the JSD-based score between Q and C_j is computed as:

$$\mathrm{JSD}(Q, C_j) = \frac{1}{2}\sum_{l=1}^{d} P_Q(l)\,\log\left(\frac{P_Q(l)}{M(l)}\right) + \frac{1}{2}\sum_{l=1}^{d} P_{C_j}(l)\,\log\left(\frac{P_{C_j}(l)}{M(l)}\right). \tag{4.4}$$

4.3.2 STAGE 2: Counterfactual Chain-of-Explanation

In this stage, we adopt a form of counterfactual reasoning tailored for table-based question answering. Given a question Q, answer options $\mathcal{O} = \{o_1, \ldots, o_M\}$, and a collapsed evidence table $\mathcal{D}_{\text{collapsed}}$, the model generates a counterfactual chain of explanation to evaluate how the evidence may support or fail to support each option as the correct answer. Specifically, for each option o_i, the language model is asked to generate an explanation conditioned on that assumption:

- ex_{o_1}: assuming the table supports o_1 as the correct answer,
- ex_{o_2}: assuming the table supports o_2 as the correct answer,
- ...
- ex_{o_M}: assuming the table supports o_M as the correct answer.

To be clear, each explanation is generated by a parameterized probabilistic model p_θ through a maximized likelihood process, step-by-step identifying relevant facts, data points, or rows in the table that directly lead to selecting the option as the answer. It clearly explains how each cited cell filters, matches, compares, or aggregates to produce this selection, ensuring the explanation is logical, precise, and grounded in specific table content.

$$p_\theta\big(ex_{o_i} \mid y = o_i,\, Q,\, \mathcal{D}_{\text{collapsed}}\big) \;=\; \prod_{k=1}^{L} p_\theta\big(t_k^{(o_i)} \mid t_{<k}^{(o_i)},\, y = o_i,\, Q,\, \mathcal{D}_{\text{collapsed}}\big) \tag{4.5}$$

where $t_k^{(o_i)}$ is the k-th token of explanation ex_{o_i}, and L is its length.

To systematically generate these explanations, we design **Chain-of-Explanation** prompts that instruct the model to reason step-by-step using only the provided table. An example Chain-of-Explanation prompt for an option o_i is as follows:

```
You are an expert for table-based question answering.
Your task is to explain why the given option is the correct answer
    using only the table.

Question: {question}
Option: {option}
Table Representation:
{table_representation}

Step-by-step, identify the relevant facts, data points, or rows in
    the table that directly lead to selecting the option as the
    answer to the question. Clearly explain how each cited cell
    filters, matches, compares, or aggregates to produce this
    selection. Your explanation should be logical, precise, and
    grounded in specific table content.
```

4.3.3 STAGE 3: Answer Generator

Given a question Q, a collapsed evidence table $\mathcal{D}_{\text{collapsed}}$, answer options $\mathcal{O} = \{o_1, \ldots, o_M\}$, and the set of explanations $\mathcal{X} = \{ex_{o_1}, \ldots, ex_{o_M}\}$, the

Answer Generator selects the final answer as:

$$\hat{y} = f_{\text{AG}}(Q, \mathcal{D}_{\text{collapsed}}, \mathcal{O}, \mathcal{X}), \tag{4.6}$$

where $\hat{y} \in \mathcal{O}$ is the predicted correct option.

Our prompting strategy empowers LLMs to deliberate on their responses, allowing them to identify and correct errors through self-reflection. This stage is inspired by Chain of Verification (CoVe) [125]. Similar to UCRET in 3, we extend it with a variant called Counterfactual Chain of Verification (CCoV) prompting, where counterfactual explanations $\mathcal{X} = \{ex_{o_1}, \ldots, ex_{o_M}\}$ (one for each answer option) generated in Stage 2 serve as baseline rationales for verification. The LLM is prompted to assess the strengths and weaknesses of each explanation in relation to the question Q and the collapsed evidence table $\mathcal{D}_{\text{collapsed}}$, then select the most rational answer choice. By integrating structured tabular evidence with natural language reasoning, this approach boosts decision accuracy and enhances model interpretability in table-based question answering. Notably, CCoV allows LLMs to introspectively evaluate their own counterfactual explanations, refining outputs via iterative verification and correction, which strengthens reasoning while promoting transparency in the decision process.

To systematically generate the final answer, we design our prompt to introspect upon the counterfactual explanations generated with Q, $\mathcal{D}_{\text{collapsed}}$, $\mathcal{O} = \{o_1, \ldots, o_M\}$, and $\mathcal{X} = \{ex_{o_1}, \ldots, ex_{o_M}\}$ as follows:

```
You are an expert answer selector for question answering.

You will be given:

Input
Question: {question}
Table Evidence: {collapsed_evidence_table}
Options:
A) {o_1}
B) {o_2}
C) {o_3}
D) {o_4}
Explanations:
A: {ex_{o_1}}
B: {ex_{o_2}}
C: {ex_{o_3}}
D: {ex_{o_4}}

Your task is to determine which single option is best supported by
    the table, using only the table and the option explanations.
```

```
Task:
    Step-by-step, identify the specific cells in the table that each
     option's explanation relies on. Choose the option whose
    explanation is fully consistent with the table, supported by the
     smallest, clearest set of cells, and makes no unsupported
    assumptions.

Output:
    Answer with only one label: A, B, C, or D.
```

4.4 Experiment Results

4.4.1 Empirical Preparation

We fix the number of clusters for the column-filtering stage to k=2. Replacing cosine with JSD, we set the acceptance gate to a JSD *distance* threshold η_{JS}=0.30. We retain the Jaccard threshold grid τ=0.20. All experiments are executed on a single NVIDIA A6000 GPU (48 GB). For semantic representations of the question and column headers, we use the BGE encoder [126], yielding 1024-dimensional embeddings.

For counterfactual explanation and answer generation, we adopt fully open-source instruction-tuned LLMs—Llama-3.3-70B-Instruct [127], Qwen2.5-14B-Instruct, and Qwen2.5-72B-Instruct [128]—to align with our zero-shot, open-model design while probing robustness across 14B–72B scales. Instruction tuning improves adherence to our structured prompts; all models are run with 4-bit quantization [129] and `do_sample` set to `False` to avoid stochastic deviations. Unless otherwise noted, we use the same experimental configuration as in the previous section.

4.4.2 Experiment Results

We select the existing tabular benchmark to evaluate the performance of our proposed framework UCRET-JS: TabMCQ [134], which contains 9092 manually annotated multiple choice questions (MCQs) with their answers and 63 tables as its knowledge. The metric is used as accuracy (%). For a baseline comparison, we evaluate our method against recent state-of-the-art approaches, TabFlash [135], TableDreamer [133] and SynTab-LLaVA [46]. Zheng et al. [133] introduces a progressive and weakness-guided data synthesis framework tailored for table instruction tuning, named TableDreamer,

to explore the input space under the guidance of newly identified weakness data to more effectively enhance the model performance. In addition, Kim et al. [135] propose TabFlash, a multimodal large language model for table understanding that integrates a pruning strategy with token focusing. This combination helps reduce redundant information while minimizing the loss of critical content during pruning. Moreover, we include SynTab-LLaVA [46], a hybrid multimodal model that operates at multiple resolutions, specifically designed to enhance the understanding of both the textual content and the structural layout of tables in images.

Table 4.1: Accuracy comparison across baselines and UCRET-JS variants using different LLM backbones on the TabMCQ dataset. Bold indicates the best result; underline indicates the second best.

Approach	Method	Acc. (%)
Baselines	TableDreamer (Llama3.1-70B-Instruct) [42]	82.99
	TableDreamer (GPT-4o) [42]	**84.29**
	TabFlash (InternVL-2.5-1B) [43]	57.80
	TabFlash (InternVL-2.5-3B) [44]	71.90
	SynTab-LLaVA (Vicuna-1.5-7B) [45]	70.55
Ours	*Qwen2.5-14B-Instruct*	
	UCRET-JS w/o Column Filtering	53.28
	UCRET-JS w/o CCoE	55.31
	UCRET-JS	60.46
	Qwen2.5-72B-Instruct	
	UCRET-JS w/o Column Filtering	70.29
	UCRET-JS w/o CCoE	72.80
	UCRET-JS	77.96
	Llama-3.3-70B-Instruct	
	UCRET-JS w/o Column Filtering	74.44
	UCRET-JS w/o CCoE	77.51
	UCRET-JS	<u>83.01</u>

Table 4.1 compares UCRET-JS with recent strong baselines on TabMCQ and reports consistent evidence that our unsupervised design contributes substantially to accuracy across open LLM backbones. First, UCRET-JS shows a clear scaling trend: performance increases from Qwen2.5-14B to Qwen2.5-72B and further to Llama-3.3-70B, indicating that the framework effectively leverages stronger reasoning capacity without any task-specific fine-tuning. More importantly, the ablations validate the necessity of both

core components. Removing Column Filtering causes the largest drop across all backbones, suggesting that early-stage evidence compression—via relevance-aware column selection—is critical for reducing noise and making the downstream reasoning tractable. Removing label-conditioned counterfactual reasoning also consistently degrades performance, confirming that generating structured rationales conditioned on each candidate label improves discrimination under ambiguity, especially when tables contain distracting but superficially related fields. With the strongest open backbone (Llama-3.3-70B), UCRET-JS reaches near-best performance and becomes competitive with closed-model systems, while maintaining the key practical advantages of being zero-shot, explainable, and based on open models. Overall, these results highlight the novelty of combining JSD-based unsupervised clustering for column relevance with counterfactual, label-aware explanation generation: the two modules are complementary, and together they enable robust table reasoning under a fully unsupervised setting.

4.4.3 Ablation Study

We evaluated our UCRET-JS method on the first 150 items of the TabMCQ dataset, a benchmark designed for table-based multiple-choice question answering. As shown in Table 4.2, our full UCRET-JS approach achieved 70.67% accuracy, outperforming two ablated versions: one without column filtering (62.33%) and another without counterfactual chain-of-explanation reasoning (66.00%). This shows that both column filtering and CCoE contribute meaningfully—removing either drops performance by 8.34 and 4.67 percentage points, respectively. Compared with the cosine-based UCRET (69.33%), **UCRET-JS** reaches 70.67%, a **+1.34** point absolute gain, indicating that replacing cosine with JSD yields a modest but consistent improvement in column selection and downstream accuracy. Column filtering helps focus the model on relevant parts of the table, reducing noise, while CCoE enables clearer, step-by-step reasoning that improves answer selection.

Table 4.2: Ablation study results on the first 150 items of TabMCQ. Bold indicates the best result; underline indicates the second best

Method	Acc. (%)
w/o Column Filtering	62.33
w/o CCoE	66.00
UCRET	<u>69.33</u>
UCRET-JS	**70.67**

4.4.4 Error Analysis

Although our approach clearly outperforms prior methods, it still has limitations. To better understand the remaining errors, we randomly inspected a failed test instance where the model does not select the correct answer. This case also exposes a labeling issue in the dataset: a small number of questions effectively allow more than one correct option, yet the benchmark provides only a single gold label. For the question "Which of the following is a carnivore?", both Andean Cat and Arctic fox appear among the candidates and are both valid carnivores, but only one is marked as correct.

Crucially, from Figure 4.3, the evidence table itself indicates that both Andean Cat and Arctic Fox are carnivores. This remains true even after Column Filtering: the collapsed table still contains the diet column and preserves the rows for both animals, each labeled "carnivore." Therefore, the model's prediction is not contradicted by the evidence; instead, the error arises because the task is evaluated as single-choice even though the table supports multiple candidates. In other words, this is best characterized as an ambiguity or multi-valid-answer case under a single-gold annotation protocol.

From a system perspective, this case highlights a tie-breaking weakness: when multiple options satisfy the same evidence constraint, the final selection can depend on minor scoring differences or prompt sensitivity, despite both being evidence-consistent. From an evaluation perspective, such items can underestimate the faithfulness of evidence-based methods, since a model may provide a verifiable and correct rationale while still being penalized for not matching the single annotated label.

A practical improvement is to make UCRET-JS ambiguity-aware. After generating the collapsed evidence table, the system can explicitly check which options are supported by the table. If more than one option is supported, the system should either (i) flag the question as under-specified given the table (with lower confidence), or (ii) return all evidence-supported options. This change would align predictions more faithfully with the evidence and make error analysis more informative in the presence of noisy or incomplete annotations.

Question: Which animal below is a carnivore?

Options:
A. Andean Cat
B. Arctic fox
C. Elephant
D. Goat

Gold Answer: Arctic fox Predicted Answer: Andean Cat

Original Evidence Table

Type of Animal		Type of Diet (herbivore, omnivore, carnivore)
Buffalo	is a(n)	herbivore
Cattle	is a(n)	herbivore
Zebra	is a(n)	herbivore
Donkey	is a(n)	herbivore
Crickets	is a(n)	omnivore
Opaleye	is a(n)	omnivore
Wasps	is a(n)	omnivore
Squirrel monkey	is a(n)	omnivore
Raccoons	is a(n)	omnivore
Andean Cat	is a(n)	carnivore
Arctic Fox	is a(n)	carnivore
Gray Fox	is a(n)	carnivore
Wolverine	is a(n)	carnivore

Column Filtering

Collapsed Evidence Table

Type of Animal		Type of Diet (herbivore, omnivore, carnivore)
Buffalo	is a(n)	herbivore
Cattle	is a(n)	herbivore
Zebra	is a(n)	herbivore
Donkey	is a(n)	herbivore
Crickets	is a(n)	omnivore
Opaleye	is a(n)	omnivore
Wasps	is a(n)	omnivore
Squirrel monkey	is a(n)	omnivore
Raccoons	is a(n)	omnivore
Andean Cat	is a(n)	carnivore
Arctic Fox	is a(n)	carnivore
Gray Fox	is a(n)	carnivore
Wolverine	is a(n)	carnivore

Figure 4.3: Case-study example of UCRET-JS on TabMCQ.

4.5 Conclusion

In this chapter, we presented UCRET-JS, a fully unsupervised and explainable framework for table-based multiple-choice question answering. The key change from UCRET is in the column selection stage: we keep the lexical prefilter but replace cosine-based matching in the semantic step with a Jensen–Shannon divergence criterion after mapping embeddings onto the probability simplex. This design transfers the distribution-aware similarity principle from our USCRaKe study to the tabular setting, while preserving the same

three-stage pipeline: Column Filtering, option-conditioned Counterfactual Chain-of-Explanation, and Answer Generation.

Experiments on TabMCQ show that UCRET-JS benefits from stronger open LLM backbones and that both major components are necessary. Across Qwen2.5 (14B and 72B) and Llama-3.3-70B, removing Column Filtering yields the largest accuracy drop, confirming that early evidence compression is critical for reducing noise in tables. Removing the option-conditioned explanation stage also consistently degrades performance, indicating that structured, per-option rationales help the final selector discriminate among close choices. On the strongest open backbone (Llama-3.3-70B), UCRET-JS achieves 83.01% accuracy on TabMCQ without any task-specific fine-tuning.

UCRET-JS also clarifies the scope of this table-only setting. Since the model is restricted to the evidence contained in the input table (after column filtering), it cannot introduce additional facts that are not present in the table. This limitation motivates Part II of the dissertation, where we incorporate external evidence through unsupervised retrieval and knowledge-graph reasoning to support knowledge-augmented reasoning with explicit and verifiable traces.

Finally, our error analysis highlights a practical limitation of the benchmark: some questions admit multiple evidence-consistent options, but only a single gold label is provided. In such cases, the model can produce a table-consistent answer and still be counted as incorrect. This suggests a clear next step: incorporate an ambiguity-aware check that detects multiple supported options from the collapsed table and either flags under-specified instances or returns the set of evidence-supported answers.

Chapter Summary

This chapter introduced UCRET-JS, an unsupervised framework for table-based multiple-choice question answering. The method follows the same structure as UCRET: a lexical pre-filter first removes obviously irrelevant columns, and a semantic selector then chooses the most relevant columns to form a collapsed evidence table. The main extension is that we replace cosine similarity in the semantic step with Jensen–Shannon divergence by mapping dense embeddings onto the probability simplex using temperature-scaled softmax. This makes the semantic comparison distribution-aware and consistent with the principle used in USCRaKe.

Given the collapsed table, the framework generates option-conditioned explanations, one for each candidate answer, grounded in cited cells from the collapsed table. A final answer generator compares these explanations

against the evidence table and selects the single best-supported option. Experiments on TabMCQ show that performance scales with stronger open LLM backbones and that both Column Filtering and option-conditioned explanations contribute meaningfully in ablations. A case-study error analysis further shows that some failures are caused by single-label annotations in questions where the table supports multiple candidates. Overall, UCRET-JS extends the unsupervised and explainable design of UCRET to table question answering by strengthening the similarity measure used for column relevance selection.

Finally, this chapter completes the setting of *unsupervised semantic retrieval and verifiable reasoning without external knowledge* in the table domain. In the next chapters, we study unsupervised retrieval and reasoning with external knowledge through extracted knowledge graphs and distribution-aware text retrieval.

Chapter 5

K-Bloom: Unleashing the Power of Pre-trained Language Models in Extracting Knowledge Graph with Pre-defined Relations

5.1 Introduction

Pre-trained language models have revolutionized natural language processing (NLP) by demonstrating an unprecedented ability to process vast amounts of unstructured text data, achieving state-of-the-art results across various tasks. Models like BERT [20], GPT-3 [21], and T5 [22] are trained on extensive corpora, leading to the hypothesis that their parameters encapsulate a significant amount of factual knowledge. Recent studies suggest that PLMs can be conceptualized as knowledge bases [51], which offer a flexible and powerful tool to extract and utilize embedded information. However, this raises a critical question: Which specific facts are stored within these models, and how can this internal knowledge be systematically represented and accessed?

A promising approach involves the use of knowledge graphs, which are structured representations of information that capture relationships between entities within a specific domain, making information accessible to both humans and machines [23]. KGs consist of collections of triples, where each triple represents a relation r between a head entity h and a tail entity t. Examples of real-world KGs include Freebase [136], WordNet [137], and YAGO [138], which are used to support efficient supply chain management [139], improve accurate and explainable recommendation systems [140], enable commonsense reasoning [141], enhance business scenarios [142], and detect financial fraud [143].

Building KGs from raw text data is a challenging task that requires advanced NLP techniques such as Named Entity Recognition [144], Relation

Extraction [145], and Question Answering [26]. The integration of PLMs with KGs, as seen in recent studies [146], has shown potential to improve the quality and diversity of the knowledge generated, thus advancing applications such as search engines [24], recommendation systems [25], and question answering systems [26, 27]. This synergy between PLMs and KGs not only provides a snapshot of the knowledge stored within these models but also offers a pathway for systematically harnessing and expanding that knowledge.

The most relevant area to our approach lies in knowledge graph extraction from language models. BertNet, as introduced by [28], is a framework specifically designed to obtain knowledge bases directly from language models. This methodology focuses on extracting relation triplets from masked sentences to generate knowledge graphs. However, the knowledge graphs produced by BertNet often include inconsistent entity pairs, which decreases the reliability and correctness of the extracted information, leading to lower accuracy. To address these limitations, we propose an unsupervised knowledge extraction pipeline from PLMs, called Knowledge Bloom (**K-Bloom**). K-Bloom aims to explore the facts stored within the internal knowledge bases of PLMs and provide a comprehensible representation that facilitates both manual and automatic evaluation of what PLMs know and what they do not. By leveraging PLMs, K-Bloom extracts unseen tuples without supervision from minimal input, consisting of an initial prompt and a few seed entity pairs. Our approach is designed to generate new entity tuples whose semantic similarity closely aligns with the seed entity pairs, without being constrained by pre-existing datasets.

Despite the growing interest in utilizing PLMs as knowledge bases, to our knowledge, our framework is the first to select the most reasonable tuples for an arbitrary relation within a set of pre-specified relations from any PLMs (with minimal definition of relations as input) based on comprehensive scoring. We convert implicit PLM knowledge into an explicit KG with an OT-based tuple consistency filter. Unlike hard token matching, our OT-based approach enables soft alignment and retains only those relation tuples that are consistent with their contextual usage, thereby improving the quality and usability of the extracted knowledge. Specifically, after harvesting all candidate entity pairs for each relation that consistently satisfy a diverse set of prompts, we evaluate and rank these pairs to identify the most relevant and meaningful tuples. The key component of our scoring function leverages the concept of Word Mover's Distance, grounded in Optimal Transport theory, to enhance the selection of reasonable tuples extracted from PLMs. The core insight of our approach lies in its ability to measure the semantic distance between generated tuples and seed entity pairs, ensuring that the selected tuples maintain high contextual similarity to the original seed entities. This

not only enhances the accuracy of the knowledge graph completion process but also amplifies the diversity and quality of the generated knowledge. The novelty of our scoring function lies in its integration of Optimal Transport principles to effectively align and map the semantic spaces of different entities, thereby producing more meaningful and contextually appropriate knowledge tuples.

5.2 Related Works

5.2.1 Knowledge Probing using Prompt

Since the advent of pre-trained language models, one key question has been how well these models capture the factual information in the texts on which they are trained. Knowledge probing is vital for understanding the knowledge transfer mechanisms within PLMs. Researchers have leveraged PLMs such as BERT [20] and conditional masked language modeling [62] as powerful tools for probing tasks. These probing tasks have provided valuable insights into the extent to which PLMs capture linguistic information, shedding light on their suitability for various NLP applications. LAMA [51] is the first framework that illustrates that PLMs can be viewed as knowledge bases. The LAMA methodology focuses on extracting relation triples from cloze "fill-in-the-blank" statements to generate knowledge graphs from PLMs. In recent work, BertNet [28] proposes an automatic framework to generate weighted prompts using GPT-3 [21] and search for new entity pairs with PLMs.

5.2.2 Knowledge Graph Construction

Knowledge Graph Construction (KGC) tasks mainly employ pipelines that use Attribute Extraction [52], Relation Extraction [53], and Entity Extraction [54] techniques to extract structural information from unstructured texts, capturing entities, their relationships, and associated attributes. KGC is constructed through manual and automated approaches. For example, Freebase [55], WordNet [56], and Wikidata [57] are well-known large KGs developed by human labor. In contrast, OpenIE [58–60] and YAGO [61] are developed to reduce human effort.

The work most similar to ours, in addition to BertNet, is "Prompting as Probing" [147], where the authors use a prompt engineering approach to obtain knowledge from GPT-3 [21]. However, while the authors of "Prompting as Probing" use fact probing to ask the language model to predict possible objects of a triple, given the subject and relation, our framework

leverages masked LMs to predict all potential entity pairs from a given prompt.

5.2.3 Language Models as Knowledge Graphs

Recent research has focused on probing knowledge in pre-trained language models, a process commonly referred to as factual probing. This approach aims to measure how much factual knowledge is encoded within the internal representations of these models. For example, LAMA, introduced by [51], employs prompting methods and leverages masked PLMs to assess this knowledge. Numerous studies have been conducted to achieve state-of-the-art results through text mining, paraphrasing [148], prompt tuning [149], and AutoPrompt techniques [150].

However, using PLMs as knowledge bases comes with several limitations: (1) they tend to learn shallow heuristics rather than developing a deep factual understanding [151], (2) their responses can be inconsistent [152], (3) they exhibit dependency on prompts and are prone to bias [153], and (4) they often underperform when dealing with long-tail scenarios [154].

Nevertheless, it is important to note that the research mentioned above differs from our research goals. Instead of focusing on measuring the internal knowledge of language models, our method aims to automatically extract new, reasonable knowledge graphs from PLMs based on minimal input.

5.3 Methodology

5.3.1 Problem Formalization

In this section, we present the components of our proposed framework to construct a KG by extracting purely from a given PLM. The framework consists of two main stages: the first stage focuses on the prompt creation task for generating paraphrased prompts from PLMs based on initial prompts and seed entity pairs while the second stage involves harvesting diverse and accurate knowledge from LMs. The core component is an efficient search with scoring functions aimed at discovering all possible entity pairs that compose the desired KG. Specifically, given seed entity pairs *sepas* and an initial prompt p of a particular relation r (e.g., "`AtLocation`" as illustrated in Figure 5.1), our framework generates a comprehensive knowledge graph G that includes new head and tail entities of r. This graph is enriched with head and tail entities pertinent to the specified relation r, thus expanding the breadth and depth of the generated information. Figure 5.1 illustrates our proposed

framework. We employ two practical methods: generate prompt candidates from the given input and discover meaningful entity tuples through seed pairs according to our efficient scoring functions.

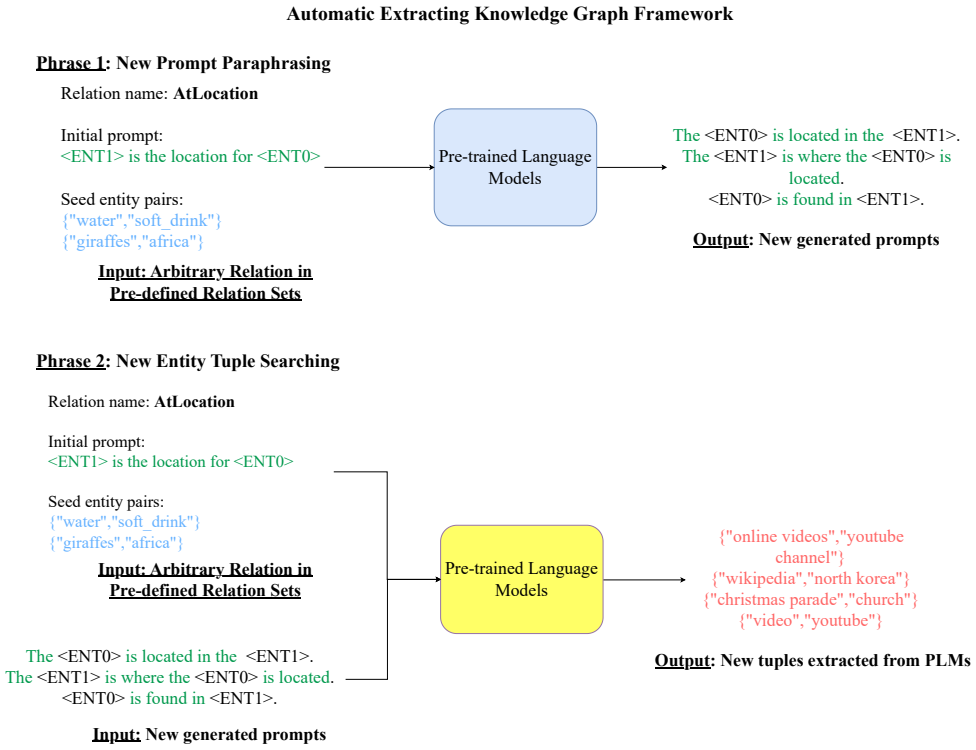


Figure 5.1: Our automatic extracting knowledge graph proposal framework: Probing PLMs to harvest a complete KG.

5.3.2 The New Prompt Paraphrasing

In this part, we introduce our unsupervised prompt generation framework. The idea behind our framework is based on two basic principles. First, the paraphrased prompt must contain seed entity pairs and the keywords of the initial prompt. Second, we recognize that rearranging these keywords while linking them with appropriate connectors can produce novel and grammatically sound paraphrases. In other words, our approach aims to create diverse permutations while ensuring that the paraphrase captures the core meaning of the original sentence. Here are the reasons why our approach relies on three key assumptions:

- **Semantic Preservation**: Using keywords ensures that the paraphrased sentences retain the core meaning of the original sentence.

Keywords act as anchors, preventing the paraphrase from straying too far from the initial prompt.

- **Expressive Flexibility**: Allowing keywords to be expressed using synonyms enables the generation of diverse versions without compromising the core meaning.
- **Order Flexibility**: Permitting different keyword orders makes the paraphrase grammatically correct and natural-sounding.

Based on these three assumptions, our unsupervised prompt generation framework is divided into four key components:

- **Keyword Extraction**: We extract important keywords from the initial prompt, along with the seed entity pairs, and generate all possible permutations of these keywords.
- **Keyword Substitution**: We use a lexical substitution system to generate the most appropriate synonyms for each keyword.
- **PLM-based Text Generation**: By leveraging the contextual understanding and linguistic proficiency of LMs, we use these keyword permutations to generate fluent sentences that retain all key information and maintain the order of keywords.
- **Candidate Prompt Scoring**: After generating a set of semantically consistent prompts with linguistic variety, a maximum of 20 prompts will be selected as the best candidates based on our scoring metrics.

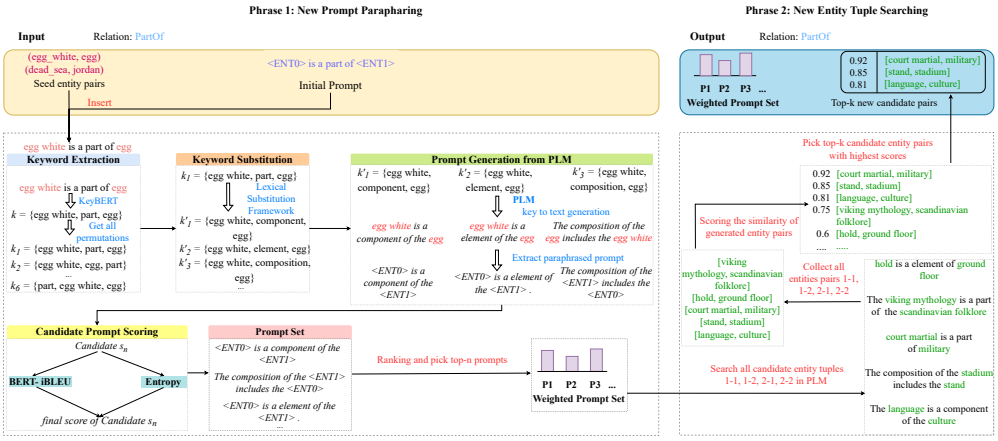


Figure 5.2: Our proposed framework for extracting internal knowledge from PLMs

Figure 5.2 illustrates a framework for generating diverse prompts from an initial input that includes a few pairs of seed entities and an initial prompt. Starting with the initial prompt, we insert any entity pair from the seed

entity pairs into the prompt to create a complete sentence s. The process can be broken down into several stages:

- **Keyword Extraction**: In the first stage, keyword extraction is performed on the sentence "`egg white is a part of egg`" using KeyBERT [155], which identifies the most relevant keywords in the sentence, resulting in a set of keywords:

  $$k = \{\texttt{eggwhite,part,egg}\}$$

  From this set, all permutations are generated, resulting in multiple keyword sets k_1, k_2, ..., k_6.
- **Keyword Substitution**: The second stage involves keyword substitution, where each permutation of keywords is processed through XLNet with embeddings [156] to produce the possible alternatives for each keyword without changing the meaning of the context. This framework generates substituted keywords to create various alternative expressions. For instance, $k_1 = \{\texttt{eggwhite,part,egg}\}$ may produce $k_1^{'} = \{\texttt{eggwhite,component,egg}\}$, $k_2^{'} = \{\texttt{eggwhite,element,egg}\}$, and $k_3^{'} = \{\texttt{eggwhite,composition,egg}\}$
- **Prompt Generation from PLM**: In the third stage, these substituted keyword sets are used to generate prompts with the assistance of the text-to-text framework (T5) [22], which takes keywords as inputs and generates sentences as outputs with the same meaning. T5 generates fluent sentences while preserving all key information and the order of keywords. For example, from $k_1^{'}$, the prompt "`egg white is a component of the egg`" is generated. Similarly, other keyword sets produce prompts like "`egg white is an element of the egg`" and "`The composition of the egg includes the egg white`". Differing from the baseline, we utilize T5 instead of GPT-3 [21] since T5 can be fine-tuned for specific tasks, generating more controlled and relevant output based on a given context or set of keywords. Additionally, the architecture of T5 treats all NLP tasks as text-to-text transformations, allowing greater flexibility in sentence generation tasks.
- **Candidate Prompt Scoring**: The fourth stage involves scoring the candidate prompts using two metrics: BERT-iBLEU and Certainty. BERT-iBLEU evaluates the semantic similarity between the original and generated prompts, while Certainty measures the certainty of the generated text. These scores help to assess the quality and relevance of each candidate prompt.
- **Prompt Set Formation**: Based on the scores from the previous stage, a set of high-quality paraphrased prompts is formed. This set includes top-scoring prompts such as "`<ENT0> is a component of the`

<ENT1>", "The composition of the <ENT1> includes the <ENT0>", and "<ENT0> is an element of the <ENT1>".

- **Ranking and Selection**: In the final stage, the paraphrased prompts are ranked based on their weighted scores. The top-n prompts are selected, resulting in the formation of the final weighted prompt set. In this paper, n is equal to 20. This set represents the most accurate and contextually appropriate interpretation of the initial prompt.

5.3.3 Candidate Prompt Scoring

Automatically generated prompts often lack precision, resulting in prompts that fail to capture the intended relational context. Such inaccuracies can adversely affect the model's performance in extracting the internal knowledge of PLMs, as knowledge retrieval requires high-quality prompts that are semantically consistent with the desired relation. To address this limitation, we propose a novel prompt scoring method to evaluate and enhance the quality of generated prompts. Specifically, we introduce two distinct scoring methods: (i) BERTScore combined with iBLEU, and (ii) Certainty. Using these scoring mechanisms, our approach offers a more comprehensive evaluation, facilitating the generation of high-quality prompts that are both accurate and better at capturing the semantics of a relation.

5.3.3.1 BERT-iBLEU

To evaluate the quality of prompt generation, we propose BERT-iBLEU, a metric designed to measure both meaning similarity and fluency between the source sentence s and candidate sentence c. To access semantic similarity, we use the BERTScore [157], which leverages the contextual embeddings of the BERT model to calculate the cosine similarity between reference and candidate sentences. This approach provides a robust similarity measure between the reference text and the generated text at both the word and sentence levels. The BERTScore metric ranges from 0 to 1, where a score of 1 signifies perfect similarity between the generated text and the reference text, and a score of 0 indicates no semantic overlap with the reference text.

To measure surface-form dissimilarity, we propose a metric called iBLEU (where i stands for inverse). BLEU [158] typically assesses the similarity of generated text to reference text, with scores also ranging from 0 to 1. A BLEU score nearing 1 indicates that the generated text is highly similar to the reference, whereas a score of 0 suggests significant differences from the reference corpus. iBLEU emphasizes the dissimilarity between the generated text and the reference text. A higher $\frac{1}{BLEU}$ value means that the generated

text is less similar to the reference, which can indicate greater dissimilarity or uniqueness. Our idea for iBLEU is that we can assess the fluency and distinctiveness of generated text, as it focuses on how different the generated text is from the reference. In this way, higher values of $\frac{1}{BLEU}$ may suggest that the generated text is more distinct and less similar to the reference.

By combining these two metrics, we hypothesize that the integration of these metrics will not only enhance the quality of the generated text but also provide valuable insights on the trade-off between semantic similarity and the "uniqueness". To facilitate the integration of iBLEU with other metrics, such as BERTScore, it is necessary to standardize the iBLEU score to a common range. Applying the sigmoid function to iBLEU ensures that the standardized iBLEU values smoothly transition between 0 and 1, making them directly comparable with BERTScore.

$$BERT-iBLEU(s,c) = BERT_{score}(s,c) + sigmoid\left(\frac{1+0.01}{BLEU_{score}(c)+0.01}\right) \tag{5.1}$$

where $BERT_{score}$ represents the BERTScore between the source sentence s and the candidate sentence c, $BLEU_{score}$ denotes the BLEU score between the source sentence s and the candidate sentence c; 0.01 is a small smoothing factor added to both the numerator and the denominator in the i-BLEU term.

5.3.3.2 Certainty

The BLEU metric evaluates the quality of generated texts, but ignores their certainty. The certainty associated with language models plays a crucial role in determining the reliability and authenticity of the generated output. In this paper, to evaluate the certainty of the generated prompt, we can use the entropy of the probabilistic generative model to measure the degree of uncertainty or randomness, where the higher values show greater uncertainty. Entropy, as a measure of uncertainty, proves particularly insightful in the context of language models as it quantifies the level of uncertainty associated with each word or sequence. Specifically, we use the formula to calculate the entropy of a generated sentence s, as given below, where *P(x)* is the probability of the word x. In our context, a lower entropy is desirable because it indicates that the output prompt is more certain and is focused on generating coherent and contextually appropriate sequences.

$$Certainty(s) = \frac{1}{Entropy(s)} \tag{5.2}$$

where:

$$Entropy(s) = \sum_{x} -P(x)log(P(x)) \tag{5.3}$$

5.3.3.3 Compatibility Score

The final compatibility score combines BERT-iBLEU and Certainty to provide a holistic evaluation of the quality of the prompt generated by PLMs. The combined metric is formulated as follows:

$$CompatibilityScore(s, c) = \alpha BERT - iBLEU(s, c) + \beta Certainty(c) \tag{5.4}$$

Here, s represents the initial prompt and c denotes the generated prompt. The α and β are weighting factors that balance the contributions of semantic similarity, syntactic fluency, and model certainty. In our setting, α and β are equal to 1. The BERT-iBLEU component ensures that the generated prompt is semantically and syntactically aligned with the reference, while the Certainty component evaluates the reliability of an automatically created prompt. This metric enables a nuanced assessment of text generation quality, ensuring that the prompts generated by the PLM are relevant and varied in response to the given initial prompts.

5.3.4 Efficient Knowledge Tuple Searching

In the context of this paper, the entity pairs extracted from PLMs using our framework can be categorized into four distinct types: 1-1, 1-2, 2-1, and 2-2. In this classification scheme, 1-1 denotes entity pairs where each entity consists of only one word, and 1-2 signifies cases where the first entity comprises a single word and the second entity consists of precisely two words. Conversely, 2-1 represents instances where the first entity comprises two words and the second entity is a single word, while 2-2 represents entity pairs where both entities are composed of exactly two words.

In contrast to using a single minimum heap for all entity pairs in BertNet, our method improves extracted fact count by employing four distinct heaps, each with a maximum size of 1,000, dedicated to searching entity pairs of types 1-1, 1-2, 2-1, and 2-2. This multi-heap approach allows efficient management and surpasses the single-heap system of BertNet in finding possible pairs of entities. During the candidate entity pair search, we implement a pruning strategy similar to the baseline but adapted to our multi-heap system. For each tuple type, we maintain a dedicated minimum heap to keep track of the log-likelihoods of the entity tuples. For instance, when searching for 1,000 entity tuples of type 1-1, we maintain a minimum

heap specifically for 1-1 tuples. The maximum size of this heap is 1,000, and the heap top serves as a threshold for future searches because it represents the 1,000th largest log-likelihood for that tuple type. Additionally, we restrict each entity to a maximum of 10 occurrences to enhance the diversity of the generated knowledge.

When searching for consistent tuples, if the log-likelihood at any time step is lower than the threshold of the corresponding heap, we can immediately prune the search. This ensures that the log-likelihood of the current tuple will not surpass any existing tuples in the heap. If a new entity tuple is found without being pruned, we remove the heap top and insert the log-likelihood of the new tuple into the heap.

Maintaining separate heaps for each type of tuple in our method ensures a more targeted and efficient pruning process, leading to better management of candidate entity pairs. This strategy offers several advantages. Firstly, it facilitates type-specific prioritization, optimizing each heap for its respective entity pair format, and potentially improving retrieval efficiency. Secondly, it reduces the risk that lower-priority entities of one type are overshadowed by high-priority entities of another type within a single, unified heap. This segregation ensures fairer competition within each category. Lastly, maintaining separate heaps not only enables efficient filtering and retrieval based on specific entity pair formats but also enhances the overall efficiency and accuracy of the entity pair generation process. Consequently, this strategic utilization of minimum heaps not only expands the pool of potential candidate entity pairs but also provides a broader spectrum of choices, enabling the selection of entity pairs that align more closely with the semantic context of the relation under consideration.

In summary, our strategy offers several advantages over using only a single minimum heap with a single threshold:

- **Optimized Thresholds**: The distinct thresholds allow for fine-tuning of the selection process within each entity pair type, aligning with their unique likelihood distributions and relevance criteria. This enhances precision and reduces false positives within each type.
- **Granular Insights**: Our 4-heaps approach facilitates a more nuanced understanding of entity relationships by revealing patterns and trends within each entity pair type. This granularity can inform downstream tasks and knowledge graph construction.

5.3.5 Scoring Functions for All Candidate Entity Pairs

In the approach of BertNet, the sole criterion to be considered is the log-likelihood of the top minimum heap, establishing a threshold to determine the reasonableness of extracted entity pairs. However, this simplistic strategy may result in the omission of numerous pairs of potential entities. To address this limitation, we propose multiple scoring functions to improve the selection process of harvested tuples. By integrating diverse scoring criteria, our method aims to enhance the identification of the most plausible and relevant entity pairs, thereby providing a more comprehensive selection process that surpasses the constraints of a singular log-likelihood threshold.

After harvesting a large number of candidate entity pairs, it is essential to implement a robust selection process to identify the most relevant and meaningful pairs. From this pool of candidates, we extract up to 1,000 of the most pertinent tuples for any given arbitrary relation. This selection process often relies on scoring functions that incorporate two key aspects: semantic equivalence and fluency. Using these criteria, our proposed framework ensures that the extracted knowledge is contextually accurate and linguistically coherent, thus enhancing the overall quality and reliability of the output knowledge.

5.3.5.1 Semantic Score

In recent research, BERTScore has been used to model the semantic distance between system-generated and reference texts to evaluate text generation systems. As depicted in Figure 5.3, BERTScore operates a “harder" alignment strategy, meaning that each word in one sequence is matched to the most semantically similar word in the other sequence, creating a one-to-one correspondence. This allows BERTScore to evaluate the semantic similarity between system-generated and reference texts by comparing individual word pairs in a one-to-one manner. In contrast, our method does not operate on a one-to-one alignment basis. Instead, our approach utilizes “soft" alignments, allowing for many-to-one mappings of semantically related words. This means that our approach can be used to map semantically related words from one sequence to their counterparts in another sequence. This approach solves a constrained optimization problem to determine the minimum effort required to transform one text into another, thereby providing a more flexible and comprehensive assessment of textual similarity.

Given the above considerations, our framework strategically employs BERTScore to generate diverse weighted prompts similar to the initial prompt and uses the idea of Word Mover's Distance (WMD) to select the

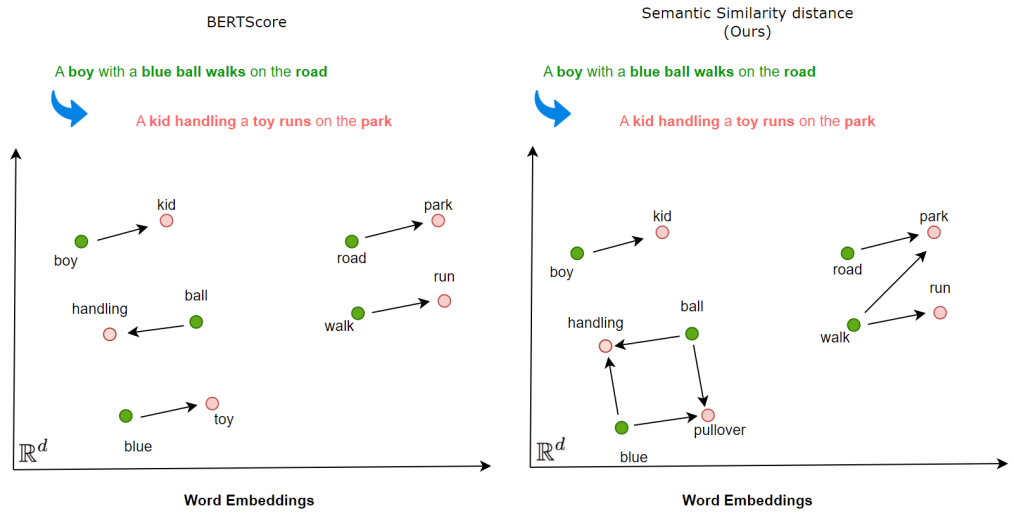


Figure 5.3: An illustration of BERTScore and our semantic score with OTP

most relevant tuples. The choice of BERTScore is based on its ability to provide a robust, hard-constrained one-to-one mapping of contextual embeddings, which ensures precise and direct semantic similarity measurement. BERTScore uses deep contextualized representations, allowing for accurate identification of prompts that closely match the initial prompt in meaning and context, thus maintaining the integrity of the original information and generating high-quality related prompts.

To harvest the relevant and reliable knowledge graph from PLMs, our semantic score is inspired by the idea of Word Mover's Distance [159] to define our non-contextualized word embedding metric. WMD, an instance of Earth Mover's Distance [160], is a well-studied method to measure text similarity using the optimal transport distance between the pre-trained word embeddings. It has proven effective for tasks such as document classification and retrieval [159]. This soft alignment approach, characterized by many-to-one mappings, measures the semantic distance between two sets of word embeddings by solving an optimal transport problem, effectively capturing both word-level and structural differences. This flexibility in alignment allows our approach to accurately filter candidate tuples to find those most relevant to the seed entity pairs, ensuring a high degree of contextual alignment and relevance.

Let x and y be two sentences viewed as sequences of one word: x^1 and y^1 consisting of n and m words.

$$x^1 = (w_1, w_2, ..., w_n), \quad y^1 = (w_1^{'}, w_2^{'}, ..., w_m^{'}) \tag{5.5}$$

Let w_i and $w_j^{'}$ be the word i in x^1 and word j in y^1, respectively. The bold symbol $\boldsymbol{w}_i \in \mathbb{R}^d$ represents the word vector representation to word w_i. In WMD, the weight of each word is uniformly assigned, and the distance function is the Euclidean distance between the embeddings of the word w_i and $w_j^{'}$. WMD defines the *transportation cost* matrix $C \in \mathbb{R}^d \times \mathbb{R}^d$ such that $C_{ij} = d(x_i^1, y_j^1)$ is the Euclidean distance between the embedding of the word i in x^1 and the embedding of the word j in y^1 as:

$$d(w_i, w_j^{'}) = \left\| \boldsymbol{w_i} - \boldsymbol{w_j^{'}} \right\|_2 \tag{5.6}$$

Therefore, the WMD distance between word sequences of 1-gram x^1 and y^1 with weighted 1-gram $\mathbf{t_{x^1}}$ and $\mathbf{t_{y^1}}$, where $\mathbf{t_{x^1}} \in \mathbb{R}_+^{|x^1|}$ is the non-negative real-valued vector of weights and one weight for each 1-gram of x^1, is defined in the following equation:

$$\begin{aligned} WMD(x^1, y^1) &= \min_{T \in \mathbb{R}^{|x^1| \times |y^1|}} \langle C, T \rangle, \\ &= \underset{T \in \mathbb{R}^{|x^1| \times |y^1|}}{minimize} \sum_{ij} C_{ij} T_{ij}, \\ \text{subject to} \quad & \mathbf{T_{ij}} \geq \mathbf{0}, \mathbf{T1} = \mathbf{t_{x^1}}, \mathbf{T}^\top \mathbf{1} = \mathbf{t_{y^1}} \end{aligned} \tag{5.7}$$

where $\mathbf{T}^\top$ denotes the transpose of $\mathbf{T}$, $\mathbf{1} \in \mathbb{R}^d$ is the vector of ones. Here, T is the *transportation flow* matrix with T_{ij} represents *how much* the $i-$th *1*-gram in x^1 travel to the $j-$th *1*-gram in y^1.

From the perspective of Equation 5.7, the distance is computed as the sum of the element-wise multiplication of the optimal transportation flow matrix T and the transportation cost matrix C. Thus, $WMD(x^1, y^1)$ is defined as the minimum distance between x^1 and y^1 weighted by $\mathbf{t_{x^1}}$ and $\mathbf{t_{y^1}}$.

Despite its intuitive formulation, the WMD often misaligns words with each other, resulting in semantic textual similarity (STS) performance that is less than that of recent methods such as SynWMD [161]. For instance, WMD might align “soup" with “burger" rather than “soup" with “ramen". This shortcoming arises because the WMD is based on the Euclidean distance, which mixes the weighting factor as norm and dissimilarity. The problematic nature of this mixing is evident in Equation 5.6, where the Euclidean transportation cost misjudges the similarity of word pairs as $\text{low}_{<SWP>}$ even when their meanings are $\text{close}_{<SYN>}$ but their concreteness or importance is very different $_{<CI>}$, e.g., “soup” and “ramen” ($\text{low}_{<SWP>}$).

To address this problem of the Euclidean distance, we first discuss the contributions of the norm and the direction of the word vectors.

- **Norm of the embedding vector of a word as weighting factor**: The norm of a word embedding vector plays a crucial role in determining its semantic contribution to the overall meaning of a sentence. In this work, we define λ and $\hat{u}$ as the norm and the direction vector of word embedding $\boldsymbol{w}$:

$$\lambda = \|\boldsymbol{w}\| \tag{5.8}$$

$$\hat{u} = \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|} = \frac{\boldsymbol{w}}{\lambda} \tag{5.9}$$

  From equation 5.9, we can conclude that $\boldsymbol{w} = \lambda \cdot \hat{u}$, and $\hat{u}$ is a unit vector whose length 1 ($\|\hat{u}\| = 1$).
- **Angle similarity between embedding vectors**: Angle similarity between embedding vectors measures the semantic relationship between words encoded in high-dimensional spaces. Cosine similarity is one method that quantifies the similarity between two vectors by measuring the cosine of the angle between them and returns a number measuring their similarity as:

$$cos(\boldsymbol{w}_i, \boldsymbol{w_j^{'}}) = \frac{\boldsymbol{w}_i \cdot \boldsymbol{w_j^{'}}}{\|\boldsymbol{w}_i\| \cdot \|\boldsymbol{w_j^{'}}\|} = {\boldsymbol{w}_i}^{\top} \boldsymbol{w_j^{'}} \tag{5.10}$$

Here is the relationship between the Euclidean distance and cosine similarity:

$$\begin{aligned} d(w_i, w_j^{'}) &= \sqrt{(\lambda_i \hat{u}_i - \lambda_j \hat{u}_j^{'})^{\top} (\lambda_i \hat{u}_i - \lambda_j \hat{u}_j^{'})} \\ &= \sqrt{(\lambda_i \hat{u}_i)^2 - 2\lambda_i \lambda_j \cos(\boldsymbol{w}_i, \boldsymbol{w_j^{'}}) + (\lambda_1 \hat{u}_j^{'})^2} \\ &= \sqrt{\lambda_i^2 + \lambda_j^2 - 2\lambda_i \lambda_j \cos(\boldsymbol{w}_i, \boldsymbol{w_j^{'}})} \end{aligned} \tag{5.11}$$

From equation 5.11, $d(\boldsymbol{w}_0, \boldsymbol{w}_1)$ is determined to be significantly large$_{<SWP>}$, even when the cosine similarity $cos(\boldsymbol{w}_0, \boldsymbol{w}_1)$ is large$_{<SYN>}$, so long as $\lambda_0^2 + \lambda_1^2$ is large$_{<CI>}$, indicating a substantial magnitude of the embedding vectors.

Figure 5.4 shows the cosine similarity and Euclidean distance between the embedding vectors of “soup", “ramen", “burger", and “bacon". Euclidean distance might judge “soup" and “bacon" as more similar than “soup" and “ramen" due to potentially larger magnitudes, despite the closer semantic relationship between “soup" and “ramen" reflected by a higher cosine similarity. This highlights the importance of considering cosine similarity when evaluating semantic relationships within embedding spaces, as it focuses on the directional alignment of vectors rather than their magnitudes.

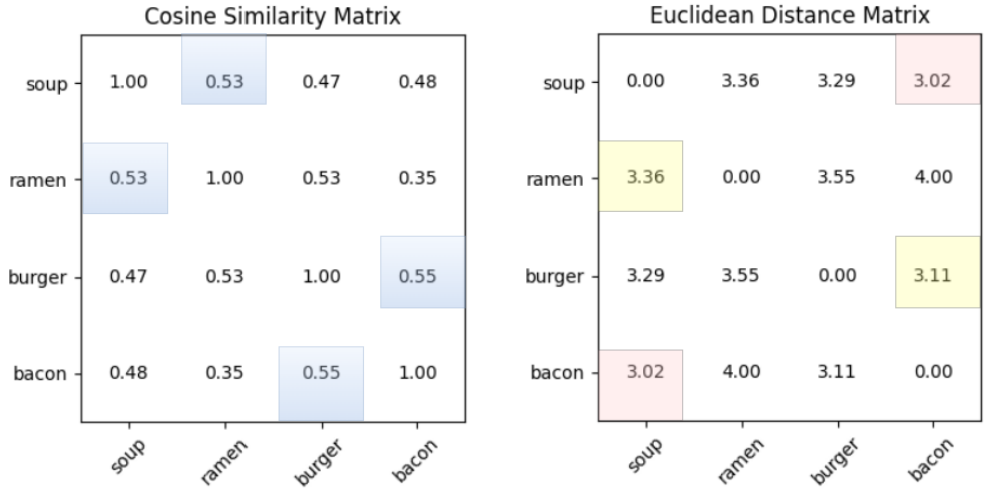


Figure 5.4: Comparative analysis of cosine similarity and Euclidean distance metrics for word embeddings derived from word2vec [1]. In the Euclidean distance matrix, the **lowest** value, indicating the correct similarity word, is colored yellow for each row, while **inappropriate** alignments are highlighted in pink. In the Cosine Similarity matrix, the **largest** value, indicating the correct similarity word, is highlighted in blue for each row.

Given the above considerations, we propose a simple but powerful sentence similarity measure that leverages the idea of WMD. The WMD approach initially used the static word embedding word2vec [1] to generate the embedded words. However, static embeddings like word2vec have limitations, as they lack the capacity to capture word order and compositionality. Therefore, by incorporating contextualized embeddings such as BERT [20], which are capable of capturing relationships between words within a sequence, our objective is to achieve a more comprehensive and dynamic representation of linguistic structures, improving the accuracy of WMD-based word similarity calculations. Additionally, for the cost function, we use the cosine similarity between the embedding representations of *1*-gram sequences:

$$d(w_i, w_j^{'}) = 1 - cos(\boldsymbol{w}_i, \boldsymbol{w_j^{'}}) \tag{5.12}$$

$$\boldsymbol{w}_i = BM25(w_i) \cdot \boldsymbol{w}_i \tag{5.13}$$

where $BM25(w_i)$ is the BM25 of word i in x^1 computed from all sentences in the corpus $\mathcal{C} = \{S_1, S_2, \ldots, S_N\}$, and $\boldsymbol{w}_i$ is its word embedding. Additionally, the weight associated to *1*-gram x_i^1 is given by:

$$t_{x_i^1} = BM25(w_i) \tag{5.14}$$

$$BM25(w_i) = \sum_{i=1}^{N} \text{idf}(w_i) \cdot \frac{f(w_i, S_i) \cdot (k_1 + 1)}{f(w, S_i) + k_1 \cdot \left(1 - b + b \cdot \frac{|S_i|}{avgdl}\right)} \tag{5.15}$$

where $f(w, S_i)$ represents the frequency of the word w in sentence S_i, while $|S_i|$ denotes the length of the sentence S_i in terms of the number of words. The parameter avgdl is the average sentence length across all sentences in the corpus $\mathcal{C}$. N is the total number of sentences, and $n(w_i)$ refers to the number of sentences that contain the word w_i. The parameters k_1 and b are set by default to 1.2 and 0.75, respectively. Finally, IDF(w_i) is the Inverse Document Frequency of the word w_i, is defined as:

$$IDF(w_i) = \ln\left(\frac{N - n(w_i) + 0.5}{n(w_i) + 0.5}\right) \tag{5.16}$$

In our context, the semantic similarity score between the generated pair *gen_pair* and seed entity pairs *sepas* is computed as:

$$s_{semantic}(sepas, gen\_pair) = \sum_{p} WMD(ip_p^1, ip_{gen\_pair}^1) \tag{5.17}$$

where ip is an initial prompt, *sepas* are the list of p seed entity pairs, and ip_p^1 is the sentence viewed as a sequence of 1-gram in which the masks in an initial prompt are replaced by the *p-th* seed entity pair. Similarly, $ip_{gen\_pair}^1$ is the sentence represented as a sequence of 1-gram, with the masks in the initial prompt replaced by the generated tuple.

5.3.5.2 Fluency-Weighted IDF

The primary condition for a reasonable tuple is fluency, hence we compute the fluency value of an initial prompt with masks replaced by a generated tuple based on perplexity. Fluency represents how smoothly and naturally a generated text reads, without considering whether it accurately conveys the original meaning. It plays a pivotal role in determining the efficacy of language models in seamlessly conveying information. The selection of perplexity as a fluency metric for evaluating tuples harvested from PLMs is motivated by its ability to measure the uncertainty or unpredictability of a language model. The perplexity metric measures how well the probability distribution predicted by a model proposed by GPT [162], with higher perplexity meaning lower fluency. In other words, a lower value of perplexity reflects better performance, suggesting that PLMs are better at predicting or understanding the sequence of words in the input sentence. In the context of tuple generation from PLMs, perplexity provides a quantitative measure

of the linguistic smoothness in the generated output. The perplexity is calculated as the exponent of the mean of the log-likelihood of all the words in an input sequence. Given ip is an initial prompt, ip_{gen_pair} is the sentence in which masks in an initial prompt are replaced by an extracted entity pair.

$$s_{perplexity}(ip_{gen\_pair}) = exp\{\frac{1}{n}\sum_{i=1}^{n} log_{P_{LM}}(w_i|w_{<i})\} \tag{5.18}$$

where $log_{P_{LM}}(w_i|w_{<i})$ represents the log-likelihood of the i-th token given the preceding tokens $w_{<i}$ according to our model. After computing the perplexity value, we obtain the fluency score by:

$$s_{fluency}(ip_{gen\_pair}) = \frac{1}{f_{perplexity}(ip_{gen\_pair})}\sum_{i=1}^{n} idf(w_i) \tag{5.19}$$

where $idf(w_i)$ is the IDF of work w_i computed from ip_{gen_pair}. The IDF is a crucial metric in information retrieval and text mining, used to evaluate how important a word is within a given document relative to a collection or corpus of documents. In this paper, we apply IDF to capture the discriminative power of individual tokens by considering their inverse prevalence between documents. Tokens with lower IDF scores appear more frequently, potentially signifying common knowledge. In contrast, tokens with high IDF scores are rarer and might contribute to factuality and uniqueness within the extracted tuple. By combining IDF with Perplexity, we can potentially achieve a more robust and informative ranking of extracted tuples. The high perplexity, along with the high IDF, could indicate a genuinely novel and informative fact, worthy of further investigation and potential inclusion in the knowledge base. This combined approach might outperform methods relying solely on perplexity or IDF, as it leverages the strengths of both metrics to capture both novelty and factual uniqueness. In other words, combining perplexity and IDF at the token level presents a promising avenue for enhancing the ranking of newly extracted tuples from PLMs. By harnessing the complementary strengths of these metrics, we can potentially improve the quality and informativeness of the knowledge extracted from neural language models.

5.3.5.3 Comprehensive Score

Finally, we leverage the linear combination of two values: $s_{semantic}$, $s_{fluency}$ to compute the comprehensive score s_{final} as follows:

$$\begin{aligned} \lambda_{sem} &= sigmoid(s_{semantic}(sepas, gen\_pair)) \\ \lambda_{flu} &= sigmoid(s_{fluency}(ip_{gen\_pair})) \\ s_{final}(gen\_pair) &= \lambda_{sem} \cdot s_{semantic}(sepas, gen\_pair) \\ &\quad + \lambda_{flu} \cdot s_{fluency}(ip_{gen\_pair}) \end{aligned} \tag{5.20}$$

By leveraging the sigmoid function, two lambda values are derived from the respective input values, effectively transforming them onto a uniform scale between 0 and 1. This transformation enables the model to prioritize high-scoring aspects (closer to 1) while gracefully diminishing the influence of lower-scoring ones (closer to 0). These lambda values act as weights, emphasizing the relative importance of each metric in the final score calculation. This adaptability is crucial for capturing the nuanced interplay between semantic coherence and fluency in evaluating generated text. The final score is obtained by multiplying the weighted perplexity by the weighted semantic similarity, capturing the appropriate balance between semantic accuracy and natural language flow, ultimately contributing to a more robust and informative comprehensive score.

Our method offers several advantages over traditional evaluation metrics that rely solely on perplexity or semantic similarity. By incorporating both aspects, it provides a more holistic assessment of the extracted tuples from PLMs, ensuring that they are not only grammatically correct and fluent but also semantically meaningful and relevant to the context. Moreover, the use of lambda weights, generated from the sigmoid functions, allows flexibility in adjusting the relative importance of each metric for a particular tuple.

5.4 Experiment Results

5.4.1 Dataset

In this study, we evaluate our framework with the relation datasets ConcepNet [163] and LAMA [51]. Similarly to BertNet [28], we only use 20 relations of Conceptnet (e.g., `AtLocation`, `HasA`, `IsA`) and 20 relations of LAMA (e.g., `P279_subclass_of`, `P37_official_language`) to mine diverse tuples from PLMs. The initial prompt and a few seed entity pairs (maximum of 5 tuples) of each relation are randomly taken from the ConceptNet and LAMA knowledge bases.

To access the novelty of our framework for extracting tuples from the ConceptNet dataset, we use the four datasets of the ConceptNet knowledge base completion task data [1].

5.4.2 Evaluation Metrics

Starting with a few pairs of seed entities *sepas* and an initial prompt p describing a relation r, our framework builds a knowledge graph G containing new entities related to r.

Ideally, our goal is to compare G with a ground truth graph formed by extending *seeds*. Given such a graph, we can calculate precision and extracted fact count by comparing the sets of triplets from both the gold and predicted sets. Nevertheless, using PLMs for graph generation presents the challenge of not having a ground-truth graph to compare the generated sets of tuples. Initially, we suppose that WikiData [164] or DBpedia [165] can verify the predicted sets of tuples, but these knowledge bases miss many correct tuples from the predicted sets by PLMs, highlighting their incompleteness. Consequently, our motivation is to find a method to verify the accuracy of the generated facts.

To overcome this challenge, we propose the following concepts of precision and extracted fact count:

Precision: The primary aim of our paper is to enhance knowledge extraction from LLMs. The evaluation of these methods using another advanced LLM, such as Gemini [2], is justified by the assumption that a robust LLM can provide accurate and nuanced assessments. Gemini, known for its advanced capabilities and extensive training data, serves as a reliable benchmark to judge the quality of extracted tuples. Its comprehensive coverage of various knowledge domains ensures that the evaluation is conducted with high accuracy and relevance. Unlike traditional knowledge bases, which often miss correct tuples due to their static and incomplete nature, Gemini's expansive and dynamic training allows it to identify correct tuples more reliably, minimizing the risk of overlooking valid information.

Given a tuple generated and prompt sentence, the Gemini judge gives this tuple a binary relevance label: correct or incorrect. In this manner, we can gauge how many accurate tuples extracted from PLMs were found. In this scenario, tuples, which consist of subject-object statements with a certain relation, are extracted without direct supervision. Gemini, with its robust contextual comprehension and semantic analysis abilities, can be

[1] https://home.ttic.edu/~kgimpel/commonsense.html

[2] https://gemini.google.com/

used to assess the accuracy, coherence, and relevance of these tuples. It evaluates whether the relationships and entities identified are meaningful and correctly represent the input data. This evaluation process is crucial for refining the quality of unsupervised extraction methods, ensuring that the generated tuples are not only syntactically correct but also semantically valid. Consequently, this enhances the overall reliability and applicability of the extracted information, making Gemini an essential tool in advancing the field of knowledge extraction from PLMs.

To be more specific, to check the correctness of an extracted tuple *(h,t)* belonging to a specific relation r, we form a query containing exactly h and t. For instance, to check the correctness of the tuple **(video, YouTube)** in the relation `AtLocation`, we use the initial prompt of this relation: "`<ENT1> is the location for <ENT0>`". Two masks in this prompt are replaced by "**video**" and "**YouTube**", respectively. Gemini then processes this query with the template: Is the statement correct or not: "`YouTube is the location for video`" and returns the result. If the response is correct, we assume that the tuple *(h,t)* is valid.

To achieve a more objective and rigorous evaluation, our second approach proposes using the ConceptNet knowledge base completion task as a ground truth. Specifically, we employ the ConceptNet training set, which contains 600,000 tuples, to validate the accuracy of the output KGs produced by our framework and the BertNet baseline. In this approach, we compare the semantic similarity between the generated tuples and the ground-truth tuples from ConceptNet. This ensures a robust evaluation, confirming that the generated KGs are accurately aligned with established knowledge. By anchoring our assessment in a well-established knowledge base, this method offers a more concrete and objective measure of the accuracy of the KG outputs.

Extracted Fact Count: Estimating extracted fact count is not possible since we do not have any ground-truth graph. Following the notion of open information extraction [166], where it is impossible to know the set of all true facts, the convention is to report the number of generated facts only. Inspired by this notion, in this paper, we only report the number of entities extracted as the extracted fact count value.

Diversity: To assess the degree of uniqueness or newness in the harvested KG, we calculate the number of unique entities in this KG. A high count of unique entities indicates the diversity and variety of entities present within a KG.

Novelty: To evaluate the new, uniquely generated tuples in comparison to the original dataset, we refer to the proportion of extracted entities that do not appear in ConceptNet datasets. A high novelty value suggests that

the output KG includes information not present in the original dataset. This introduces new facts, relationships, or insights that were not part of the initial data. KGs with a high novelty value contribute to knowledge discovery by producing diverse facets and information that might not have been part of the original dataset.

Noisy percentage. In addition to accuracy, we explicitly report the *noisy percentage* of the extracted knowledge graph, defined as the proportion of incorrectly extracted tuples among all extracted tuples under the same evaluation protocol as Tables 5.3 and 5.5. Accordingly, it is complementary to the reported extraction accuracy, so Noisy(%) = 100 − Precision(%).

5.4.3 Experimental Results

To evaluate our automatic framework for extracting internal knowledge of PLMs, we compare our generated KG results extracted from BERT-large and RoBERTa-large, denoted as K-BLOOM$_{\text{BERT-LARGE}}$ and K-BLOOM$_{\text{RoBERTA-LARGE}}$, with the output KGs in BERTNET harvested from BERT-large and RoBERTa-large, denoted as BERTNET$_{\text{BERT-LARGE}}$ and BERTNET$_{\text{RoBERTA-LARGE}}$, under three settings of prompts:

- **Initial prompt**: Without considering the effectiveness of multiple generated prompts from our framework, we use an initial prompt for each relation to harvest knowledge from PLMs. The empirical results are shown in §3.4.3.1.
- **Top-1 prompt generated (top-1 prompt)**: Similar to using the initial prompt as the sole prompt for extracting internal knowledge from PLMs, in this setting, we utilized the generated prompt with the highest weight (§5.3.2) during the knowledge search stage. The empirical results are shown in §5.4.3.2.
- **Multiple generated prompts (multi-prompts)**: To assess the impact of our method on automatically generating prompts as described in §5.3.2, we use all automatically generated prompts for knowledge extraction. The empirical results are shown in §5.4.3.3.

Furthermore, we evaluate the ability of K-BLOOM in the ConceptNet knowledge base to investigate the efficiency of our proposed method to generate reasonable pairs of entities for the Knowledge Graph Completion task. The detailed results of the evaluation are described in §5.4.3.4.

5.4.3.1 Evaluation of output KGs with initial prompt

In this section, we present the empirical evaluation of our knowledge graph expansion method, which constructs informative knowledge subgraphs expanded by existing seed entities. We begin by reporting results that demonstrate the ability of K-BLOOM to extract entity pairs from PLMs using only an initial prompt. We evaluated the diversity, novelty, and accuracy of the tuples extracted by our framework using the ConceptNet dataset and compared our results with those obtained by BERTNET, as shown in Table 5.1, Table 5.2 and Table 5.3.

Table 5.1: Diversity results (#tuples) of output KG from BERTNET and our K-BLOOM methods on ConceptNet dataset.

Method	**Number of Tuples**	**Diversity**	**Extracted Fact Count**
Initial Prompt Approach			
BERTNET$_{\text{BERT-LARGE}}$	16,244	9,076	32,488
$K-Bloom_{BERT-large}(Ours)$	18,800	25,288	37,600
BERTNET$_{\text{ROBERTA-LARGE}}$	18,566	12,700	37,132
$K-Bloom_{RoBERTa-large}(Ours)$	**19,712**	**26,252**	**39,424**
Top-1 Prompt approach			
BERTNET$_{\text{BERT-LARGE}}$	13,958	7,813	27,916
$K-Bloom_{BERT-large}(Ours)$	19,599	25,307	39,198
BERTNET$_{\text{ROBERTA-LARGE}}$	17,685	11,681	35,370
$K-Bloom_{RoBERTa-large}(Ours)$	**19,633**	**25,890**	**39,266**
Multi-Prompts approach			
BERTNET$_{\text{BERT-LARGE}}$	14,052	6,679	28,104
$K-Bloom_{BERT-large}(Ours)$	19,860	25,356	39,720
BERTNET$_{\text{ROBERTA-LARGE}}$	17,879	10,824	35,758
$K-Bloom_{RoBERTa-large}(Ours)$	**20,000**	**26,461**	**40,000**

As shown in Table 5.1, the evaluation is based on three key metrics: the number of tuples, diversity, and extracted fact count. Our preliminary experiments reveal insightful findings regarding the diversity achieved by each method. In particular, K-BLOOM$_{\text{BERT-LARGE}}$ and K-BLOOM$_{\text{ROBERTA-LARGE}}$ demonstrate substantial increases in the number of tuples, with 18,800 and 19,712 tuples, respectively, surpassing BERTNET$_{\text{BERT-LARGE}}$ and BERTNET$_{\text{ROBERTA-LARGE}}$ by 2,556 and 1,146 tuples. Furthermore, the diversity metric provides a nuanced perspective on the variety of information encapsulated by each model. K-BLOOM$_{\text{BERT-LARGE}}$ and K-BLOOM$_{\text{ROBERTA-LARGE}}$ exhibit diversity scores of 25,288 and 26,252, showing a remarkable improvement over BERTNET$_{\text{BERT-LARGE}}$ by 16,212 entities and BERTNET$_{\text{ROBERTA-LARGE}}$ by

13,138 entities. This significant increase in diversity emphasizes the enhanced capability of our framework in comprehensively covering a broader spectrum of concepts, leading to a richer and more diverse representation of KG's completeness.

We also evaluate the novelty of our resulting KG on four ConceptNet datasets. Table 5.2 presents a comprehensive analysis that compares the performance of K-BLOOM$_{\text{BERT-LARGE}}$ and K-BLOOM$_{\text{RoBERTA-LARGE}}$ with BERTNET$_{\text{BERT-LARGE}}$ and BERTNET$_{\text{RoBERTA-LARGE}}$ across four datasets: ConceptNet_train100k, ConceptNet_train300k, ConceptNet_train600k, and ConceptNet_test. The results in Table 5.2 indicate that our K-BLOOM methods outperform the baseline methods, demonstrating a significant improvement in concept understanding. Specifically, for ConceptNet_train600k, the percentage improvement of K-BLOOM$_{\text{BERT-LARGE}}$ over BERTNET$_{\text{BERT-LARGE}}$ is substantial at 34.01%, highlighting the scalability and robustness of our framework. These observed percentage improvements underscore the novelty and promise of our framework, not only in the context of specific training datasets but also in its ability to generalize well to unseen data.

Table 5.2: Novelty results (%) of output KG from BERTNET and our K-BLOOM methods on ConceptNet dataset.

Method	ConceptNet_train100k	ConceptNet_train300k	ConceptNet_train600k	ConceptNet_test
Initial Prompt Approach				
BERTNET$_{\text{BERT-LARGE}}$	71.00	63.97	46.55	95.41
$K-Bloom_{BERT-large}(Ours)$	89.40	85.82	80.56	98.52
BERTNET$_{\text{RoBERTA-LARGE}}$	80.92	75.14	62.53	97.15
$K-Bloom_{RoBERTa-large}(Ours)$	**92.70**	**89.73**	**86.02**	**99.00**
Top-1 Prompt Approach				
BERTNET$_{\text{BERT-LARGE}}$	63.33	54.50	41.80	93.8
$K-Bloom_{BERT-large}(Ours)$	89.08	85.00	80.41	98.48
BERTNET$_{\text{RoBERTA-LARGE}}$	72.01	64.24	53.04	95.47
$K-Bloom_{RoBERTa-large}(Ours)$	**90.88**	**87.53**	**84.43**	**98.70**
Multi-Prompts Approach				
BERTNET$_{\text{BERT-LARGE}}$	60.83	53.60	36.59	92.54
$K-Bloom_{BERT-large}(Ours)$	89.78	86.11	80.70	98.52
BERTNET$_{\text{RoBERTA-LARGE}}$	72.96	66.15	52.40	95.32
$K-Bloom_{RoBERTa-large}(Ours)$	**92.03**	**88.72**	**85.45**	**99**

The results shown in Table 5.3 illustrate the consistent superiority of our framework over BERTNET across multiple instances. Specifically, both K-BLOOM$_{\text{BERT-LARGE}}$ and K-BLOOM$_{\text{RoBERTA-LARGE}}$ outperform the baseline models in terms of precision in ConceptNet relations when using the Initial Prompt approach. Our K-BLOOM$_{\text{BERT-LARGE}}$ achieves an accuracy of 85.12%, exceeding the 79.80% accuracy of BERTNET$_{\text{BERT-LARGE}}$. Furthermore, K-BLOOM$_{\text{RoBERTA-LARGE}}$ consistently outperforms BERTNET$_{\text{RoBERTA-LARGE}}$ in precision, showing a notable 4.17 percentage point increase. Comparison of precision values emphasizes the reliability and efficiency of our scoring functions in accurately identifying relationships between entities, thus

contributing to a more effective way to extract internal knowledge from PLMs. Our method consistently generates higher-quality entity pairs than BERTNET, even when limited to the initial prompt. These results suggest that the semantic scoring component is a key factor in improving the accuracy of knowledge extraction from PLMs. In addition to extraction accuracy, Table 5.3 also reports the noisy percentage, which directly quantifies the fraction of incorrect tuples in the extracted knowledge graph under the same evaluation protocol. The noisy percentage results further strengthen our conclusion that K-BLOOM produces more reliable knowledge than BERTNET. Under the Initial Prompt approach, $\text{BERTNET}_{\text{BERT-LARGE}}$ exhibits a noisy percentage of 20.20%, whereas $\text{K-BLOOM}_{\text{BERT-LARGE}}$ reduces this value to 14.88%. Similarly, $\text{BERTNET}_{\text{ROBERTA-LARGE}}$ has a noisy percentage of 19.09%, while $\text{K-BLOOM}_{\text{ROBERTA-LARGE}}$ reduces it to 14.92%. This advantage persists under stronger prompting strategies. With the Top-1 Prompt approach, the noisy percentage decreases from 19.58% to 14.86% for the BERT-large backbone and from 21.35% to 15.73% for the RoBERTa-large backbone when moving from BERTNET to K-BLOOM. With the Multi-Prompts approach, $\text{BERTNET}_{\text{BERT-LARGE}}$ shows 17.64% noise, whereas $\text{K-BLOOM}_{\text{BERT-LARGE}}$ lowers it to 13.89%, and for RoBERTa-large the noise decreases from 18.86% to 16.22%. Overall, the consistent reduction in noisy percentage indicates that our semantic scoring mechanism not only improves the correctness of extracted relations but also actively suppresses spurious entity–relation pairs, resulting in a cleaner knowledge graph that is more suitable for downstream evidence-based reasoning.

Table 5.3: Precision and noisy percentage (%) of the output KG from the baseline and our K-BLOOM method on the ConceptNet dataset.

Method	**Precision (%)**	**Noisy (%)**
Initial Prompt Approach		
BERTNET$_{\text{BERT-LARGE}}$	79.80 (± 0.73)	20.20 (± 0.73)
K-Bloom$_{BERT-large}$ (Ours)	**85.12** (± 0.71)	**14.88** (± 0.71)
BERTNET$_{\text{ROBERTA-LARGE}}$	80.91 (± 0.65)	19.09 (± 0.65)
K-Bloom$_{RoBERTa-large}$ (Ours)	85.08 (± 0.69)	14.92 (± 0.69)
Top-1 Prompt Approach		
BERTNET$_{\text{BERT-LARGE}}$	80.42 (± 0.68)	19.58 (± 0.68)
K-Bloom$_{BERT-large}$ (Ours)	**85.14** (± 0.68)	**14.86** (± 0.68)
BERTNET$_{\text{ROBERTA-LARGE}}$	78.65 (± 0.90)	21.35 (± 0.90)
K-Bloom$_{RoBERTa-large}$ (Ours)	84.27 (± 0.74)	15.73 (± 0.74)
Multi-Prompts Approach		
BERTNET$_{\text{BERT-LARGE}}$	82.36 (± 0.65)	17.64 (± 0.65)
K-Bloom$_{BERT-large}$ (Ours)	**86.11** (± 0.68)	**13.89** (± 0.68)
BERTNET$_{\text{ROBERTA-LARGE}}$	81.14 (± 0.73)	18.86 (± 0.73)
K-Bloom$_{RoBERTa-large}$ (Ours)	83.78 (± 0.73)	16.22 (± 0.73)

Comparison of precision values emphasizes the reliability and efficiency of our scoring functions in accurately identifying relationships between entities, thus contributing to a more effective way to extract internal knowledge from PLMs. According to Figure 5.5, the results demonstrate a similar trend, in which K-BLOOM consistently outperforms BERTNET on both BERT-large and RoBERTa-large language models, indicating that our method consistently produces higher quality entity pairs compared to BERTNET, even when using only the initial prompt. This suggests that the semantic scoring component is a key factor in improving the accuracy of harvesting KGs from PLMs.

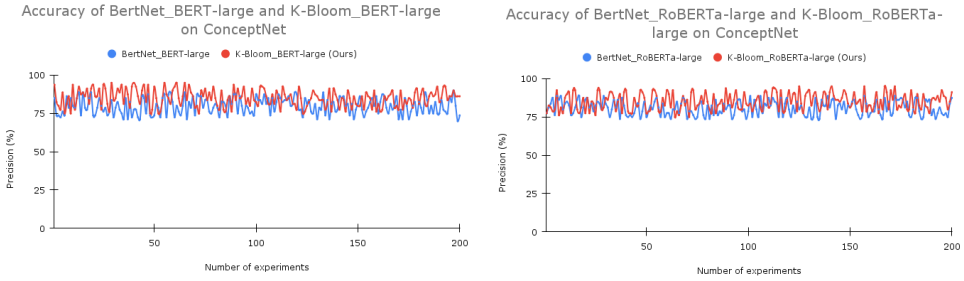


Figure 5.5: Knowledge extraction accuracy between our approach and BERTNET on ConceptNet using an initial prompt setting, with using BERT-LARGE and ROBERTA-LARGE as the PLMs. The left image shows the precision of both methods with BERT-LARGE, while the right image displays the precision of both methods with ROBERTA-LARGE.

In addition to the ConceptNet dataset, we conducted an experiment to extract internal knowledge from PLMs using 20 relations from the LAMA dataset. The statistics of our resulting KG are listed in Table 5.4. It is evident that K-BLOOM$_{\text{BERT-LARGE}}$ and K-BLOOM$_{\text{ROBERTA-LARGE}}$ outperform BERTNET, showing a higher number of tuples, greater diversity and greater extracted fact count, indicating their potential superiority in the knowledge extraction task. Furthermore, the analysis of precision for the tuples discovered from PLMs using our method, compared to BERTNET reveals significant insights. Across 200 experiments conducted on the LAMA dataset with the initial prompt option, the results show notable differences in performance between the models, highlighting the effectiveness of our approach.

As demonstrated in Table 5.5, BERTNET$_{\text{BERT-LARGE}}$ achieves an accuracy of 81.47%. In contrast, our approach, K-BLOOM$_{\text{BERT-LARGE}}$, exhibits superior consistency and enhanced performance, achieving an accuracy of 84.54%. This improvement underscores the effectiveness of our method in significantly enhancing the accuracy of tuples generated from PLMs, thereby validating its robustness in producing more reliable and precise knowledge representations. Moreover, with RoBERTa large, BERTNET$_{\text{ROBERTA-LARGE}}$ achieves an accuracy of 81.64%, while K-BLOOM$_{\text{ROBERTA-LARGE}}$ exceeds BERTNET with a higher accuracy of 85.01%. These results suggest that our approach consistently delivers improved accuracy in various tests compared to BERTNET. Additionally, the data indicate that the angular measurement of word vectors has become more precise, enhancing its effectiveness as a metric for word dissimilarity. This precision in vector angle measurement underlines the robustness of our method in accurately capturing semantic nuances, further reinforcing the superiority of our approach in language

Table 5.4: Diversity results (#tuples) of output KG from BERTNET and our K-BLOOM method on LAMA dataset.

Method	Number of Tuples	Diversity	Extracted Fact Count
Initial Prompt Approach			
$\text{BERTNET}_{\text{BERT-LARGE}}$	13,976	7,978	27,952
$K - Bloom_{BERT-large}(Ours)$	15,558	18,520	31,116
$\text{BERTNET}_{\text{ROBERTA-LARGE}}$	15,465	10,963	30,930
$K - Bloom_{RoBERTa-large}(Ours)$	**17,642**	**21,020**	**35,284**
Top-1 Prompt Approach			
$\text{BERTNET}_{\text{BERT-LARGE}}$	8,424	4,523	16,848
$K - Bloom_{BERT-large}(Ours)$	18,187	21,516	36,374
$\text{BERTNET}_{\text{ROBERTA-LARGE}}$	11,520	8,094	23,040
$K - Bloom_{RoBERTa-large}(Ours)$	**19,820**	**24,390**	**39,640**
Multi-Prompts Approach			
$\text{BERTNET}_{\text{BERT-LARGE}}$	8,424	4,523	16,848
$K - Bloom_{BERT-large}(Ours)$	18,187	21,516	36,374
$\text{BERTNET}_{\text{ROBERTA-LARGE}}$	11,520	8,094	23,040
$K - Bloom_{RoBERTa-large}(Ours)$	**19,820**	**24,390**	**39,640**

model evaluations. The substantial improvement in accuracy observed in $\text{K-BLOOM}_{\text{ROBERTA-LARGE}}$ indicates that our method optimizes the retrieval of relevant information from PLMs, leading to more precise results. These findings highlight the robustness of our approach and its capacity to extract relevant knowledge with minimal prompt engineering. In addition to accuracy, Table 5.5 also reports the noisy percentage, which offers a complementary view of extraction quality by measuring the proportion of incorrect tuples in the output KG. The noisy percentage results consistently favor K-BLOOM across all prompting settings and both PLM backbones. Under the Initial Prompt approach, $\text{BERTNET}_{\text{BERT-LARGE}}$ yields a noisy percentage of 18.53%, while $\text{K-BLOOM}_{\text{BERT-LARGE}}$ reduces it to 15.46%. For the RoBERTa-large backbone, the noisy percentage decreases from 18.36% with $\text{BERTNET}_{\text{ROBERTA-LARGE}}$ to 14.99% with $\text{K-BLOOM}_{\text{ROBERTA-LARGE}}$. Under the Top-1 Prompt approach, $\text{BERTNET}_{\text{BERT-LARGE}}$ records 20.74% noise, whereas $\text{K-BLOOM}_{\text{BERT-LARGE}}$ lowers it to 15.27%, and for RoBERTa-large the noise decreases from 21.25% to 15.96%. Under the Multi-Prompts approach, the noisy percentage is 20.18% for $\text{BERTNET}_{\text{BERT-LARGE}}$ and 15.20% for $\text{K-BLOOM}_{\text{BERT-LARGE}}$, while for RoBERTa-large it decreases from 20.60% to 15.81%. These results indicate that the improvements of K-BLOOM are not limited to higher accuracy; rather, the method consistently reduces the amount of incorrect extracted knowledge, producing a cleaner and more

reliable KG that better supports downstream retrieval and reasoning.

Table 5.5: Precision and noisy percentage (%) of the output KG from BertNet and our K-Bloom method on the LAMA dataset.

Method	**Precision (%)**	**Noisy (%)**
Initial Prompt Approach		
BertNet$_{\text{BERT-large}}$	81.47 (± 0.63)	18.53 (± 0.63)
K-Bloom$_{BERT-large}$ (Ours)	**84.54** (± 0.61)	**15.46** (± 0.61)
BertNet$_{\text{RoBERTa-large}}$	81.64 (± 0.56)	18.36 (± 0.56)
K-Bloom$_{RoBERTa-large}$ (Ours)	85.01 (± 0.55)	14.99 (± 0.55)
Top-1 Prompt Approach		
BertNet$_{\text{BERT-large}}$	79.26 (± 0.72)	20.74 (± 0.72)
K-Bloom$_{BERT-large}$ (Ours)	**84.73** (± 0.73)	**15.27** (± 0.73)
BertNet$_{\text{RoBERTa-large}}$	78.75 (± 0.72)	21.25 (± 0.72)
K-Bloom$_{RoBERTa-large}$ (Ours)	84.04 (± 0.71)	15.96 (± 0.71)
Multi-Prompts Approach		
BertNet$_{\text{BERT-large}}$	79.82 (± 0.75)	20.18 (± 0.75)
K-Bloom$_{BERT-large}$ (Ours)	**84.80** (± 0.68)	**15.20** (± 0.68)
BertNet$_{\text{RoBERTa-large}}$	79.40 (± 0.74)	20.60 (± 0.74)
K-Bloom$_{RoBERTa-large}$ (Ours)	84.19 (± 0.72)	15.81 (± 0.72)

Finally, the results in Figure 5.6 underscore the improved stability and precision of our method. The red lines representing our models consistently lie above the blue lines of BertNet, indicating superior performance. These findings highlight the effectiveness of our method in refining the performance of PLMs, particularly in terms of precision in tuple extraction tasks. Using initial prompts, our approach performs better than BertNet in terms of precision and consistency. This capability is evident in the higher accuracy achieved by our models compared to the baseline in multiple experiments. The improvement in performance underscores the robustness of our approach in handling complex language structures and varying contexts, making it a more reliable approach for knowledge extraction tasks.

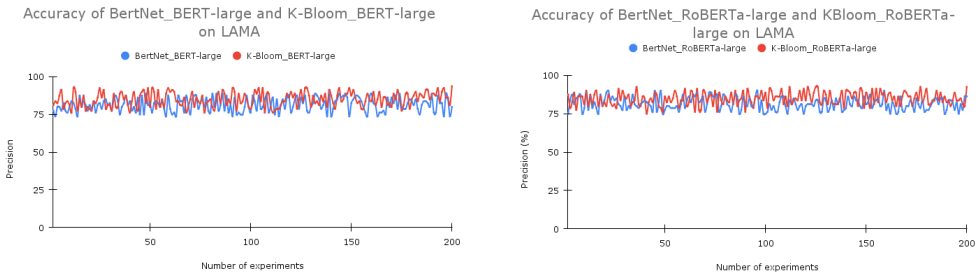


Figure 5.6: Knowledge extraction accuracy of our approach and BERTNET on LAMA using an initial prompt setting, with using BERT-LARGE and ROBERTA-LARGE as the PLMs. The left image shows the precision of both methods with BERT-LARGE, while the right image displays the precision of both methods with ROBERTA-LARGE.

5.4.3.2 Evaluation of output KGs with top-1 prompt generated

In this section, we present the value of the output tuples derived from the top-1 prompt approach, which encapsulates the most relevant and contextually appropriate information. The new prompt paraphrasing stage formulates contextually relevant queries that are then used to interact with PLMs. The top-1 prompt, generated in (§3.2) and identified by its highest weight, is used to extract knowledge tuples of a specific relation from PLMs. This prompt selection ensures that the most effective and relevant prompt is utilized for knowledge extraction.

The results of the top-1 prompt approach, as illustrated in Table 5.1, show a significant improvement in diversity and extracted fact count metrics with our method compared to BERTNET in the ConceptNet dataset. In particular, our K-BLOOM$_{\text{BERT-LARGE}}$ generates 19,599 tuples, which is a considerable increase over the 13,958 tuples generated by BERTNET$_{\text{BERT-LARGE}}$. Furthermore, our K-BLOOM$_{\text{ROBERTA-LARGE}}$ diversity score of 25,890 is significantly higher than the 11,681 diversity score of BERTNET$_{\text{ROBERTA-LARGE}}$. Similarly, our method achieves a extracted fact count metric of 39,198, surpassing BERTNET$_{\text{BERT-LARGE}}$'s extracted fact count of 27,916. These results indicate that our K-BLOOM$_{\text{BERT-LARGE}}$ method significantly outperforms the BERTNET$_{\text{BERT-LARGE}}$ method in tuple generation, diversity and extracted fact count. These findings highlight the effectiveness of our method in generating richer and more diverse knowledge graphs, making a substantial contribution to the field of knowledge graph construction and its applications.

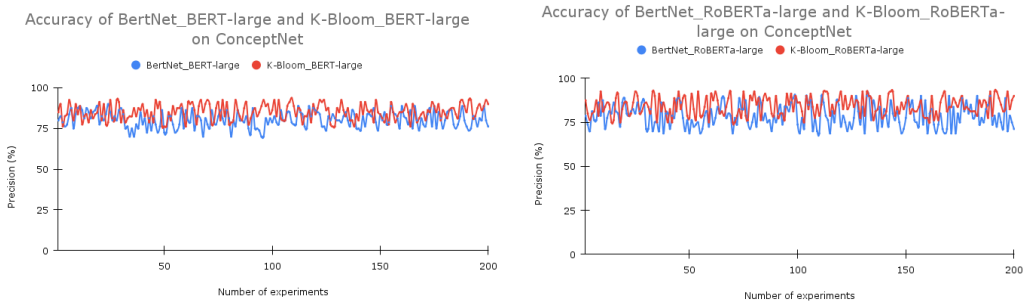


Figure 5.7: Knowledge extraction accuracy of our approach and BERTNET on ConceptNet using top-1 prompt approach, with BERT-LARGE and ROBERTA-LARGE as the PLMs. The left image shows the precision of both methods with BERT-LARGE, while the right image displays the precision of both methods with ROBERTA-LARGE.

From Table 5.2, these results demonstrate a significant improvement in the generation of reasonably large sets of knowledge using our methods. In the ConceptNet_train100k dataset, K-BLOOM$_{\text{BERT-LARGE}}$ achieves a novelty score of 89.08%, substantially higher than 63.33% achieved by BERTNET$_{\text{BERT-LARGE}}$. Similarly, K-BLOOM$_{\text{ROBERTA-LARGE}}$ outperforms BERTNET$_{\text{ROBERTA-LARGE}}$ with a novelty score of 90.88% compared to 72.01%. This trend is consistent across larger datasets, with K-BLOOM$_{\text{BERT-LARGE}}$ and K-BLOOM$_{\text{ROBERTA-LARGE}}$ achieving 85% and 87.53% novelty, respectively, on the ConceptNet_train300k dataset, compared to 54.5% and 64.24% for BERTNET$_{\text{BERT-LARGE}}$ and BERTNET$_{\text{ROBERTA-LARGE}}$. In the ConceptNet_train600k dataset, our methods continue to excel with novelty scores of 80.41% for K-BLOOM$_{\text{BERT-LARGE}}$ and 84.43% for K-BLOOM$_{\text{ROBERTA-LARGE}}$, outperforming BERTNET$_{\text{BERT-LARGE}}$ and BERTNET$_{\text{ROBERTA-LARGE}}$, which score 41.8% and 53.04%, respectively. In the ConceptNet_test dataset, K-BLOOM$_{\text{BERT-LARGE}}$ and K-BLOOM$_{\text{ROBERTA-LARGE}}$ maintain high novelty scores of 98.48% and 98.7%, respectively, compared to 93.8% for BERTNET$_{\text{BERT-LARGE}}$ and 95.47% for BERTNET$_{\text{ROBERTA-LARGE}}$. The consistent trend across these datasets underscores the robustness and effectiveness of K-BLOOM in promoting diversity within the knowledge extraction process, thereby enhancing the overall quality and applicability of the generated knowledge graphs. These findings highlight the superiority of our approach in capturing a broader spectrum of knowledge, making it a valuable tool for expanding and enriching knowledge bases with novel and diverse information.

Furthermore, our analysis with the top-1 prompt approach in the LAMA dataset, as presented in Table 5.5, demonstrates a notable per-

formance disparity between the models examined. Specifically, BERTNET$_{\text{BERT-LARGE}}$ records an accuracy of 79.26%, while our proposed model, K-BLOOM$_{\text{BERT-LARGE}}$, significantly outperforms it with an accuracy of 84.73%. A similar pattern emerges with the RoBERTa variant: BERTNET$_{\text{ROBERTA-LARGE}}$ achieves 78.75% accuracy, while our K-BLOOM$_{\text{ROBERTA-LARGE}}$ model improves this with an accuracy of 84.04%. These findings underscore the superior efficacy of our K-BLOOM approach in different model architectures (as shown in Figure 5.7), reinforcing its robustness and reliability in improving model performance. It can be seen that our K-BLOOM method significantly enhances the generation of new knowledge, outperforming BERTNET in different dataset sizes. This improvement underscores the efficacy of our method in utilizing the top-1 prompt generated, which has the highest confidence weight, to extract unique and previously unseen knowledge from PLMs.

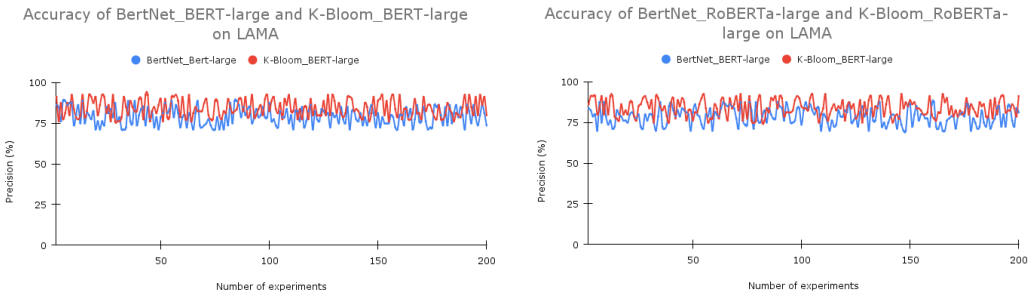


Figure 5.8: Knowledge extraction accuracy of our approach and baseline on LAMA using top-1 prompt approach, with BERT-LARGE and ROBERTA-LARGE as the PLMs. The left image shows the precision of both methods with BERT-LARGE, while the right image displays the precision of both methods with ROBERTA-LARGE.

Finally, the experimental results on the LAMA dataset indicate that our K-BLOOM enhances the performance of PLMs in knowledge extraction tasks compared to BERTNET (as illustrated in Figure 5.8). This suggests that our top-1 prompt effectively guides the PLMs towards relevant and factual entity relationships, contributing to the high precision of extracted pairs. In other words, our prompt paraphrasing method generates high-quality prompts that closely align with the semantic content of the initial prompt. From the experiments with the top-1 prompt generated, it is evident that our prompt paraphrasing strategy, focusing on the most reliable output from the initial part of our framework, significantly contributes to the robustness and creativity of the results. Furthermore, our top-1 prompt approach achieves

a 0.19% increase in accuracy compared to the initial prompt approach (according to Table 5.5), highlighting the efficiency of our prompt search algorithm in generating high-quality prompts. This improvement suggests that our top-1 prompt effectively guides PLMs toward identifying relevant and factual entity relationships, thereby contributing to the high precision of the extracted pairs. In other words, our prompt paraphrasing method generates the top-1 high-quality prompt that is semantically aligned with the initial prompt, ensuring that the extracted information is more precise and reliable.

5.4.3.3 Evaluation of output KGs with multiple generated prompts

To further assess the effectiveness of our complete framework, we evaluated a variant that uses all diverse automatic generated prompts to extract knowledge from the PLMs. Specifically, we utilize all the generated prompts from the New Prompt Paraphrasing stage (§3.2) to exploit the internal knowledge of PLMs.

From Table 5.1, it is evident that our multi-prompts approach in the K-Bloom model generates 26,461 tuples compared to BertNet's 10,824. This improvement in diversity indicates that our new prompt paraphrasing module is more effective in generating a wide range of unique entity pairs, thereby enriching the knowledge graph with varied and comprehensive information. Next, Table 5.2 shows that our K-Bloom method consistently exceeds the baseline BertNet method in all sizes of ConceptNet training datasets and the test set. For example, on the ConceptNet_train100k dataset, K-Bloom achieves an impressive novelty of 92.03%, while BertNet only manages 72.96%. This significant performance gap highlights the superior capability of our approach in introducing novel and valuable insights into the knowledge graph. Such advancements are critical to improve the comprehensiveness and practical utility of the extracted knowledge, demonstrating the effectiveness of K-Bloom in increasing the quality and depth of the knowledge graph construction.

In addition, Table 5.3 and Figure 5.9 clearly demonstrate the superior performance of our K-Bloom method in terms of accuracy in all experimental settings. In particular, under the multi-prompts approach, our method achieves an accuracy of 86.11%, significantly surpassing BertNet, which records an accuracy of 82.36%. These results underscore the robustness and effectiveness of our prompt generation and entity extraction framework, which strategically leverages the strengths of the BERT and RoBERTa models. Moreover, our approach successfully captures a more diverse set of relationships between entities by generating a broader spectrum of prompts,

leading to a more comprehensive and precise extraction of extracted tuples. Our multi-prompts strategy enhances accuracy by an additional 0.97% compared to our top-1 prompt, demonstrating that integrating diverse prompts more effectively captures the semantics of relations. By generating a broader spectrum of prompts, our approach successfully captures a more diverse set of relationships between entities, leading to a more comprehensive and precise extraction of new tuples. The improvements in precision indicate that our prompts not only generate new information, but do so with a high degree of accuracy. This increased diversity ensures that the knowledge graph remains rich and comprehensive, while the high novelty scores affirm our method's ability to introduce relevant new information.

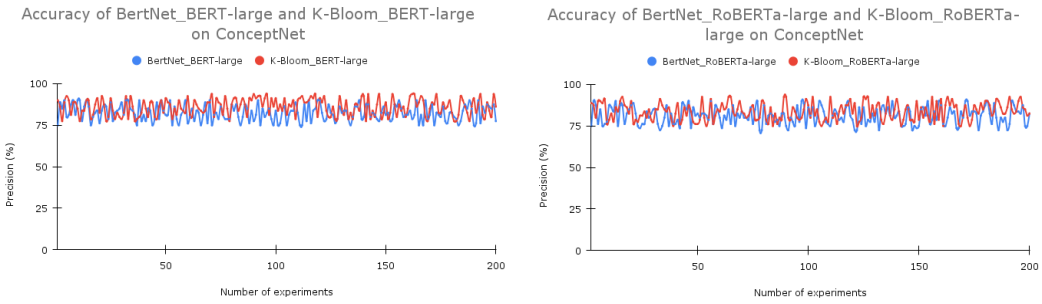


Figure 5.9: Knowledge extraction accuracy of our approach and baseline on ConceptNet using multi-prompts setting, with BERT-LARGE and ROBERTA-LARGE as the PLMs. The left image shows the precision of both methods with BERT-LARGE, while the right image displays the precision of both methods with ROBERTA-LARGE.

In the LAMA relations, our proposed model shows a marked improvement in accuracy over baseline, as detailed in Table 5.5. Specifically, the accuracy ranges from 79.82% to 84.8% with BERT-large and from 79.4% to 84.19% with RoBERTa-large, reflecting a lower incidence of false positive connections. This suggests that our model effectively filters out "counterfeit" associations, leading to a more reliable and trustworthy knowledge graph. The detailed insights from Figures 5.9 and 5.10 further substantiate the superior performance of K-BLOOM in extracting rich, precise, and diverse knowledge. These results highlight the robustness and adaptability of the K-BLOOM framework, underscoring its potential to significantly improve the quality and diversity of knowledge graphs. This advancement emphasizes the method's potential to elevate the quality and diversity of knowledge graphs, thereby making a significant contribution to the field of knowledge extraction

from PLMs.

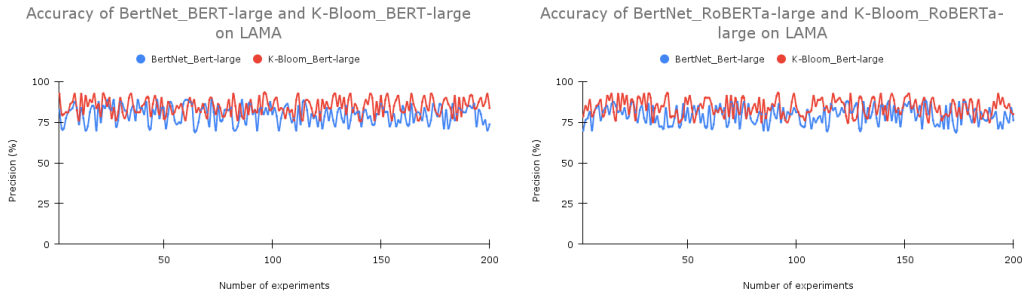


Figure 5.10: Knowledge extraction accuracy of our approach and baseline on LAMA using multi-prompts approach, with BERT-LARGE and ROBERTA-LARGE as the PLMs. The left image shows the precision of both methods with BERT-LARGE, while the right image displays the precision of both methods with ROBERTA-LARGE.

Our prompt generation framework produces fewer diverse prompts than the baseline. However, the prompts generated by our framework are semantically superior to those of BERTNET, since our method selects synonyms of keywords in the initial prompt to create diverse and meaningful prompts. The results indicate that the scoring functions in our entity tuple search stage capture more accurate and reasonable tuples compared to BERTNET. Additionally, our prompt paraphrasing stage, combined with our new entity pair searching method, proves effective in capturing relation semantics. These experimental results demonstrate the adaptability of K-BLOOM, which is plug-and-play and can be implemented in various scenarios. In general, our framework leverages the diverse perspectives offered by multiple prompts to unlock a wealth of accurate and novel entity pairs within PLMs.

5.4.3.4 Evaluation of output KGs with ConceptNet knowledge base completion data

To validate the accuracy of the knowledge graph extracted from PLMs in our unsupervised framework, we use the ConceptNet knowledge base completion task as a surrogate ground truth. Specifically, we use a training set of 600,000 tuples from ConceptNet as a proxy for the ground truth. The validation process involves comparing the tuples generated by our framework and the BERTNET framework with the ground-truth tuples. This comparison is conducted 200 times, where, for each iteration, we randomly select 100 tuples of the same relation from the ground truth. The semantic similarity between

each generated tuple and the 100 ground-truth tuples was calculated. If the similarity value exceeds a threshold of 0.75, the number of accuracies will increase by one.

A tuple is considered valid if it exhibited semantic similarity to more than half of the ground-truth tuples, which means it is deemed valid in at least 50 out of 100 randomly selected tuples. The precision of our approach and BertNet was then evaluated in three different prompt scenarios: initial prompt, top-1 prompt, and multi-prompts generation.

Table 5.6: Accuracy results (%) of output KG from BertNet and our K-Bloom method on ConceptNet_train600k dataset.

Method	**Accuracy (%)**
Initial Prompt Approach	
$\text{BertNet}_{\text{RoBERTa-large}}$	35.76 (± 0.66)
$K - Bloom_{RoBERTa-large}$ (Ours)	**43.46** (± 0.85)
Top-1 Prompt Approach	
$\text{BertNet}_{\text{RoBERTa-large}}$	37.90 (± 0.59)
$K - Bloom_{RoBERTa-large}$ (Ours)	**47.08** (± 0.95)
Multi-Prompts Approach	
$\text{BertNet}_{\text{RoBERTa-large}}$	40.28 (± 0.70)
$K - Bloom_{RoBERTa-large}$ (Ours)	**50.95** (± 0.86)

Our approach surpasses BertNet in accuracy in various scenarios, as illustrated in Table 5.6. Specifically, for the init prompt setting, BertNet records an accuracy of 35.76%, while our method achieves a considerably higher accuracy of 43.46%. In the top-1 prompt scenario, our approach further improves performance, reaching an accuracy of 47.08%, compared to BertNet's precision of 37.9%. The most significant enhancement is observed in the multi-prompts scenario, where our framework attains an accuracy of 50.95%, outperforming BertNet, which manages 40.28%. These results consistently affirm the superior precision and efficacy of our method in generating semantically relevant tuples, validating its advantage over BertNet in all scenarios tested. By leveraging LLMs for knowledge graph extraction and using multiple prompts, our framework demonstrates a robust capability to capture accurate and relevant knowledge, reinforcing the

value of our approach to produce semantically precise and reliable knowledge graphs.

In general, K-BLOOM consistently outperforms BERTNET in various prompt settings, indicating that our approach produces a more precise and reliable knowledge graph, which is essential for tasks such as answering questions and fact-checking. This gain supports our novelty claim: soft OT matching better preserves contextual consistency than hard alignment, leading to higher-quality extracted tuples.

5.4.3.5 Precision evaluation of novel-only tuples not produced by BERTNET

To directly validate the quality of our *novel* extracted knowledge, we evaluate the precision of novel-only tuples produced by K-BLOOM that do not appear in BERTNET: $T_{\text{novel}} = T_{\text{K-Bloom}} \setminus T_{\text{BERTNET}}$. We use ConceptNet_train600k (600,000 tuples) as a ground truth.

We follow the same validation protocol as Section 5.4.3.4 and repeat the evaluation 200 times. In each iteration, we randomly sample 100 ground-truth tuples of the same relation and compute the semantic similarity between one generated tuple (sampled from T_{novel}) and each of the 100 ground-truth tuples. If the maximum similarity exceeds a threshold $\tau = 0.75$, the tuple is counted as *correct* (i.e., the score increases by one). We report the mean ($\pm$ std) accuracy across 200 iterations under three prompt scenarios: initial prompt, top-1 prompt, and multi-prompts generation.

Table 5.7: Novel-only precision (%) of K-BLOOM on ConceptNet_train600k, evaluated on tuples in $T_{\text{novel}} = T_{\text{K-Bloom}} \setminus T_{\text{BERTNET}}$.

Setting	**Novel-only Precision (%)**
Initial Prompt	37.41 ($\pm$ 0.77)
Top-1 Prompt	42.09 ($\pm$ 0.86)
Multi-Prompts	44.82 ($\pm$ 0.81)

Table 5.7 reports the precision of *novel-only* tuples extracted by K-BLOOM, i.e., tuples in $T_{\text{novel}} = T_{\text{K-Bloom}} \setminus T_{\text{BERTNET}}$ that are not produced by the baseline. When evaluation is restricted to novel-only tuples that BERTNET does not produce, K-BLOOM still achieves stable precision, improving from 37.41% (Initial) to 42.09% (Top-1) and reaching 44.82% under Multi-Prompts. This consistent upward trend indicates that our prompt selection and multi-prompt aggregation not only increase tuple quantity but also improve the correctness of genuinely new extractions. Importantly, these

results directly support our novelty claim: K-Bloom is able to produce additional knowledge beyond BertNet while keeping a reasonable precision level, with multi-prompt generation yielding the most reliable novel tuples.

5.4.3.6 Human–AI inter-annotator agreement

The inter-annotator agreement results show that the Human and AI judges are moderately consistent beyond chance when assessing whether K-Bloom triples are correct. Across two independent runs of 100 randomly sampled triples, exact agreement is 77–80%, and Cohen's κ is 0.481–0.502. Following Landis & Koch [167], κ values between 0.41 and 0.60 correspond to *moderate agreement*, indicating that the observed alignment is not solely explained by chance agreement induced by similar label prevalences. The close κ values across the two runs (0.481 vs. 0.502) further suggest that the agreement level is reasonably consistent across different random samples.

This directly supports K-Bloom's contribution: the method aims to make implicit PLM knowledge explicit as reusable KG triples, and these results show that the extracted triples are verifiable under independent judging rather than relying on a single evaluator. In other words, K-Bloom does not only increase tuple quantity; it produces extracted knowledge that can be validated with moderate, reproducible agreement, strengthening the *evidence-based* and *inspectable* design goal of the framework.

Table 5.8: Unweighted Cohen's kappa across Human–AI triple correctness labels (N=100).

Settings	**Cohen's κ**	**Exact Percent**
Human–AI (Run-1)	0.481	80.0
Human–AI (Run-2)	0.502	77.0

5.4.4 Ablation Study

In this study, we perform an ablation analysis to assess the impact of our scoring function, which addresses the limitations of the traditional Word Mover Distance by replacing the Euclidean distance metric with cosine similarity to account for angular dissimilarity in embedding vectors. WMD methods using Euclidean distance tend to overlook the angular dissimilarity of embedding vectors, potentially leading to less accurate semantic comparisons. In contrast, our approach utilizes cosine similarity to measure angular similarity between embedding vectors, thus offering a more precise metric for evaluating semantic relationships.

To assess the effectiveness of our scoring function, we systematically replace the semantic scoring function of our architecture with the conventional Euclidean distance metric in selecting the most semantically reasonable tuples generated by pre-trained language models. The evaluation is conducted on the ConceptNet_train600k dataset, and Table 5.9 shows the results of an ablation experiment recorded in accuracy, novelty, and diversity.

Table 5.9: An ablation study comparing the output knowledge graphs generated from pre-trained language models using our proposed semantic scoring function versus the conventional Euclidean metric on ConceptNet_train600k dataset

Our semantic scoring	Euclidean metric	**Accuracy (%)**	**Novelty**	**Extracted Fact Count**	**Diversity (%)**
		Initial-Prompts Approach			
✓	✗	**43.46** (± 0.85)	26,252	39,424	**86.02**
✗	✓	37.5 (± 0.73)	26,252	39,424	82.15
		Multi-Prompts Approach			
✓	✗	**50.95** (± 0.88)	26,461	40,000	**85.45**
✗	✓	41.72 (± 0.75)	26,461	40,000	81.83

The results, detailed in Table 5.9, demonstrate that our method outperforms the Euclidean metric using the initial prompt and multi-prompts approaches. The ablation study highlights that our approach significantly outperforms the traditional method in terms of accuracy, diversity, and overall accuracy. Specifically, our semantic scoring function demonstrates superior precision and diversity, achieving a precision of 43.46% and a diversity score of 86.02%, compared to an accuracy of 37.5% and a diversity score of 82.15% when using the Euclidean metric. Similarly, under the multi-Prompts approach, our semantic scoring attains an accuracy of 50.95% and a diversity of 85.45%, outperforming the Euclidean metric's 41.72% accuracy and 81.83% diversity. Notably, while the novelty and extracted fact count remain unchanged between the two scoring methods, the enhanced precision and diversity achieved by our approach highlight its robustness in generating semantically rich and accurate knowledge representations. These results emphasize the superiority of cosine similarity over Euclidean distance in our semantic scoring function, particularly in capturing nuanced, semantically rich relationships, thereby contributing to more accurate and diverse knowledge graph construction from PLMs.

5.4.5 Error Analysis

In Table 5.10, we show an example of tuples extracted from PLMs by our proposed method and the approach of BertNet under two settings of

prompts: an initial prompt and multiple generated prompts. We can easily see the benefits of our proposed method over BERTNET from these results.

Table 5.10: Error analysis of top 10 generated tuples of relation *UsedFor* from BertNet and our $K - Bloom$ method on ConceptNet dataset.

Our generated tuples	**Generated tuples of BertNet**
Initial Prompt Approach	
(“sanskrit grammar", “karnataka")	(“solar radiation", “solar panels")
(“sanskrit",“karnataka")	(“solar cells", “solar panels")
(“coffee maker",“brewing coffee")	(“de facto", “temporary status")
(“thermometer", “measuring temperature")	(“mechanical vibration", “mechanical engineering")
(“parliament", “government")	(“natural turf", “soccer")
(“russian script",“cyrillic")	(“natural turf", “grass")
(“europe",“greek macedonia")	(“dry rot", “treatment")
(“sanskrit words",“kerala")	(“legend", “historical significance")
(“english translations",“english")	(“de facto", “temporary situations")
(“clock",“waking up")	(“heavy fog", “visibility")
Multi-Prompt Approach	
(“october bench",“sit upright")	(“ladder",“climbing ladder")
(“christmas distribution",“gift")	(“ladder",“climbing stairs")
(“probability",“statistical model")	(“window",“open window")
(“piano",“playing musical")	(“television",“broadcast television")
(“drug therapy",“treatment")	(“electric stove",“heat")
(“car",“transportation")	(“camera",“video surveillance")
(“august week",“vacation break")	(“window",“open windows")
(“oven",“heating food")	(“pulpit”,“preaching sermons”)
(“calendar",“tracking time")	(“beach",“surf surfing")
(“august week",“holiday")	(“pulpit",“preaching sermon")

Our proposed method demonstrated a significant improvement in accuracy compared to BERTNET when evaluating the factual consistency of newly extracted tuples using Gemini. In the case of using an initial prompt, while both methods exhibited some errors which are highlighted by red color (our method: 3/10, baseline: 5/10), ours achieved a higher success rate (7/10) compared to the baseline (5/10). Similarly, the results reported in Table 5.10 using the multiple prompt approach show that our method performs significantly better with a multiple prompt approach (8 successes out of 10) compared to BERTNET's 6 successes out of 10. Our Multiple-Prompt approach achieves an accuracy improvement of 2 tuples compared to the baseline. This shows that combining multiple prompts is more effective in capturing the semantics of a relation.

This improvement can be attributed to the effectiveness of our method in capturing the underlying relationships between entities. However, it is important to acknowledge that both methods still produced errors, such as the misinterpretation of context, leading to inaccuracies in tuple extraction in the output KG of our method (“sanskrit words, kerala"), and the presence of irrelevant information in the baseline (“de facto, temporary status" and “de facto, temporary situations"). Despite these errors, the notable improvement in accuracy achieved by our proposed method underscores its efficacy and

importance in enhancing tuple extraction from PLM, signifying a significant advancement in the field.

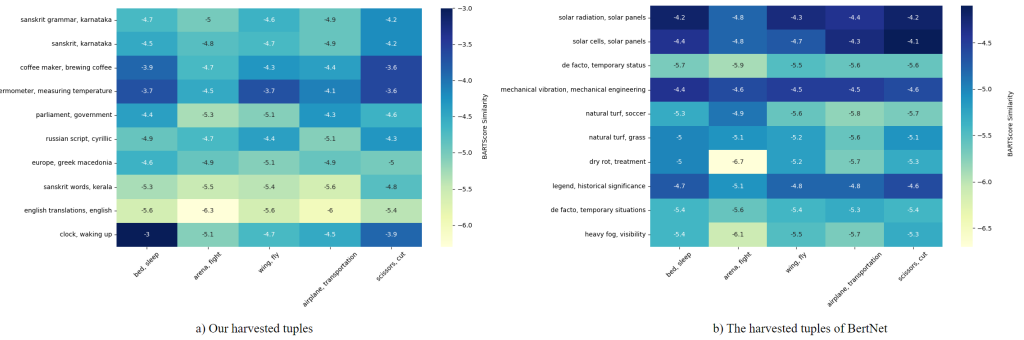


Figure 5.11: The correlation matrices between the generated tuples of two methods: (a) our harvest tuples and (b) the generated tuples of BertNet. The x-axis represents seed entity pairs, the y-axis symbolizes new entity pairs.

Moreover, in our comprehensive error analysis, we use the BARTScore [168] to evaluate the quality and relevance of the generated entity pairs from our proposed approach and the baseline BertNet. As shown in Figure 5.11, our approach achieves an average BART score of -4.69, ranging from -2.97 to -6.27. This is an improvement over BertNet's average score of -5.12, which ranges from -4.15 to -6.70. Notably, our K-Bloom produces fewer extremely low-scoring tuples, with a minimum score of -6.3 compared to -6.7 for BertNet, and significantly better high-scoring tuples, with a maximum score of -3.0 versus -4.1 for BertNet. These findings indicate that our framework consistently enhances the overall quality of the pairs of extracted entities. Furthermore, our framework obtains higher average BART scores for six out of the ten pairs of extracted entities. This suggests that our approach is more effective in identifying semantically relevant entity pairs for arbitrary relation types compared to BertNet, thereby demonstrating its robustness and reliability in knowledge extraction tasks.

5.5 Conclusion

In this section, we present a pipeline for extracting knowledge graphs from pre-trained language models (e.g., Bert, RoBERTa), using minimal input, defined as a few seed entity pairs and an initial prompt. By relying solely on language models as the source of knowledge, without the need for external

training data, our method offers significant flexibility, enabling the dynamic integration of arbitrary pre-defined relations directly into the model. A key aspect of our approach is the introduction of scoring functions for all possible entity pairs extracted from these models. In contrast to traditional methods that require extensive annotation and domain-specific expertise, our framework reveals hidden knowledge within pre-trained LMs with minimal input. This reduces both the complexity and the cost of constructing knowledge graphs, making the process more accessible to a wider range of applications. Our proposed framework demonstrates significant advancements over BertNet by using scoring functions to refine the selection process, ultimately leading to a richer and more accurate extraction of knowledge from pre-trained language models. In this paper, we first indicate the limitation of Euclidean transportation cost in the Word Mover's Distance, which misestimates the similarity of word pairs. Based on this finding, we propose a novel unsupervised distance measure that serves as a better proxy for word similarity.

Experimental results on the ConceptNet and LAMA benchmark datasets show that our framework can extract more meaningful tuples in terms of semantics and diversity for each relation, resulting in a more comprehensive knowledge graph. This work establishes a powerful framework for constructing KGs from PLMs, fueled by seed entity pairs. This approach facilitates efficient knowledge discovery, improves model interpretability, and improves the performance of various NLP tasks, including question answering, information retrieval, and knowledge-based reasoning. These advances are also particularly beneficial in specialized domains such as biomedical and legal fields, where accurate and diverse knowledge representation is critical. However, while these results are promising, they highlight an important dependency on the quality of the initial seed triples, which poses a critical challenge in ensuring the reliability of the constructed KG. Overall, K-Bloom's novelty is an OT-based consistency criterion that turns noisy PLM generations into a reusable KG with verifiable tuple selection.

One of the biggest challenges in harvesting knowledge graphs from pre-trained language models is their heavy reliance on the initial seed triples, which consist of relationship descriptions and entities. These seeds serve as the foundation, guiding the model in generating the rest of the graph. However, if the seed data contains biases, inaccuracies, or lacks diversity, these flaws can propagate and even amplify in the harvested KG, leading to distorted or incomplete representations of the knowledge domain. Consequently, the reliability and utility of the extracted KG are largely determined by the quality of these starting points. Addressing this limitation requires rigorous curation and validation of seed data, with a focus on ensuring di-

versity and minimizing bias to enhance the accuracy and comprehensiveness of the resulting KGs.

Our current framework and experimental results indicate that while our approach has improved the variety of generated tuples, some extracted entity pairs may not be perfectly accurate and could contain misleading facts and disinformation. As a remedy, we propose a multi-pronged approach. Fine-tuning PLMs on task-specific datasets allows for refining general models to specialized knowledge domains, mitigating ambiguities and enhancing precision. Moreover, error detection and correction mechanisms play a critical role in identifying and addressing inaccuracies. Techniques such as meta-learning help models recognize uncertainty or erroneous outputs while self-ensembling and out-of-domain detection further flag potential errors. In addition, we are investigating the use of open-source large language models that can be deployed locally for harvesting the internal knowledge embedded within these models, such as LLaMA [169] or Mistral [170]. Future work should focus on developing advanced techniques to mitigate hallucination errors, ensuring that the generated tuples maintain semantic consistency and relevance to the target domain. Finally, our proposed "extracted fact count" metric might not be optimal, as measuring knowledge coverage itself is inherently challenging. In the future, we aim to develop advanced techniques to address this limitation. Methods such as entity matching and contextual validation could provide more precise evaluations. Additionally, leveraging external knowledge bases for alignment and fine-tuning PLMs with objectives tailored to extracted fact count optimization may significantly enhance extraction performance. These approaches hold promise for constructing more comprehensive and accurate knowledge graphs.

Chapter Summary

This chapter introduces **K-Bloom**, our unsupervised knowledge extraction module designed to make latent knowledge in pretrained language models explicit and reusable as a knowledge graph. We begin with the observation that many downstream reasoning tasks—especially in biomedicine—benefit from structured relational facts; yet, most model knowledge remains implicit and cannot be directly inspected or verified. To address this, K-Bloom extracts candidate relational triples and scores their consistency with context using an **Optimal Transport** formulation with a similarity-based transport cost. This Optimal Transport-based scoring acts as a principled filter, reducing noisy or inconsistent triples and yielding higher-precision extractions than naive pattern- or prompt-based generation. The extracted

triples are then organized into a lightweight knowledge graph resource that can be reused across tasks without task-specific labels or fine-tuning. We also discuss remaining challenges of unsupervised extraction, including coverage limitations and maintaining precision when scaling to broader domains.

This chapter establishes the first **core principle** of the dissertation: **unsupervised knowledge extraction**. The resulting knowledge graph is not only an end product for analysis but also a reusable resource that supports Chapters 6 (UGAT-MedQA) and 7 (USCRaKe). In particular, the graph serves as an external knowledge source for biomedical question answering. By transforming implicit model knowledge into an explicit and verifiable format, K-Bloom directly advances the dissertation goal of enabling evidence-based reasoning in biomedical settings.

At the same time, this chapter clarifies a practical limitation of using **cosine similarity** as a semantic signal. Cosine is simple and fast, but it compares only the *direction* of vectors, which can miss how information is distributed inside a representation; two sentences may align in direction while differing in their internal feature makeup. Cosine also does not directly compare probability distributions of features, which becomes important when semantic evidence is expressed as distributional differences rather than a single vector direction. These limitations motivate the next chapter, **USCRaKe**, which extends the same Optimal Transport foundation from knowledge extraction to text retrieval: instead of relying on cosine similarity, USCRaKe compares contextual token distributions via Optimal Transport with **Jensen–Shannon divergence** as the ground cost to improve unsupervised evidence selection.

Chapter 6

UGAT-MedQA: Unsupervised Graph Attention Network Empowered by LLMs for Medical Question Answering

6.1 Introduction

Biomedical question answering (BQA) is increasingly vital for clinical decision support, medical education, and patient-facing health information systems, where accuracy, interpretability, and reliability are critical. Effective QA systems depend on robust access to relevant knowledge and strong reasoning capabilities to process it. Modern approaches typically leverage two primary sources: knowledge implicitly encoded in large language models (LLMs) [3, 4, 21, 51, 169, 171–173], and knowledge explicitly structured in knowledge graphs (KGs) such as Freebase [136], ConceptNet [163], and Wikidata [164]. While LLMs achieve strong results on diverse natural language processing (NLP) tasks, including QA, they have important limitations: reliance on static, pre-trained knowledge; difficulty with specialized, up-to-date, or multi-hop information needs; and a lack of transparency and interpretability [174–176]. Updating LLMs is costly and cannot guarantee knowledge freshness, while answers may be prone to hallucination or lack clear evidence. Conversely, KGs organize curated factual knowledge as explicit triples (*head, relation, tail*), such as `<Australia` → `language_spoken` → `English>`, supporting structured and interpretable reasoning [177–179]. However, their coverage is inherently limited; KGs may be noisy [180, 181], and they work best for questions that can be directly mapped to the graph, limiting open-domain applicability [182, 183].

Retrieval-augmented generation (RAG) [9] provides a promising bridge by enriching LLM input with external knowledge, e.g., from KGs. This integration leverages a wide range of relevant data sources, including patient

records and medication lists, to enhance the model's decision-making capabilities. However, standard RAG often incorporates irrelevant or noisy content [184], and subgraph expansion methods tend to include many semantically unrelated nodes [81, 179], diluting answer evidence and reducing reasoning quality. Most frameworks treat the question and KG as separate modalities, making it difficult to accurately identify the most relevant nodes for a given query [185, 186].

Despite these advances, several challenges remain unsolved. First, naive RAG faces challenges in generating novel insights and often falls short in tasks that demand a comprehensive understanding of large document collections. Second, KGs contain millions of facts, making it difficult to isolate the evidence most relevant to a given question; irrelevant retrieval not only introduces noise but also confuses downstream reasoning in LLMs. Existing approaches [47–50] either rely on generic natural language processing (NLP) retrievers or heuristic graph traversal, which are not tailored for KG reasoning or require costly LLM-driven traversal that is impractical in production cases. Third, heuristic or supervised approaches to node selection in KGs require either hand-crafted rules or labeled training data, both of which limit scalability and generalizability across biomedical domains. Estimating node relevance in an unsupervised manner is particularly challenging, as it requires capturing subtle interactions between the question and graph nodes without explicit supervision.

A natural question is why we adopt a Graph Attention Network (GAT), given that GAT has been widely used in graph-based QA. Our key insight is that node selection in large biomedical knowledge graphs is fundamentally a neighbor-weighting problem: for a given question, only a small subset of connected entities provides useful evidence, while many neighbors act as noise or distractors. In standard GCN-style message passing, neighbors are aggregated with structure-based normalization (e.g., degree-based scaling) that is typically independent of the question, which can dilute relevant signals in noisy neighborhoods. In contrast, GAT computes adaptive, edge-level attention weights, allowing the model to emphasize informative neighbors and suppress distractors. In UGAT-MedQA, we further couple attention-based aggregation with question-guided scoring so that nodes can be ranked by relevance to the question. Importantly, our use of GAT differs in its role from many existing GAT-based QA systems. Many approaches train GAT end-to-end with supervision (e.g., answer labels or reasoning annotations) to directly predict answers or rank reasoning paths. In UGAT-MedQA, we instead use GAT in an unsupervised manner to support evidence retrieval: it produces node representations that are then scored by alignment to the question embedding (e.g., cosine similarity), without requiring any node-level

relevance labels. This allows us to prune the KG into a compact, question-specific subgraph *before* extracting reasoning paths, yielding more focused and interpretable evidence chains. This design is particularly valuable in retrieval-augmented generation, where evidence transparency and traceability are as important as answer accuracy.

These challenges underscore the need for a modular, unsupervised, and interpretable framework that can facilitate robust biomedical reasoning. Building upon these considerations, we propose **UGAT-MedQA**, a novel retrieval-augmented framework that leverages an unsupervised Graph Attention Network (GAT) to dynamically quantify the contextual relevance of each node to a given question for biomedical QA. Unlike prior methods that rely on heuristic graph traversal or supervised node classification, our unsupervised attention-based scoring eliminates the need for labeled training data and reduces reliance on hand-crafted traversal rules, facilitating easier adaptation to new domains. By using our GAT-based node scoring, our method identifies and retains only the most informative entities, yielding compact and contextually faithful subgraphs. This enables explicit, interpretable multi-hop reasoning chains, which are verbalized as natural language evidence for retrieval-augmented answer generation. UGAT-MedQA is fully modular and portable: each component—including entity linking, subgraph retrieval, GAT-based node scoring, reasoning path extraction, and knowledge verbalization—is an independent, clearly defined module. This modularity enables component-level upgrades or replacements without affecting the rest of the system, supporting scalability across domains and facilitating integration with emerging LLMs or specialized biomedical knowledge graphs with minimal changes. Each stage is assigned a specific function and can be optimized independently, with scope for potential future optimization. This architectural clarity makes the framework both theoretically sound and implementable. Furthermore, analysis of GAT attention weights can provide indications of which biomedical entities and relations contribute most to the reasoning process. To the best of our knowledge, UGAT-MedQA is the first framework that integrates unsupervised graph attention mechanisms into retrieval-augmented biomedical QA, enabling node relevance estimation without labeled supervision and yielding interpretable reasoning chains. Experiments on MedQA-USMLE, MedMCQA, and MMLU-Med benchmarks demonstrate that UGAT-MedQA achieves accuracies of 86.71%, 77.50%, and 91.02%, respectively, surpassing state-of-the-art methods by up to 2.8 points. In summary, UGAT-MedQA performs label-free, question-conditioned node selection via multi-layer graph attention, producing a compact biomedical reasoning path before verbalization and answer generation.

The main contributions of this chapter are as follows:

- We propose UGAT-MedQA, a novel retrieval-augmented framework that applies an unsupervised graph attention network to select relevant entities in knowledge graphs, thereby improving reasoning for biomedical multiple-choice question answering (MCQA).
- Our approach tightly integrates GAT-based node scoring with retrieval-augmented answer generation, constructing compact, stepwise reasoning chains that are directly explainable.
- UGAT-MedQA is modular and plug-and-play: each stage is independently replaceable and easily adaptable to diverse KGs and LLMs.
- Extensive experiments on MedQA-USMLE, MedMCQA, and MMLU-Med demonstrate state-of-the-art results, with ablation studies confirming the critical impact of GAT-based node selection.

6.2 Related Works

6.2.1 Question answering over knowledge graph

Question answering over knowledge graph aims to retrieve and apply relevant facts from the knowledge graph to answer natural language questions. Given a natural language question q, a list of options $\mathcal{A}_q$, and a KG $\mathcal{G}$, the task aims to design a function f to reason answers $a \in \mathcal{A}_q$ based on knowledge from $\mathcal{G}$, i.e., $a = f(q, \mathcal{G})$. Recent advancements in question answering over a knowledge graph can be broadly categorized into two paradigms: semantic parsing-based and retrieval-based approaches. Semantic parsing (SP)-based methods utilize LLMs to convert natural language queries into structured logical forms such as S-expressions or SPARQL queries, which can be directly executed on a knowledge graph to retrieve precise answers [91], [92], [93]. These methods exploit the structured nature of KGs to enable interpretable and deterministic reasoning. In contrast, retrieval-based approaches aim to enhance response generation by extracting relevant entities, relations, or relational paths from the knowledge graph and conditioning large language models on this contextual information [94, 95]. Recently, approaches based on LLMs have utilized the reasoning abilities of these models to generate answers in a step-by-step process without requiring additional training [50], [96], [97]. The study by He et al. [48] focuses on retrieving additional contextual information relevant to a given question and using it as extra input to enhance the performance of LLMs, aiming to improve answer accuracy. This highlights the benefit of augmenting language models with external knowledge to improve performance in specialized domains.

6.2.2 Graph-Augmented Language Models for Question Answering

Integrating language models (LMs) with graphs containing natural language information has emerged as a significant research area [98]. Existing methods generally fall into two main categories: (i) leveraging latent graph-based features—often extracted with graph neural networks—to enhance language models [99, 100], and (ii) explicitly feeding verbalized graph information as part of the language model's input [96, 101]. The first category struggles with the inherent differences between graph structures and natural language, which can lead to limited effectiveness on knowledge-intensive tasks [102]. The second category, meanwhile, often suffers from the inclusion of noisy or irrelevant information retrieved from large graphs, which can hinder the reasoning performance of language models [48, 103]. To overcome these challenges, our approach integrates GAT-based retrieval with RAG for multiple-choice question answering, yielding superior results compared to previous methods.

6.3 Preliminary

Knowledge Graphs: Knowledge graphs are structured representations of factual knowledge, encoded as a set of triples: $\mathcal{G}_K = \{(e, r, e') \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}\}$, where $\mathcal{E}$ and $\mathcal{R}$ denote the set of entities and relations, respectively. KGs capture rich semantic relationships between entities, enabling the representation of complex real-world knowledge in a machine-readable format.

Reasoning Paths: Reasoning Paths are sequences of consecutive triples of depth l in KGs: $w_z = e_0 \xrightarrow{r_1} e_1 \xrightarrow{r_2} \ldots \xrightarrow{r_l} e_l$, where $\forall(e_{i-1}, r_i, e_i) \in \mathcal{G}_K$. where $e_i \in \mathcal{E}$ represents the i-th entity and $r_i \in \mathcal{R}$ represents the i-th relation in the path. Reasoning paths reveal implicit connections between entities by chaining relations, facilitating inferential reasoning. The paths reveal the connections between knowledge that potentially facilitate reasoning. For example, the reasoning path: $w_z = \text{Davis} \xrightarrow{\text{spouse\_of}} \text{Jenifer} \xrightarrow{\text{Lives\_in}} \text{New York}$ indicates that "Davis" is the spouse of "Jenifer" and "Jenifer" lives in "New York". Therefore, "Davis" could be reasoned to live in "New York".

Graph Attention Mechanism: Graph attention mechanism enables nodes in a graph to focus on relevant edges when aggregating information, based on the similarity between node features. We define a graph $\mathcal{G}$ as a structure composed of three main components: a set of nodes $\mathcal{V}$, a collection of node features $X = (h_1, \ldots, h_{|\mathcal{V}|})$, and a list of directed edge sets $\mathcal{E} = (\mathcal{E}_1, \ldots, \mathcal{E}_K)$, where K represents the total number of edges. Each node $i \in \mathcal{V}$

has its own representation $h_i \in \mathbb{R}^{d_h}$, where d_h is the number of features in each node. Assuming that an edge connects two nodes i and j, and h_i and h_j are the features of these two nodes, the attention score e_{ij} is defined as follows:

$$e_{ij} = \frac{\left(h_i \mathbf{W}^Q\right)\left(h_j \mathbf{W}^K\right)^T}{\sqrt{d_h}} \tag{6.1}$$

For any given node i, let $\mathcal{N}_i$ represent the set of its neighboring nodes. Then, we normalize the attention coefficients of the node i by using the softmax function across all the neighbor nodes $j \in \mathcal{N}_i$ as follows:

$$\alpha_{ij} = \text{softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})} \tag{6.2}$$

The output of a single attention head, denoted as z_i, is calculated as a weighted sum of the linear transformed input elements.

$$z_i = \sum_{j \in \mathcal{N}_i} \alpha_{ij} h_j \mathbf{W}^V \tag{6.3}$$

In these equations, $\mathbf{W}^Q$, $\mathbf{W}^K$, and $\mathbf{W}^V$ are learnable parameter matrices with dimensions $\mathbb{R}^{d_h \times d_z}$, where d_h is the hidden size and d_z is the output size of a single attention head. The relationship between hidden size and the number of attention heads m is given by $d_z \times m = d_h$.

Finally, the multi-head attention result $z'_i \in \mathbb{R}^{d_h}$ is obtained by concatenating the outputs of all m individual attention heads:

$$z'_i = \Big\|_{k=1}^{m} z_i^k \tag{6.4}$$

6.4 Methodology

In this work, we propose UGAT-MedQA, a novel retrieval mechanism that integrates the reasoning capabilities of Graph Attention Networks for unsupervised node classification in a retrieval-augmented generation setting with LLMs for medical MCQA tasks. Instead of keeping all nodes from hop-based subgraph expansion, we train an unsupervised GAT to score each node by its semantic alignment to the question and retain only the top-ranked nodes. Multi-hop reasoning paths are then extracted from this pruned, question-focused subgraph. Specifically, given a medical query q and a set of candidate answers $C = \{c_1, c_2, \ldots, c_n\}$, we assume the availability of a retrieved

subgraph $\mathcal{G}_q$, which serves as a source of supplementary information potentially relevant for answering the question. Unlike conventional supervised approaches, our method does not require labeled data for training, enabling flexible adaptation to diverse medical domains. Notably, our GAT model is employed to perform unsupervised node classification over $\mathcal{G}_q$, identifying the medical concepts or evidence nodes most relevant to the query intent without requiring labeled training data. Then, the shortest paths in the KG that connect answer entities and GAT-based answers are extracted to represent interpretable reasoning paths. These paths are subsequently verbalized into natural language statements and provided as input to an LLM, enabling it to reason over both retrieved knowledge and the question context. In this setup, the GAT functions as a dense subgraph reasoner that extracts structured relational evidence, while the LLM leverages its natural language understanding capabilities to synthesize and validate answers. By integrating structured graph reasoning with the natural language understanding strengths of LLMs, our framework provides a scalable and interpretable solution for complex medical MCQA tasks, effectively leveraging external knowledge to enhance answer accuracy.

In this section, we introduce our proposed method, **UGAT-MedQA**, and outline its overall framework. Figure 6.1 presents a block diagram of UGAT-MedQA, highlighting the clear flow from concept recognition to answer generation. Each stage of the pipeline is designed as an independent module, supporting flexible adaptation and easy integration. The six core components of our approach are further detailed in Figure 6.2, which visually maps out the specific function and data flow of each stage.

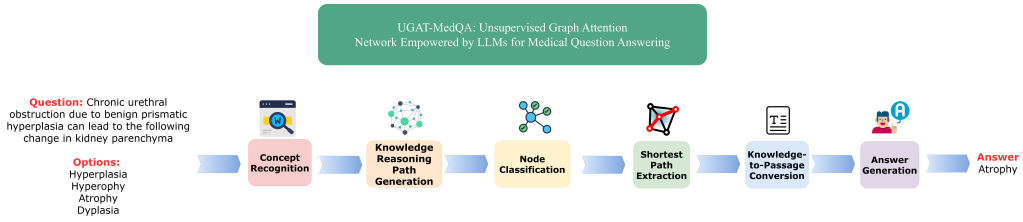


Figure 6.1: Block Diagram of our framework

- **Stage 1: Concept Recognition.** We extract medical concepts from the question using keyword extraction and hierarchical entity mapping to align them with entities in the knowledge graph.
- **Stage 2: Knowledge Reasoning Path Generation.** Based on the recognized concepts, we retrieve a question-specific subgraph from

the knowledge graph to constrain reasoning within relevant medical knowledge.

- **Stage 3: Node Classification.** We employ a graph attention network to classify nodes within the subgraph and identify clinically relevant entities for downstream reasoning.
- **Stage 4: Shortest Path Extraction.** Based on the identified clinically relevant entities from Stage 3, we extract the shortest paths in the knowledge graph that connect these entities to the answer nodes, constructing concise and interpretable reasoning chains.
- **Stage 5: Knowledge-to-Passage Conversion.** We convert the extracted reasoning paths into natural language passages to align with the input format expected by large language models.
- **Stage 6: Answer Generation.** Finally, we provide the assembled passages along with the original question to an LLM, which synthesizes the information to generate the final answer.

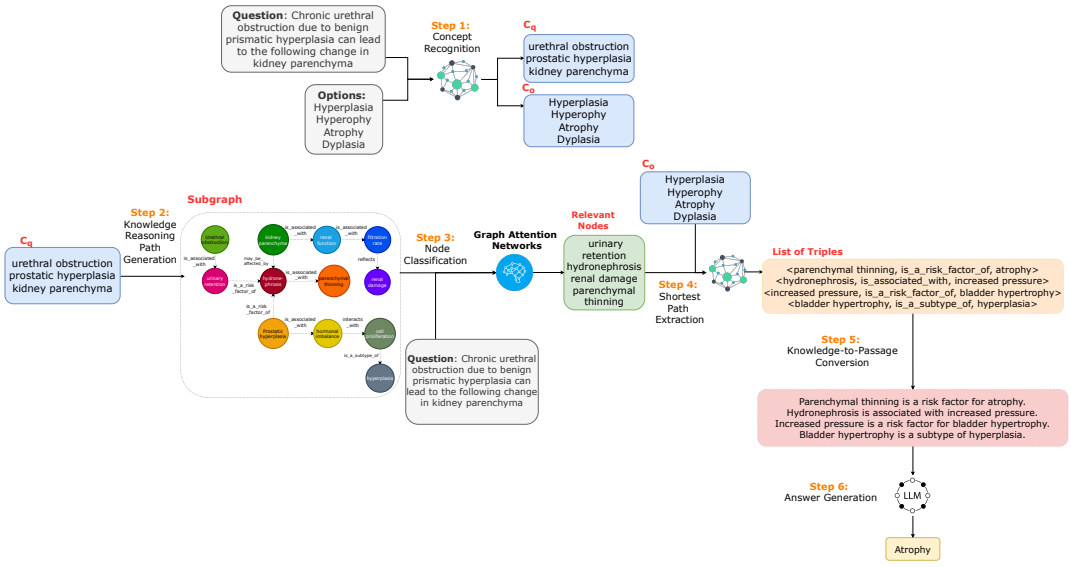


Figure 6.2: The detail of our framework. The GAT reasons over a subgraph to retrieve candidate answer nodes related to the question, and then we aim to find the corresponding reasoning paths (shortest paths from question entities to answer entities). These reasoning paths are then converted into natural language explanations and provided to the large language model (LLM) as part of the retrieval-augmented generation (RAG) process.

6.4.1 Concept Recognition

In this study, we leverage our previous work [187] in Chapter 5 to construct a self-curated biomedical knowledge graph, denoted as $\mathcal{G}_K$, that integrates both the Disease Database component of the Unified Medical Language System (UMLS) [188] and DrugBank [189]. The graph is organized as a collection of triples (h, r, t), where h and t correspond to the head and tail concepts drawn from the concept set V, and r represents a relation type selected from a predefined set R. Specifically, our KG includes 15 distinct relation types (e.g., *belongs_to_the_category_of*, *is_a_category*, ...), as shown in Table 6.1, comprising a total of 45,751 nodes and 178,147 edges.

In this stage, we first identify related medical entities, denoted as O_q^{raw}, and then we map each entity $o_i \in O_q^{\text{raw}}$ to the corresponding entity in the knowledge graph $\mathcal{G}_K$, resulting as C_q. Specifically, O_q^{raw} denotes the set of raw entity candidates extracted directly from the question text, while C_q is the set of these entities after mapping to KG entities in $\mathcal{G}_K$.

For example, given the query: `"Chronic urethral obstruction due to benign prostatic hyperplasia can lead to the following change in kidney parenchyma,"` the extracted entities O_q^{raw}, which are extracted by matching 1-grams and 2-grams from the input question, include terms such as `"Chronic, Urethral, Prostatic, following, kidney parenchyma, urethral obstruction, prostatic hyperplasia"`. However, this direct matching approach often retrieves irrelevant or ambiguous terms, including unrelated words like `"following"`. Efficient retrieval of contextually relevant knowledge from noisy resources remains an open research challenge in itself [190]. Previous studies have addressed this issue using basic techniques such as lemmatization or stop-word removal [191, 192], but these methods often lack robustness. To improve on this, we design the module **Concept Recognition** that integrates statistical, lexical, and semantic matching mechanisms to identify related medical entities C_q in $\mathcal{G}_K$. We begin by applying KeyBERT-based keyword extraction [193] to identify keywords and keyphrases within the question, followed by a hierarchical entity mapping strategy. This mapping process prioritizes exact surface-form matching and subsequently incorporates fuzzy string matching and embedding-based semantic similarity to address lexical variations and bridge gaps between the query and the entries in $\mathcal{G}_K$. This hybrid approach effectively mitigates lexical noise and enhances the recall of semantically relevant concepts, even in the presence of morphological variants or paraphrased expressions.

6.4.2 Knowledge Reasoning Path Generation

For a given set of concept entities C_q from the previous stage, we capture the induced subgraph $\mathcal{G}_q \subseteq \mathcal{G}_K$ by expanding around each entity $e \in C_q$. For each entity, we retrieve knowledge triples associated with its $D_{\max}$-hop neighbors, thereby incorporating query-relevant and faithful KG information into $\mathcal{G}_q$. The result subgraph, $\mathcal{G}_q$, is defined as $(\mathcal{E}_q, \mathcal{R}_q, \mathcal{T}_q)$, where $\mathcal{E}_q$ encompasses C_q together with the set $\{N_{\mathcal{G}_K}(e, D_{\max}) \mid e \in C_q\}$, where $N_{\mathcal{G}_K}(e, D_{\max})$ denotes the set of all entities in $\mathcal{G}_K$ reachable from e within $D_{\max}$ hops, effectively linking all relevant entities and their connective paths within the defined hop distance.

6.4.3 Node classification

In this paper, we introduce an unsupervised Graph Attention-based node classification method to identify relevant nodes within a subgraph. The algorithm utilizes a multi-head Graph Attention Network to classify nodes within a subgraph $\mathcal{G}_q$ based on their relevance to a given natural language question q. Initially, node embeddings are generated using a SentenceTransformer to capture semantic information, while the adjacency matrix encodes the graph's structure. The model incorporates K attention heads, each attending to local neighborhoods within the graph to compute node-level feature representations, $\tilde{h}_v^{(k,l)}$, for each layer l. These features are iteratively aggregated using a non-linear transformation, ψ, to produce updated node representations, $h_v^{(l)}$.

At the end of L layers, the final node embeddings, h_v^{out}, are projected into a query-aligned space through a learned linear transformation W_q. A cosine similarity-based loss function minimizes the dissimilarity between node embeddings and the query embedding during training. Nodes are classified as relevant or non-relevant by comparing their similarity scores against a predefined threshold. This process effectively integrates graph structure and semantic meaning, enabling robust relevance classification for nodes in $\mathcal{G}_q$.

After relevant nodes are identified through our unsupervised graph attention-based classification, these nodes act as key anchors for subsequent reasoning within the pipeline. Specifically, we use the relevant nodes to extract multi-hop reasoning paths that connect question concepts to potential answer candidates through medically meaningful relationships in the subgraph. These paths are then verbalized into natural language evidence passages, providing interpretable and contextually rich support for downstream answer prediction by the language model. By integrating this step, our approach ensures that only contextually important and semantically

aligned subgraphs are utilized for answer generation, which both reduces irrelevant noise and enhances the faithfulness and reliability of the medical QA system.

Learning parameters of GAT in "unsupervised" fashion: In this study, after T steps, we use the final node embedding matrix H^{out} to infer node relevance scores. We learn our model parameters (including attention weights and node embeddings) by minimizing the sampled cosine similarity loss function, applied to each node $v \in \mathcal{V}_q$, as follows:

$$\mathcal{L}_{\text{GAT}}(v) = 1 - \frac{h_v^{\text{proj}} \cdot h_q}{\|h_v^{\text{proj}}\| \|h_q\|} \tag{6.5}$$

where $h_v^{\text{proj}} = \mathbf{W}_q h_v^{\text{out}}$ is the projected embedding of node v, and h_q is the query embedding.

In our framework, the GAT model is not pre-trained across all questions. Instead, for each question, we build a specific subgraph and then train a new GAT model just for that question. This process is called "unsupervised" because there are no human-annotated labels indicating which nodes are relevant. The only signal used for learning comes from the question embedding itself: the model tries to identify and rank nodes whose representations are most similar to the question. As a result, for every question, the GAT learns to highlight nodes in the subgraph that are most likely to help answer that question, based solely on the structure of the knowledge graph and the meaning of the question, without any manual labeling or prior training.

Algorithm 6.1 Graph Attention-Based Node Classification

Input: Question q, subgraph $\mathcal{G}_q = (\mathcal{V}_q, \mathcal{E}_q)$, hyperparameters T, K, L, similarity threshold θ_0.

Output: Relevant node set V_{relevant}.

Initialize:

- Node features $H^{\text{in}} = [M(v) \mid v \in \mathcal{V}_q]$ using SentenceTransformer.
- Construct adjacency matrix $\mathbf{A} \in \mathbb{R}^{|\mathcal{V}_q| \times |\mathcal{V}_q|}$, where:

$$A_{ij} = \begin{cases} 1, & \text{if } (v_i, v_j) \in \mathcal{E}_q \\ 0, & \text{otherwise} \end{cases}$$

- Initialize multi-head attention parameters.

for $t = 1$ to T **do** ▷ For each epochs
  for $l = 1$ to L **do** ▷ For each attention layer
    for $k = 1$ to K **do** ▷ For each attention head
      Compute attention-based representation:

$$h_v^{(l,k)} \;=\; \phi\big(\{\, m_{v' \to v} : v' \in \mathcal{N}(v), i^{(k)}\}\big)$$

    end for
    Aggregate node representations explicitly:

$$h_v^{(l)} = \psi\Big(h_v^{(l-1)}, \{h_v^{(l,k)}\}_{k=1}^{K}\Big)$$

  end for
  Obtain final node representation h_v^{out} (from the final layer L).
  Project node embeddings explicitly:

$$h_v^{\text{proj}} = W_q h_v^{\text{out}}$$

  Calculate cosine-based dissimilarity loss explicitly:

$$\mathcal{L}_{\text{GAT}}(v) = 1 - \frac{h_v^{\text{proj}} \cdot h_q}{\|h_v^{\text{proj}}\| \|h_q\|}$$

  Backpropagate and update parameters using $\mathcal{L}_{\text{GAT}}(v)$.
end for
for each $v \in \mathcal{V}_q$ **do** ▷ Compute relevance of each node
  Project node embedding to query space: $h_v^{\text{proj}} = \mathbf{W}_q h_v^{\text{out}}$
  Compute cosine similarity with query embedding: $S_v = \frac{h_v^{\text{proj}} \cdot q}{\|h_v^{\text{proj}}\| \|q\|}$
end for
Select Relevant Nodes: $V_{\text{relevant}} = \{v \mid S_v > \theta_0\}$
Output: Relevant node set V_{relevant}.

6.4.4 Shortest Path Extraction

To construct coherent reasoning chains between the identified relevant nodes and potential answer concepts, we extract the shortest paths within $\mathcal{G}_K$. For each pair comprising a question-relevant entity and an answer candidate, we compute the shortest path based on the graph topology. These paths consist of sequences of knowledge graph triples that capture meaningful semantic relationships, such as *(parenchymal thinning, is_a_risk_factor_of, atrophy)* and *(bladder hypertrophy, is_a_subtype_of, hyperplasia)*. By focusing on the shortest paths, we maintain the interpretability and contextual relevance of the reasoning process while reducing noise from indirect or unnecessarily complex connections. These paths function as concise, clinically meaningful explanations that bridge the gap between medical phenomena and answer choices, providing structured knowledge traces to guide answer generation.

6.4.5 Knowledge-to-Passage Conversion

Once the reasoning paths have been extracted, we convert the structured graph-based knowledge into natural language passages suitable for large language models (LLMs). Each path is linearized into coherent textual sequences that maintain the semantic relationships between entities and relations. We provide a detailed overview of relation mapping in Table 6.1. The left column lists the relation types, while the right column presents the natural language templates that we use to verbalize each triple. For example, the triple *(parenchymal thinning, is_a_risk_factor_of, atrophy)* is verbalized into the passage: “Parenchymal thinning is a risk factor for atrophy". This transformation enables the LLM to process graph-derived knowledge in a format aligned with its pretraining paradigm, effectively bridging the modality gap between structured data and unstructured text. Furthermore, assembling these passages provides contextualized prompts that enrich the LLM's input space, enhancing its capacity to reason over medical knowledge and generate accurate predictions.

Table 6.1: Natural language template phrases for relations in our knowledge graph.

Relation Type	**Template Phrase**
is_a_subtype_of	is a subtype of
belongs_to_the_category_of	belongs to the category of
may_cause	may cause
may_be_allelic_with	may be allelic with
is_a_risk_factor_of	is a risk factor for
is_associated_with	is associated with
see_also	see also
interacts_with	interacts with
may_contraindicate	may contraindicate
belongs_to_the_drug_family_of	belongs to the drug family of
is_a_vector_for	is a vector for
belongs_to_drug_super-family	belongs to the drug super-family
is_a_category	is a category of
is_an_ingredient_of	is an ingredient of
may_treat	may treat

6.4.6 Answer Generation

In the final stage, we input the assembled knowledge passages alongside the original question into an LLM to perform answer generation. An LLM, augmented by the curated and contextually relevant evidence, synthesizes information across the provided passages to infer the most likely answer among the candidate options. By grounding the generation process in explicit reasoning paths derived from the knowledge graph, we enhance the interpretability and reliability of the model's predictions. The integration of knowledge-aware prompts enables the LLM to effectively navigate complex medical scenarios, yielding answers that are not only accurate but also supported by verifiable knowledge traces.

To systematically elicit the final answer, we design the following prompt template, encouraging introspective reasoning based on the provided passages:

```
You are a biomedical question answering assistant.
Your task is to read the given question, the candidate answer
    options, and the retrieved passage evidences, then generate the
    most accurate answer.

You are given:
```

```
- A multiple-choice question: {question}
- Candidate answer options: {candidate_list}
- Retrieved passage evidences: {passages_formatted}

Your task:
1. Carefully read all the passage evidences and understand their
    content.
2. Identify which information in the evidences directly supports or
    contradicts each candidate option.
3. Select the candidate answer that is most strongly supported by
    the evidence.
4. When necessary, use multi-hop reasoning across multiple passages
    to connect facts to the question.

Guidelines:
- Always ground your reasoning in the provided passages; do not rely
     on external knowledge or unsupported assumptions.
- If none of the passages support an answer, indicate that the
    evidence is insufficient.
- Provide a short explanation (from 2 to 4 sentences) that shows how
     the selected evidence leads to the final answer.
- Focus on factual accuracy, logical flow, and clear justification.

Your final output should be in the following format:

Answer: <final_answer_choice>
Explanation: <concise explanation citing relevant passages>
```

To illustrate final stage in our pipeline, we provide a concrete biomedical multiple-choice example in which the model is prompted with a question, four candidate options, and a small set of retrieved evidential passages converted from the upstream knowledge representation. In this instance, the passages are intentionally arranged to support multi-hop inference: they link chronic urethral obstruction to hydronephrosis and increased pressure, and then relate pressure-induced pathological changes to kidney parenchymal outcomes, enabling the LLM to select the option most strongly supported by evidence rather than relying on prior knowledge. The prompt also constrains the model to ground its decision in the provided passages and to return a brief justification, making the final prediction both traceable and suitable for error analysis.

```
You are a biomedical question answering assistant.
Your task is to read the given question, the candidate answer
```

```
    options, and the retrieved passage evidences, then generate the
    most accurate answer.

You are given:
- A multiple-choice question: Chronic urethral obstruction due to
    benign prismatic hyperplasia can lead to the following change in
     kidney parenchyma
- Candidate answer options:
  A) Hyperplasia
  B) Hyperophy
  C) Atrophy
  D) Dyplasia
- Retrieved passage evidences:
  [Passage 1] Parenchymal thinning is a risk factor for atrophy.
  [Passage 2] Hydronephrosis is associated with increased pressure.
  [Passage 3] Increased pressure is a risk factor for bladder
    hypertrophy.
  [Passage 4] Bladder hypertrophy is a subtype of hyperplasia.

Your task:
1. Carefully read all the passage evidences and understand their
    content.
2. Identify which information in the evidences directly supports or
    contradicts each candidate option.
3. Select the candidate answer that is most strongly supported by
    the evidence.
4. When necessary, use multi-hop reasoning across multiple passages
    to connect facts to the question.

Guidelines:
- Always ground your reasoning in the provided passages; do not rely
     on external knowledge or unsupported assumptions.
- If none of the passages support an answer, indicate that the
    evidence is insufficient.
- Provide a short explanation (from 2 to 4 sentences) that shows how
     the selected evidence leads to the final answer.
- Focus on factual accuracy, logical flow, and clear justification.

Your final output should be in the following format:

Answer: <final_answer_choice>
Explanation: <concise explanation citing relevant passages>
```

6.5 Experiment Results

6.5.1 Experiment Preparation

We evaluate our system using three established multiple-choice question datasets that assess a range of medical knowledge and reasoning abilities. Specifically, the benchmarks include the following:

- **MedQA** [194] is a large multiple-choice question-answering dataset created from USMLE exam questions. It includes questions in English, simplified Chinese, and traditional Chinese. For our evaluation, we use the English subset, denoted as MedQA (USMLE), which contains 1,273 questions.
- **MedMCQA** [195] is a large-scale dataset containing over 194k MCQs drawn from the Indian NEET PG and AIIMS medical entrance exams. It spans 21 medical subjects and approximately 2,400 healthcare topics, with our evaluation focusing on the development set, which comprises 4,183 questions.
- The medical subset of the Massive Multitask Language Understanding (**MMLU-Med**) benchmark [196] includes 1,089 four-option test questions across six medically relevant subjects: Anatomy, Clinical Knowledge (Clinical. KG), Professional Medicine (Pro. Med), Medical Genetics (Med. Gen), College Medicine (Col. Med), and College Biology (Col. Bio). This subset is specifically designed to evaluate model performance in specialized biomedical domains.

Next, our evaluation includes two major categories of models. The first category consists of *closed-book models*, which are pre-trained or fine-tuned specifically for the medical domain. These models rely entirely on their internal parameters to answer questions, without retrieving or consulting any external data sources during inference. Representative models in this group include **Meditron-70B** [197], **Med-PaLM 2** [198], **UltraMedical** [199], **LLM-AMT** [200], **BiMediX2** [201], **HuatuoGPT-o1** [202], and **OpenBioLLM** [203]. The second category, *external knowledge-augmented models*, integrates information retrieved from external knowledge sources, such as Wikipedia or Textbooks [194], to support the answer generation process. Notable works in this category include **Codex** [204], **MIRAGE** [205], and **Med-R**^{2} [206]. These frameworks aim to enhance their internal reasoning capabilities by leveraging external knowledge sources, thereby improving accuracy on complex medical queries.

6.5.2 Main Results

We evaluate our framework across two prominent medical question-answering benchmarks, **MedQA (USMLE)** and **MedMCQA (Dev)**, as well as six medical-related subsets from the **MMLU** benchmark: *Anatomy*, *Clinical Knowledge*, *College Biology*, *College Medicine*, *Medical Genetics*, and *Professional Medicine*. The results, summarized in Table 6.2 and Table 6.3, compare our models against a range of strong baselines, including Med-PaLM 2 [198], UltraMedical [199], OpenBioLLM [203], BiMediX2 [201], HuatuoGPT-o1 [202], and Meditron-70B [197]. To evaluate the performance of our models, we use the Accuracy metric in our experiments.

Table 6.2: Performance of baseline and our proposed models on MedQA (USMLE) and MedMCQA (Dev). The best and second-best results are shown in **bold** and underline, respectively.

Models	**MedQA (USMLE)**	**MedMCQA (Dev)**
Baseline		
Meditron-70B [197]	52.00	53.30
Med-PaLM 2 [198]	85.40	72.30
UltraMedical [199]	85.40	<u>74.70</u>
LLM-AMT [200]	88.10	74.60
BiMediX2 [201]	74.30	70.50
Codex [204]	78.20	62.70
MIRAGE [205]	83.97	69.88
HuatuoGPT-o1 [202]	83.30	73.60
OpenBioLLM [203]	78.16	74.01
Med-R^2 [206]	<u>86.37</u>	74.48
Our implementation		
UGAT-MedQA (Qwen2.5-72B)	**86.71**	**77.50**
UGAT-MedQA (Qwen2.5-14B)	65.33	57.14
UGAT-MedQA (LLama3.3-70B)	75.05	70.28

Table 6.2 presents that our UGAT-MedQA with Qwen2.5-72B model achieves 77.50% accuracy on MedMCQA and 86.71% on MedQA (USMLE), both the highest among all models evaluated. In addition, UGAT-MedQA also achieves leading performance across all benchmarks, obtaining an average score of 91.02% on the MMLU medical subsets, outperforming OpenBioLLM (90.39%) and UltraMedical (89.90%), as shown in Table 6.3. These results indicate the effectiveness of our proposed method compared to other robust baseline models on multiple-choice medical question-answering

tasks across diverse benchmarks.

Table 6.3: Performance of baseline and our proposed models on MMLU medical subsets. – denotes results not evaluated from prior publications. * denotes results not reported from prior publications. The best and second-best performances are highlighted in **bold** and underline, respectively.

Models	Anatomy	Clinical. KG	Col. Bio	Col. Med	Med. Gen	Pro. Med	Average
Baseline							
Meditron-70B [197]	69.40	75.50	86.70	68.00	85.90	82.30	77.97
Med-PaLM 2 [198]	84.40	88.70	95.80	83.20	92.00	95.20	89.88
UltraMedical [199]	85.20	89.40	95.10	82.10	**95.00**	92.60	89.90
LLM-AMT [200]	–	–	–	–	–	–	–
BiMediX2 [201]	82.20	86.80	95.10	79.80	94.00	91.50	88.23
Codex [204]	–	–	–	–	–	–	–
MIRAGE [205]	*	*	*	*	*	*	87.24
HuatuoGPT-o1 [202]	–	–	–	–	–	–	–
OpenBioLLM [203]	83.90	**92.93**	93.83	**85.75**	93.20	93.75	90.39
Med-R^2 [206]	*	*	*	*	*	*	84.65
Our implementation							
UGAT-MedQA (Qwen2.5-72B)	**89.27**	90.71	**96.22**	79.74	93.32	**96.81**	**91.02**
UGAT-MedQA (Qwen2.5-14B)	70.89	83.61	88.48	80.63	86.96	83.04	82.27
UGAT-MedQA (LLama3.3-70B)	66.90	69.08	75.87	58.51	77.56	74.28	70.37

Across the MMLU domains, the UGAT-MedQA with Qwen2.5-72B achieves the best performance in *Anatomy* (89.27%), *College Biology* (96.22%), and *Professional Medicine* (96.81%), and ranks second in *Clinical Knowledge* (90.71%). These results highlight the strong ability of the model to reason over both foundational biomedical knowledge and clinical scenarios.

Moreover, to investigate the efficiency of UGAT-MedQA besides Qwen2.5-72B, we evaluate UGAT-MedQA with two smaller LLMs, including Qwen2.5-14B and Llama3.3-70B. According to Table 6.3, Qwen2.5-14B achieves competitive results, reaching 82.27% on the MMLU subsets, while Llama3.3-70B achieves 70.37%, with comparatively lower performance on MedMCQA. These results demonstrate that UGAT-MedQA significantly boosts the performance of LLMs on the medical question answering task.

Overall, these results demonstrate that our UGAT-MedQA framework, combining unsupervised graph attention-based node selection with retrieval-augmented answer generation, achieves substantial improvements over strong baselines. These results suggest that attention-based subgraph selection adds structured medical evidence while avoiding the irrelevant nodes that often accumulate under naive multi-hop expansion. Remarkably, our fully open-source pipeline is able to exceed the performance of many closed-source medical QA systems in complex medical reasoning tasks.

6.5.3 Ablation Study

To assess the contribution of our retrieval-enhanced reasoning framework, we conduct an ablation study in two scenarios: (1) the LLM answers using only the question and answer options, with no access to retrieved knowledge or intermediate reasoning paths (w/o knowledge retrieval); (2) the LLM answers using retrieved subgraphs but without GAT-based node filtering (w/o node filtering); and (3) we keep the node filtering stage and the same query, where we replace the Graph Attention Network with a Graph Convolutional Network (*w/o attention*). The results are reported in Table 6.4 and Table 6.5.

Table 6.4: Ablation study results on MedQA (USMLE) and MedMCQA (Dev).

Models	**Settings**	**MedQA (USMLE)**	**MedMCQA (Dev)**
Qwen2.5-72B	*Full*	**86.71**	**77.50**
	w/o knowledge retrieval	74.97	67.75
	w/o node filtering	79.20	70.18
	w/o attention	84.46	75.30
Qwen2.5-14B	*Full*	**65.33**	**57.14**
	w/o knowledge retrieval	59.46	50.92
	w/o node filtering	61.22	53.47
	w/o attention	64.10	56.04
LLama3.3-70B	*Full*	**75.05**	**70.28**
	w/o knowledge retrieval	66.52	59.43
	w/o node filtering	70.09	65.15
	w/o attention	73.56	68.74

To quantify the contribution of each component in our retrieval-enhanced reasoning framework, we evaluate four settings on MedQA (USMLE) and MedMCQA (Dev), as reported in Table 6.4. The *Full* setting consistently achieves the best performance for all LLM backbones. For Qwen2.5-72B, the full framework attains an accuracy of 86.71% on MedQA and 77.50% on MedMCQA. When knowledge retrieval is removed (*w/o knowledge retrieval*), performance drops markedly to 74.97% on MedQA and 67.75% on MedMCQA, confirming that answering from the question and options alone is insufficient for knowledge-intensive medical reasoning. When we keep retrieved subgraphs but disable node filtering (*w/o node filtering*), Qwen2.5-72B improves to 79.20% on MedQA and 70.18% on MedMCQA, indicating that raw subgraph evidence is helpful; however, the gains remain limited because simple neighborhood expansion introduces substantial irrelevant or noisy information. To isolate the effect of attention-based neighbor selection, we further test *w/o attention (GCN-based node filtering)*,

which retains the same query-aligned scoring and thresholding strategy but replaces the GAT encoder with a GCN-style mean aggregation. Under this configuration, Qwen2.5-72B reaches 84.46% on MedQA and 75.30% on MedMCQA, outperforming *w/o node filtering* by 5.26 and 5.12 absolute points, respectively, while remaining below the full model by 2.25 and 2.20 points. For Qwen2.5-14B, the full framework achieves 65.33% on MedQA and 57.14% on MedMCQA, whereas *w/o knowledge retrieval* yields 59.46% and 50.92%, and *w/o node filtering* yields 61.22% and 53.47%; introducing *w/o attention (GCN-based node filtering)* raises performance to 64.10% and 56.04%. For Llama3.3-70B, the full framework achieves 75.05% on MedQA and 70.28% on MedMCQA; removing retrieval lowers results to 66.52% and 59.43%, and removing node filtering yields 70.09% and 65.15%, while *w/o attention* recovers to 73.56% and 68.74%. Overall, Table 6.4 shows a clear hierarchy: knowledge retrieval provides a large performance foundation, node filtering reduces noise introduced by neighborhood expansion, and attention further strengthens filtering quality by selectively emphasizing the most question-relevant neighbors, thereby enabling the strongest downstream LLM reasoning.

Table 6.5: Ablation study results on MMLU medical subsets.

Models	Settings	Anatomy	Clinical. KG	Col. Bio	Col. Med	Med. Gen	Pro. Med	Average
Qwen2.5-72B	*Full*	**89.27**	**90.71**	**96.22**	**79.74**	**93.32**	**96.81**	**91.02**
	w/o knowledge retrieval	81.34	81.82	90.21	72.67	85.86	88.56	83.41
	w/o node filtering	84.25	84.40	90.95	74.86	88.70	90.12	85.99
	w/o attention	87.76	88.82	94.64	78.28	91.93	94.80	89.51
Qwen2.5-14B	*Full*	**70.89**	**83.61**	**88.48**	**80.63**	**86.96**	**83.04**	**82.27**
	w/o knowledge retrieval	64.92	77.65	82.51	71.51	78.78	77.86	75.54
	w/o node filtering	66.28	77.79	81.42	74.11	80.30	77.32	75.79
	w/o attention	69.51	81.86	86.36	78.67	84.96	81.32	80.33
LLama3.3-70B	*Full*	**66.90**	**69.08**	**75.87**	**58.51**	**77.56**	**74.28**	**70.37**
	w/o knowledge retrieval	62.69	62.50	69.23	54.07	72.72	69.01	65.04
	w/o node filtering	62.24	64.86	70.37	54.19	72.47	68.12	65.35
	w/o attention	65.50	67.81	74.22	57.21	76.03	72.43	68.86

We further validate these findings on six medical subsets of MMLU, with results summarized in Table 6.5. Across all backbones, the full framework yields the best average accuracy, and removing knowledge retrieval causes the largest degradation. For Qwen2.5-72B, the full framework achieves an average accuracy of 91.02% across Anatomy, Clinical Knowledge, College Biology, College Medicine, Medical Genetics, and Professional Medicine. Without knowledge retrieval, the average accuracy decreases to 83.41%. When node filtering is disabled but subgraphs are still provided (*w/o node filtering*), the average accuracy increases to 85.99%, which is higher than the retrieval-free setting but still substantially below the full framework due to the inclusion of many irrelevant nodes and relations from unfiltered neighborhood expansion. Under *w/o attention (GCN-based node filtering)*,

the average accuracy rises to 89.51%, demonstrating that non-attentive message passing can already improve relevance estimation by refining node representations through neighborhood aggregation. However, the remaining gap to the full model is still evident, with 91.02% under *Full* compared to 89.51% under *w/o attention*, indicating that attention-based neighbor weighting provides additional benefit beyond generic propagation. This trend is consistent for Qwen2.5-14B and Llama3.3-70B. For Qwen2.5-14B, the average accuracy is 82.27% in the full framework, 75.54% without retrieval, 75.79% without node filtering, and 80.33% without attention. For Llama3.3-70B, the average accuracy is 70.37% in the full framework, 65.04% without retrieval, 65.35% without node filtering, and 68.86% without attention. Taken together, Table 6.5 confirms that retrieval is necessary to supply critical domain knowledge, node filtering mitigates the noise of naive neighborhood expansion, and attention is a key mechanism for selecting the most informative neighbors, which ultimately yields the most accurate and robust medical question answering across diverse medical subdomains.

In general, the ablation study results highlight the essential role of GAT-based attention in our approach. While simple subgraph expansion provides some useful information, it often includes irrelevant or distracting details that limit its effectiveness. In contrast, GAT allows the framework to focus precisely on the most relevant entities and connections for each question, providing clear and high-quality evidence that significantly improves question-answering accuracy.

6.6 Results Analysis

6.6.1 Analysis of Number of Hops

Our UGAT-MedQA framework is specifically designed to reason over subgraphs constructed solely from biomedical concepts extracted from the question, deliberately excluding any information from the answer candidates to ensure that the inference process remains general and unbiased. However, the framework's effectiveness depends on two factors: the accuracy of entity linking and the structure of the knowledge graph. If key concepts are not correctly linked to entities, or if these entities are poorly connected in the graph, the model may struggle to build complete reasoning paths. In such cases, the shortest path module may return empty or incomplete paths, which deprives the language model of sufficient evidence to generate an accurate answer. It is important to note that this limitation stems not from the UGAT-MedQA architecture itself, but from upstream dependencies, specifi-

cally the robustness of concept-to-entity alignment and the local connectivity of the knowledge graph.

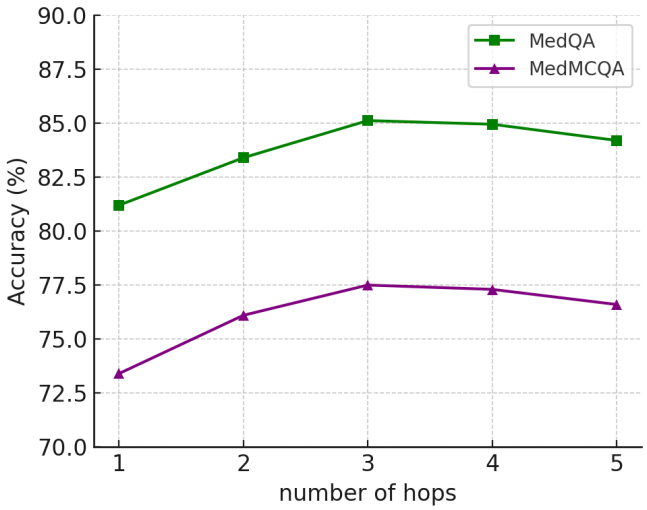


Figure 6.3: Effect of the number of hops ($D_{\max}$) on accuracy for MedQA and MedMCQA. Accuracy peaks at $D_{\max} = 3$, demonstrating the importance of multi-hop expansion for capturing relevant reasoning paths while controlling information noise.

Figure 6.3 demonstrates that accuracy for both MedQA and MedMCQA reaches its highest point at $D_{\max} = 3$, highlighting the importance of multi-hop expansion for capturing relevant reasoning paths while effectively controlling information noise. To show why looking at more hops helps, we looked at example cases comparing models set to 1 hop versus 3 hops, as shown in Figure 6.4. In one clinical scenario with symptoms like blurry vision, chest pain, and fast heartbeat, a 1-hop model can only find facts directly tied to those symptoms—like possible findings or simple associations. But it can't connect those symptoms to deeper causes, such as drug toxicity (from cocaine or nicotine), and often makes the wrong choice. When allowed to explore up to 3 hops, the model can find links to important concepts like specific drugs, substance abuse, and toxin exposure. This richer reasoning chain helps the model connect symptoms to their likely causes and make a more accurate prediction, though sometimes it can still get confused if irrelevant facts slip in. Overall, these findings emphasize the necessity of controlling subgraph scope to balance completeness and relevance in graph-based medical question answering.

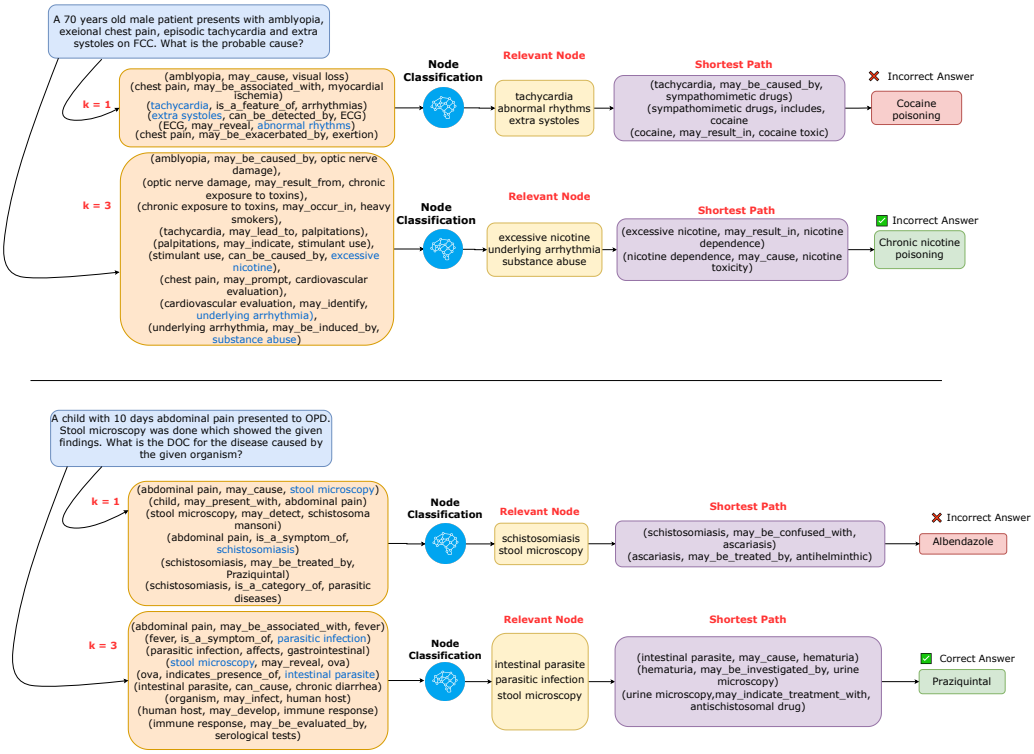


Figure 6.4: Qualitative comparison of 1-hop and 3-hop reasoning chains on representative clinical and parasitic infection cases. 3-hop expansion enables the model to access richer biomedical knowledge and establish informative reasoning paths connecting clinical features to underlying causes and treatments.

Overall, we saw a similar pattern with questions about parasitic infections. With only 1 hop, the model just links symptoms like abdominal pain to basic tests or organisms, but can't reach the right treatment. With 3 hops, it discovers not only the parasite but also the correct medicine and related conditions, leading to much better answers. These examples show why being able to "reason" over several steps in the knowledge graph is so helpful, but also why we need to be careful not to go too far and include irrelevant details.

6.6.2 Error Analysis

A key limitation of UGAT-MedQA emerges in scenarios that require precise, guideline-based fact-checking. As shown in Figure 6.5, consider the following example question: "Lamivudine used as monotherapy in post-exposure prophylaxis (t/f)". After constructing the reasoning subgraph via entity linking

and three-hop neighborhood expansion, the framework successfully retrieves facts indicating that lamivudine is used for HIV, that antivirals are used in prophylaxis, and that prophylaxis is recommended after needle injury. However, the subgraph does not contain any triples specifying whether lamivudine monotherapy is recommended or if dual therapy is required.

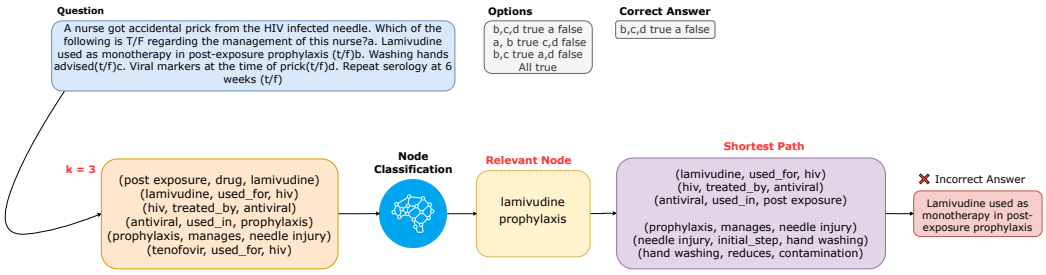


Figure 6.5: Illustration of a reasoning failure in UGAT-MedQA: the subgraph includes relevant facts but omits explicit guideline constraints, leading the model to incorrectly accept monotherapy for post-exposure prophylaxis.

When presented with this statement, the model, relying solely on the available knowledge graph paths, predicted it as **true**. However, this is incorrect, since current clinical guidelines recommend combination therapy (such as tenofovir and lamivudine) rather than lamivudine monotherapy for post-exposure prophylaxis. The source of error here is not a failure of the UGAT-MedQA reasoning process itself, but rather the absence of explicit guideline-specific or negative evidence in the constructed subgraph. Because the knowledge graph does not encode that monotherapy is insufficient or outdated, the model cannot reject the incorrect statement.

This example highlights a key limitation of UGAT-MedQA. The reasoning subgraph is built automatically using entity linking and neighborhood expansion. If important entities are missed or connections are incomplete, the subgraph may fail to include critical information, even if the knowledge graph has the necessary data. As a result, the model cannot reliably distinguish between answers that are only partially correct and those that fully align with clinical guidelines. Ultimately, the effectiveness of guideline-based medical QA with UGAT-MedQA depends not only on the coverage, specificity, and up-to-dateness of the knowledge graph but also on the accuracy of the entity linking and subgraph extraction processes.

6.7 Conclusion

In this chapter, we presented UGAT-MedQA, a novel unsupervised framework that combines graph-based neural reasoning with large language models to advance multiple-choice medical question answering. By integrating a Graph Attention Network for node-level relevance assessment and extracting explicit multi-hop reasoning paths from knowledge graphs, UGAT-MedQA enables interpretable and guideline-oriented answer generation. Our experiments on MedMCQA and MMLU-Med benchmarks demonstrate that UGAT-MedQA achieves new state-of-the-art performance, outperforming strong baselines. While our approach demonstrates clear strengths in biomedical multiple-choice QA, its modular design suggests it can easily generalize to other knowledge-intensive domains such as law, finance, or general scientific reasoning by replacing the underlying knowledge graph and updating entity recognition. Through qualitative and quantitative error analysis, we identified that the ultimate effectiveness of the framework depends on both the coverage and specificity of the underlying knowledge graph and the accuracy of entity linking and subgraph construction processes. Future work could focus on integrating multiple complementary medical knowledge graphs to increase coverage and reduce missing links, as well as developing automated KG updating methods to keep information current. Such advances would help address issues of outdated or incomplete evidence and improve the reliability of model outputs in fast-evolving medical domains. Furthermore, the interpretable reasoning chains produced by our approach may help facilitate integration into future clinical decision support systems, thereby advancing transparency and trustworthiness in medical artificial intelligence systems. Overall, UGAT-MedQA represents a promising step toward transparent and robust biomedical question answering by tightly coupling structured graph reasoning with the generative capabilities of LLMs. Moreover, its modular design facilitates easy adaptation to other knowledge-intensive domains beyond biomedicine, such as legal, financial, or general scientific reasoning. In summary, UGAT-MedQA's novelty is unsupervised, question-driven graph attention that turns a biomedical KG into an interpretable reasoning path for QA.

Chapter Summary

This chapter presents UGAT-MedQA, an unsupervised framework for biomedical question answering that performs evidence-based reasoning over a biomedical knowledge graph. Building on the triples extracted in K-Bloom,

we first construct a domain KG that makes part of the model's latent knowledge explicit and reusable. Given a question and its answer options, the system retrieves a query-focused subgraph and then applies an unsupervised Graph Attention mechanism to identify the most salient nodes and relations. From these high-attention regions, UGAT-MedQA extracts multi-hop paths that connect question concepts to candidate answers, providing a compact set of structured evidence. To keep the reasoning inspectable, the selected paths are verbalized into natural-language statements that can be read and checked by humans. An LLM then uses these verbalized chains to produce a step-by-step answer grounded in the retrieved evidence, rather than relying purely on parametric recall. The chapter also discusses the practical trade-off between evidence coverage and noise when expanding to longer-hop neighborhoods and shows why attention-based filtering is needed to control this noise. We further analyze typical failure cases, such as missing or incomplete KG links and situations where different paths support competing interpretations. Overall, UGAT-MedQA demonstrates how the thesis philosophy—explicit knowledge, unsupervised evidence selection, and verifiable reasoning traces—can be realized in the graph modality for medical QA. In the next chapter, we move from graph-based evidence to unstructured text retrieval, introducing USCRaKE to improve evidence selection with distribution-aware semantic matching.

Chapter 7

USCRaKE: Unsupervised Semantic Chunk Retrieval and Knowledge-Enhanced Reasoning for Multiple-Choice Question Answering

7.1 Introduction

The multiple-choice question answering (MCQA) task requires not only factual recall but also the ability to reason through multiple, interrelated concepts. Although pre-trained language models such as BERT [207] and RoBERTa [208] have delivered strong performance on a wide range of NLP benchmarks, the reasoning behaviors they exhibit are largely an emergent by-product of learning from unstructured corpora. Consequently, their intermediate decisions are often opaque, and their predictions can be brittle in settings that require relational knowledge, compositional linking, or multi-hop inference [51].

Structured resources, notably knowledge graphs such as ConceptNet [163] and Freebase [136], offer a complementary perspective by encoding facts as interpretable entity–relation triples. When paired with PLMs, KG evidence can enable more explicit and explainable forms of reasoning [177, 179]. However, many KG-assisted MCQA pipelines still depend on fixed, coarse, or generic triple retrieval strategies that are insufficiently sensitive to the specific context of each question [209, 210]. In parallel, Chain-of-Thought (CoT) prompting [211, 212] encourages large language models to produce stepwise explanations and often improves accuracy on challenging problems. Despite this benefit, CoT can also amplify inconsistency: different reasoning trajectories may produce conflicting conclusions [213, 214], and long, monolithic rationales make it difficult to attribute the final decision to particular facts or intermediate steps. Moreover, many retrieval-based approaches [29–33, 35] rely on Euclidean distance or cosine similarity, which can be too coarse

to capture nuanced semantic correspondences in contextualized embedding spaces, especially when multi-hop reasoning requires fine-grained alignment.

These observations motivate a reasoning framework that is modular, interpretable, and able to combine unstructured textual evidence with structured KG cues, while keeping the reasoning process for each answer option separated and traceable. Such a design supports clearer inspection of model behavior, more actionable error analysis, and greater robustness even when the final prediction is incorrect. Following this direction, we introduce **USCRaKE** (Unsupervised Semantic Chunk Retrieval and Knowledge-Enhanced Reasoning), an unsupervised hybrid retrieval-and-reasoning framework for MCQA. To the best of our knowledge, USCRaKE is the first to employ Jensen–Shannon Divergence (JSD) as the ground cost within an optimal transport formulation for evidence retrieval, providing a principled alternative to standard Euclidean or cosine measures. Compared with these conventional metrics, JSD is symmetric and bounded, and it is better suited to comparing distributional representations under shift, which is particularly relevant when contextualized semantics vary across domains. This integration of JSD into optimal transport directly targets the need for more robust semantic similarity in retrieval settings where subtle distributional alignment can materially affect downstream answer accuracy.

USCRaKE draws evidence from two complementary channels. First, for semantic chunk retrieval, we formulate evidence selection as an unsupervised optimal transport problem and use JSD as the ground cost, rather than the Euclidean or cosine distances commonly adopted in prior work. This choice yields a probabilistically grounded alignment between question–answer pairs and candidate chunks, improving robustness across diverse datasets and domains while retaining a clear theoretical motivation. Second, we inject structured cues by retrieving 1-hop subgraphs from ConceptNet and converting the retrieved triples into natural-language statements. Instead of collapsing all evidence into a single aggregated rationale, USCRaKE constructs *separate* reasoning chains for each answer choice. This dual-evidence strategy allows OT–JSD retrieval to capture fine-grained semantic correspondence, while KG retrieval supplies explicit relational signals without introducing additional graph encoders such as GNNs [215]. By verbalizing triples, we enable pretrained language models to integrate external knowledge with question–answer context in a lightweight and interpretable manner. Overall, the modular design supports more precise assessment of contextual relevance and logical coherence, strengthening both interpretability and answer selection quality, while also providing a theoretically grounded lens for semantic alignment in MCQA retrieval.

To evaluate generalization, we test USCRaKE on multiple MCQA bench-

marks spanning both general and specialized settings, including biomedical and physics-focused datasets. Across these domains, USCRaKE consistently performs strongly and surpasses competitive unsupervised and few-shot baselines by up to 10.08% absolute accuracy, without any supervised fine-tuning.

In summary, this chapter makes the following contributions:

- We present USCRaKE, a fully unsupervised hybrid framework that jointly leverages fine-grained semantic chunk evidence and structured KG triples, enabling complementary and interpretable inference over unstructured and structured sources.
- We provide a theoretical motivation and analysis for using JSD as the ground cost in optimal transport-based semantic retrieval, highlighting its advantages over Euclidean distance and cosine similarity under distributional shift in contextualized embeddings.
- We propose a modular reasoning architecture that builds and evaluates distinct explanation chains for each answer candidate, improving transparency, interpretability, and robustness in MCQA answer selection.

The remainder of this chapter is organized as follows. Section 7.2 reviews related work, and Section 7.3 details the proposed framework. Section 7.4 describes the experimental setup. Results and ablations are reported in Section 7.5, followed by error analysis in Section 7.6. Finally, Section 7.7 concludes the chapter and discusses future directions.

7.2 Related Works

7.2.1 Retrieval augmented question answering systems

Retrieval-Augmented Generation (RAG) [9] has become a widely adopted paradigm for open-domain question answering and other knowledge-intensive NLP tasks, largely because it improves evidence grounding, scales to large corpora, and mitigates hallucination relative to fully parametric generators. In its canonical form, RAG couples a parametric sequence-to-sequence generator (e.g., BART [64]) with a non-parametric memory implemented as an indexed collection of dense document representations, typically searched using FAISS [63]. For an input query, the retriever maps the query into a dense vector, retrieves a small set of relevant passages from an external knowledge store, and the generator then produces an answer conditioned on these retrieved contexts. A key advantage of this design is that the overall system can be trained end-to-end, enabling joint optimization of retrieval

and generation while explicitly tying outputs to external evidence.

Empirically, RAG-style pipelines have shown strong results on benchmarks such as Natural Questions [65], where access to external documents helps improve factual accuracy over parametric-only baselines. Similar benefits have also been reported in knowledge-grounded dialogue generation [66], in which grounding responses in retrieved passages tends to increase informativeness and faithfulness to source content.

Despite these strengths, many modern RAG pipelines [67,68] still rely on dense retrievers that utilize traditional metrics such as cosine or Euclidean distance to compute similarity over dense embeddings. For queries that require subtle semantic matching or multi-faceted reasoning, these metrics can be brittle: retrieval may return long chunks that are only weakly aligned with the query intent, and performance can become overly sensitive to the corpus chunking strategy. Moreover, a large family of RAG variants [69–72] improves in-domain accuracy by supervised joint tuning of retriever–generator pairs, but this often reduces portability when moving to unseen domains or settings with limited labeled data. These limitations motivate retrieval mechanisms that (i) capture fine-grained semantic alignment beyond standard embedding-space distances and (ii) remain effective without task-specific supervision. In contrast to prior pipelines that depend heavily on cosine or Euclidean similarity and supervised retriever tuning, the proposed USCRaKE framework introduces a theoretically grounded OT–JSD retrieval strategy that operates in an unsupervised manner, aiming to improve semantic matching while supporting stronger cross-domain generalization.

7.2.2 Knowledge Graph-Augmented Question Answering

Alongside unstructured text, many QA systems exploit structured knowledge graphs to enhance factual consistency and to make reasoning traces more interpretable. In our setting, ConceptNet is used as an auxiliary evidence source for multiple-choice question answering, which aligns with a broader literature on KG-augmented QA. Existing KG-based QA methods are commonly grouped into three lines: embedding-based approaches, semantic parsing approaches, and retrieval-augmented approaches.

Embedding-based methods learn continuous representations of entities and relations, and then perform reasoning using neural architectures such as key–value memory networks [75], sequence modeling approaches [76], or graph neural networks [2]. These models aim to connect the question to relevant KG elements by learning latent paths or aggregations over graph

neighborhoods.

In contrast, semantic parsing approaches translate natural language questions into formal queries (e.g., SPARQL) using a parser, and then execute the resulting query against a KG to obtain answers [77–80]. While this pipeline offers explicit, executable reasoning, it often externalizes the core inference to the KG engine, which can limit how much the neural model itself contributes to reasoning beyond parsing.

Retrieval-augmented KG methods seek a middle ground by retrieving relevant triples or subgraphs and injecting them into the model's inference process. Representative systems include GraftNet [81], which performs entity linking and expands local subgraphs, as well as PullNet [82], SR [83], DiFar [84], and UniKGQA [85], which employ dense retrieval to access semantically related KG content

With the emergence of large language models, retrieval-augmented generation has also been extended to incorporate structured and unstructured evidence within generative pipelines [9, 86–88]. In such settings, KG triples are frequently verbalized into natural language so that they can guide generation or support intermediate reasoning checks. For example, [89] leverages KG-derived facts to revise or correct hallucinated reasoning steps, while DECAF [90] jointly retrieves knowledge and produces both structured queries and free-form answers, combining advantages from semantic parsing and generative modeling.

Nevertheless, the usefulness of KGs is constrained by practical issues. Public KGs are often incomplete or noisy, and building high-quality graphs is costly due to ontology design challenges and annotation requirements. In addition, corpus-derived KG construction typically involves a trade-off: highly detailed graphs preserve information but increase computational overhead, whereas compact graphs improve efficiency at the risk of discarding critical relations. These factors suggest that effective KG use in QA should be lightweight, context-sensitive, and complementary to robust semantic retrieval over text.

Motivated by these observations, USCRaKE combines (i) optimal transport–based semantic chunk retrieval with Jensen–Shannon Divergence to achieve fine-grained alignment over unstructured evidence and (ii) 1-hop ConceptNet retrieval, whose triples are verbalized into natural language to provide structured relational grounding without training additional graph encoders. Overall, prior retrieval-augmented QA systems exhibit several limitations. First, many pipelines rely on relatively coarse embedding-space similarity functions (e.g., cosine similarity or Euclidean distance), which often fail to reflect fine-grained semantic correspondence, especially for compositional or multi-facet queries. Second, a substantial line of work improves accuracy by

supervised joint optimization of the retriever and generator; while effective in-domain, this strategy can reduce robustness and transferability when moving to new domains with different terminology and limited annotations. Third, although knowledge graphs can improve interpretability by providing explicit relations, KG-centric QA remains constrained by practical issues—including incompleteness, noise, and the non-trivial cost of building and maintaining high-quality graphs—so the resulting reasoning support is often neither comprehensive nor lightweight.

These observations motivate a unified framework that combines precise semantic retrieval over unstructured text with efficient, context-aware structured grounding. Such a hybrid design leverages the coverage and adaptability of textual evidence while preserving the explicit relational signals offered by knowledge graphs, thereby addressing the most common failure modes of prior KG-augmented QA methods. In this thesis, USCRaKE follows this principle by introducing an unsupervised OT–JSD retrieval mechanism, which is theoretically motivated to capture finer semantic alignment than standard distance-based dense retrieval. In addition, instead of training dedicated graph encoders over KG structure, USCRaKE applies lightweight 1-hop KG retrieval and verbalizes the retrieved triples into natural language, allowing LLMs to exploit structured relations directly without additional graph-model training. By integrating these two components, the framework simultaneously mitigates coarse retrieval alignment and avoids heavyweight KG modeling, thereby addressing key weaknesses in both retrieval-only RAG and KG-augmented QA approaches.

7.3 Methodology

In this section, we detail the proposed framework and clarify how its components interact. Section 7.3.1 first summarizes the overall design and task formulation. Next, Section 7.3.2 introduces the hybrid retrieval module, which gathers supporting evidence chunks and the associated knowledge triples. Building on the retrieved evidence, Section 7.3.3 describes how we instantiate a set of candidate-specific reasoning chains by pairing the input question with each answer option. Finally, Section 7.3.4 explains how the framework consolidates these candidate-wise chains to produce a single final prediction for the given question.

7.3.1 Problem Setup

We study multiple-choice question answering (MCQA), focusing on scenarios where solving a question benefits from multi-step reasoning over heterogeneous knowledge. Each dataset instance contains a question q and a collection of answer candidates $C = c_1, c_2, \ldots, c_n$. In addition to (q, C), the model has access to two external resources: (i) a structured knowledge graph G that stores relational facts between concepts, and (ii) an unstructured textual database DB (e.g., Wikipedia) that provides broader contextual descriptions. Given (q, C) together with G and DB, the objective is to select the correct option $c^* \in C$, and we evaluate performance using accuracy. An overview of the proposed framework is shown in Figure 7.1.

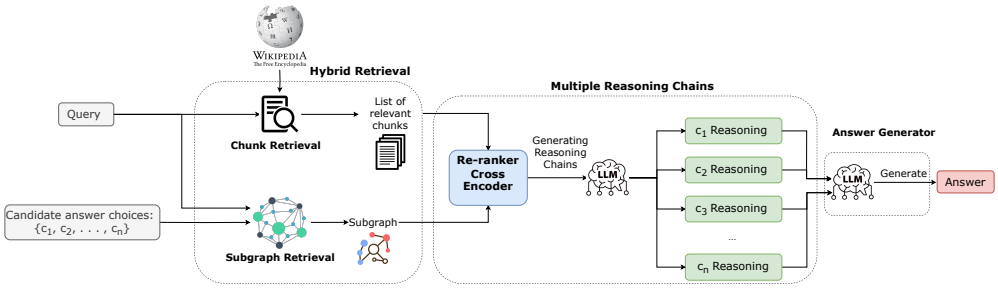


Figure 7.1: Architecture of the USCRaKE framework for multiple-choice question answering.

USCRaKE is designed to be unsupervised and interpretable, and it explicitly combines unstructured text evidence with structured relational knowledge for reasoning. The framework is organized into three main stages:

- **Hybrid Retrieval**: For each question–candidate pair (q, c_i), we identify relevant information from the corpus DB and complementary relational evidence from the knowledge graph G. For *chunk retrieval*, we use a semantic matching function inspired by WMD [159] to surface informative text chunks from DB. Instead of using the standard Euclidean transport cost, we employ JSD to quantify the distance between word distributions. Because JSD is symmetric and bounded, it is better suited to capturing fine-grained semantic discrepancies while mitigating noisy alignments. In parallel, for *Subgraph Retrieval*, we extract key entities from both the question and the candidate answer and then retrieve their one-hop neighbors in G to form relevant triples. These triples are subsequently verbalized into natural-language sentences, preserving the semantics of their original relational structure. Finally, we re-rank all retrieved evidence units—corpus chunks

and verbalized triples—using a pretrained cross-encoder that directly scores the relevance of each evidence unit concerning (q, c_i), thereby prioritizing the most informative and contextually aligned content.

- **Multiple Reasoning Chains**: To make the decision process transparent, we construct an independent *reasoning chain* for every candidate c_i. Each chain integrates both retrieved textual passages with the verbalized triples to explain why c_i may be correct or incorrect. Instead of collapsing all candidates into a single shared rationale, USCRaKE intentionally produces multiple candidate-specific chains and assesses them based on plausibility and internal consistency.
- **Answer Generator**: This component performs a comparative assessment over reasoning chains and outputs the answer whose explanation appears most convincing. By explicitly judging the quality of the generated explanations, this introspective step improves both the robustness of the final decision and the interpretability of the overall reasoning process.

To enable evaluation across diverse MCQA domains, we build a unified textual knowledge source from the October 1, 2023 Wikipedia snapshot[1] , which contains 6.41 million articles. We segment this resource into 11.4 million chunks, with each chunk limited to 512 tokens. Each chunk is encoded into a semantic embedding using the BGE model [126], and the resulting vectors are indexed with FAISS [216] to support efficient retrieval. This setup enables efficient matching between a question–candidate pair and semantically related chunks, serving as the foundation of our retrieval-augmented reasoning pipeline. Concretely, we first use FAISS to retrieve an initial candidate pool R (e.g., the top-20k chunks) from the full corpus as raw evidence. We then run our fine-grained chunk retriever only on this subset, which keeps the framework computationally feasible and focuses computation on the most informative evidence. The following sections describe these three stages of USCRaKE in detail.

7.3.2 Stage 1: Hybrid Retrieval

This section presents the hybrid retrieval stage and clarifies how its two complementary components interact. We first detail the proposed chunk retrieval module in Section 7.3.2.1, whose role is to identify evidence chunks that are semantically most aligned with the input question. Next, Section 7.3.2.2 describes the knowledge-graph retrieval module, which extracts relational triples that capture relational knowledge related to the same question.

[1] https://huggingface.co/datasets/wikimedia/wikipedia/viewer/20231101.en

7.3.2.1 Chunk Retrieval

We start by distinguishing **hard alignment** from **soft alignment** formulated through **Optimal Transport**, motivating an OT view of chunk retrieval.

We then cast **Word Mover's Distance** as a representative OT-based retrieval objective and highlight its main limitation: it relies on a **Euclidean transport cost**, which is insensitive to context and can misalign semantically close words. To address this issue, we substitute the Euclidean ground cost with a **Jensen–Shannon divergence** computed over **normalized contextual embeddings**. This yields a distributional transport cost that better captures semantic discrepancy. The key novelty of our retrieval design is to employ Jensen–Shannon divergence as the Optimal Transport ground cost, producing a bounded and symmetric similarity signal that supports fine-grained alignment between the question and candidate evidence.

Finally, we integrate **BM25-weighted marginals** and define the complete OT-based retrieval scoring function used to score and rank chunks. This stage returns the top-k chunks with the lowest scores, since these are the closest in similarity to the input question.

Limitations of Hard Alignment. Although BERTScore [217] is widely used to measure semantic similarity in natural language generation, its underlying **hard alignment** mechanism imposes a key limitation. In particular, BERTScore determines word-level similarity by matching each token in the system-generated text to the single most similar token in the reference text, thereby enforcing a strict **one-to-one matching** constraint. As shown in our prior work [187], this strategy yields a rigid correspondence: each word is forced to select only one counterpart, which can fail to capture the many-to-one or one-to-many associations that frequently occur in natural language. To address this limitation, we introduce our chunk retrieval approach as a more flexible way to model semantic alignment. In contrast to one-to-one matching, our method adopts "soft" alignments, allowing a word in one sequence to be linked to multiple semantically related words in the other sequence, and vice versa. Operationally, the approach is formulated as a constrained optimization problem that estimates the minimum effort required to transform one text into another, thereby providing a more comprehensive and flexible assessment of semantic similarity.

Word Mover's Distance as Soft Alignment. To retrieve the most relevant chunks for a question q from the corpus DB, we draw inspiration from **Word Mover's Distance** [159], which is a specific instance of Earth

Mover's Distance [160]. WMD leverages pre-trained word embeddings to calculate the dissimilarity between two text sequences. It assigns uniform weights over words and uses the Euclidean distance between embeddings as the **transport cost**. This mechanism provides an intuitive and effective way to assess semantic similarity by minimizing the total distance required to "move" words from one document to another.

Limitations of Euclidean Transport Cost. Despite its intuitive formulation, WMD exhibits notable limitations that affect its performance in the semantic textual similarity task. In particular, it ignores syntactic structure and word order by treating each sentence as an unordered collection of embeddings. This simplification can produce misalignments of semantically distinct words, particularly in sentences with high lexical overlap but different meanings. Consequently, WMD may fail to separate pairs where subtle contextual or syntactic differences are crucial. This limitation is especially pronounced when using static word embeddings, which lack contextual sensitivity. Recent work, such as Structure Mover's Distance [218], addresses these issues by integrating sentence-level structure through BERT's self-attention matrix (SAM), thereby achieving substantially better performance on STS benchmarks compared to the original WMD approach.

Let x and y be two sentences viewed as unigram sequences $x^1 = (w_1, \ldots, w_n)$ and $y^1 = (w'_1, \ldots, w'_m)$ consisting of n and m words, respectively.

$$x^1 = (w_1, w_2, \ldots, w_n), \quad y^1 = (w_1^{'}, w_2^{'}, \ldots, w_m^{'}) \tag{7.1}$$

where w_i and $w_j^{'}$ indicate the i-th word in x^1 and the j-th word in y^1. We denote the embedding of each word w_i by a bold vector $\boldsymbol{w}_i \in \mathbb{R}^d$, where d is the embedding dimensionality. In the WMD setting, each word is assigned an equal weight, and the distance between two words is computed as the Euclidean distance between their embedding vectors. Accordingly, we construct a *transportation cost* matrix $C \in \mathbb{R}^d \times \mathbb{R}^d$, where each entry $C_{ij} = d(x_i^1, y_j^1)$ measures the cost of moving the i-th word in x^1 to the j-th word in y^1:

$$d(w_i, w_j^{'}) = \left\| \boldsymbol{w_i} - \boldsymbol{w_j^{'}} \right\|_2 \tag{7.2}$$

The WMD between the unigram sequences x^1 and y^1, considering the weight vectors $\mathbf{t}_{\mathbf{x}^1}$ and $\mathbf{t}_{\mathbf{y}^1}$, where $\mathbf{t}_{\mathbf{x}^1} \in \mathbb{R}_+^{|x^1|}$ represents non-negative weights assigned to each unigram in x^1, is formalized as follows:

$$
\begin{aligned}
WMD(x^1, y^1) &= \min_{T \in \mathbb{R}^{|x^1| \times |y^1|}} \langle C, T \rangle, \\
&= \underset{T \in \mathbb{R}^{|x^1| \times |y^1|}}{\text{minimize}} \sum_{i,j} C_{ij} T_{ij}, \qquad (7.3) \\
\text{subject to} \quad & \mathbf{T}_{ij} \geq 0, \quad \mathbf{T}\mathbf{1} = \mathbf{t}_{\mathbf{x}^1}, \quad \mathbf{T}^\top \mathbf{1} = \mathbf{t}_{\mathbf{y}^1}.
\end{aligned}
$$

where $\mathbf{T}^\top$ denotes the transpose of $\mathbf{T}$, and $\mathbf{1} \in \mathbb{R}^d$ is an all-ones vector. The matrix T corresponds to the *transportation flow*: each element T_{ij} specifies the proportion of the i-th unigram in x^1 that is "transported" to the j-th unigram in y^1.

Under this formulation, WMD is the minimum total cost required to transport the unigrams from x^1 to y^1, derived from the element-wise multiplication of the optimal transportation flow matrix T and the cost matrix C. Therefore, $WMD(x^1, y^1)$ can be interpreted as the minimum transportation cost that aligns the word distributions of x^1 and y^1, weighted by $\mathbf{t}_{\mathbf{x}^1}$ and $\mathbf{t}_{\mathbf{y}^1}$.

Advantage of Jensen–Shannon Divergence (JSD). Although WMD has an intuitive interpretation, it often results in suboptimal word alignments, leading to inferior performance in semantic textual similarity compared to more recent techniques such as [161]. A key reason is that WMD typically uses Euclidean distance as the transportation cost, which merges the weighting component with the dissimilarity metric in a single calculation. Concretely, Euclidean costs may underestimate the similarity of word pairs as low$_{<SWP>}$ even when their meanings are close$_{<SYN>}$ but their concreteness or importance differs substantially$_{<CI>}$. For instance, "noodles" and "pho" are semantically close$_{<SYN>}$ but can appear low$_{<SWP>}$ under Euclidean-based transport because their concreteness or frequency of usage in the corpus is different, causing WMD to assign an unexpectedly low similarity. This highlights the limitations of using Euclidean distance in capturing nuanced semantic relationships.

Figure 7.2 further contrasts JSD and Euclidean distance computed between the embedding vectors of "**noodles**", "**pho**" (a Vietnamese noodle soup dish), "**burger**", and "**pizza**". In this setting, Euclidean distance can rank "noodles" closer to "pizza" than to "pho" due to potentially effects tied to vector magnitudes, even though "noodles" and "pho" are semantically more aligned, which is reflected by a smaller JSD. Unlike Euclidean distance, which relies on the norm of vectors when estimating the similarity of word pairs, JSD operates by comparing the underlying distributions, making it a more robust indicator of semantic discrepancy in contexts when embeddings are viewed in probabilistic representations.

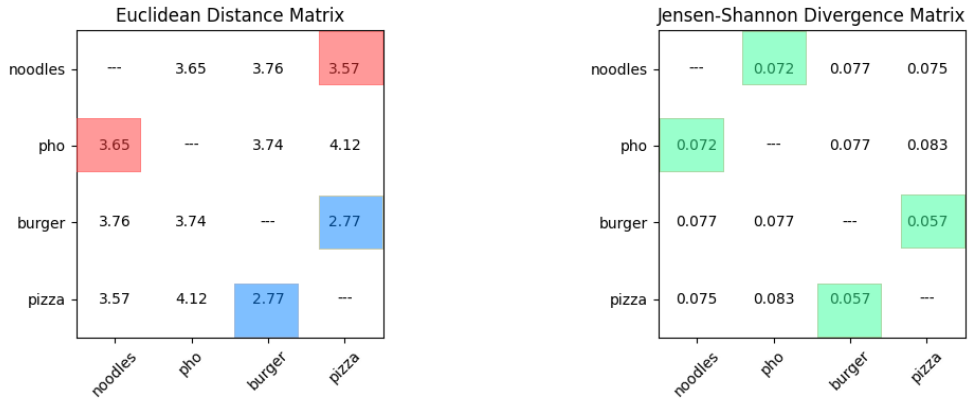


Figure 7.2: Comparative analysis of **JSD** and **Euclidean distance** for word embeddings generated by word2vec [1]. In the Euclidean distance matrix, the **lowest** value, representing the correct similarity word, is marked in blue for each row, while **inappropriate** alignments are highlighted in red. Conversely, in the JSD matrix, the **lowest** value, indicating the correct similarity word, is highlighted in green for each row.

Integration with BM25 Weighting and Contextual Embeddings. Considering these factors, we introduce a robust yet efficient sentence-level similarity measure, drawing inspiration from the Word Mover's Distance method. While traditional WMD leverages static embeddings such as word2vec [1] to map words into continuous vector spaces, these representations are inherently limited because they cannot adjust word meaning to the surrounding context. To mitigate this limitation, our approach replaces static vectors with context-aware embeddings, leveraging dynamically contextualized representations from advanced language models such as BERT [207]. This contextualization allows the similarity computation to better reflect nuanced and compositional semantic relations in sentences, improving the accuracy and reliability of similarity evaluations in diverse linguistic scenarios. For the cost function, we utilize JSD to measure the distance between the embedding representations of *1*-gram sequences. JSD provides a principled, symmetric notion of distributional divergence by averaging the Kullback–Leibler (KL) divergence with respect to a midpoint distribution. When word embeddings are normalized into probability distributions (e.g., via softmax transformations), JSD captures subtle shifts in their distributional structure, potentially revealing nuanced semantic differences that simple geometric distances can overlook. Consequently, using JSD as the cost function yields a probabilistically grounded and semantically richer cost function that aligns

naturally with the nuanced expressiveness of **contextual embeddings**.

$$d(w_i, w'_j) = \frac{1}{2}\sum_{l=1}^{d} P(\boldsymbol{w}_{i_l}) \log\left(\frac{P(\boldsymbol{w}_{i_l})}{M_l}\right) + \frac{1}{2}\sum_{l=1}^{d} P(\boldsymbol{w}_{j'_l}) \log\left(\frac{P(\boldsymbol{w}_{j'_l})}{M_l}\right) \quad (7.4)$$

where M_l is the mixture distribution at dimension l:

$$M_l = \frac{1}{2}\left(P(\boldsymbol{w}_{i_l}) + P(\boldsymbol{w}_{j'_l})\right) \quad (7.5)$$

$$P(\boldsymbol{w}_i) = \frac{\exp(\mathbf{z}_i)}{\sum_{l=1}^{d}\exp(\mathbf{z}_l)} \quad (7.6)$$

where $P(\boldsymbol{w}_i)$ denotes the probability distribution associated with word i in x^1. In Equation 7.6, $\mathbf{z}_l$ refers to the l-th component of the embedding vector for word i. The softmax function converts the embedding components into a normalized vector that $P(\boldsymbol{w}_i) = [p_1, p_2, \ldots, p_d]$ forms a valid probability distribution where each $p_l \geq 0$ and $\sum_{l=1}^{d} p_l = 1$.

$$\boldsymbol{w}_i = \text{BM25}(w_i) \cdot \boldsymbol{w}_i \quad (7.7)$$

where $\text{BM25}(w_i)$ is the BM25 score of the word w_i within sentence x^1, computed with respect to the corpus $\mathcal{DB} = \{S_1, S_2, \ldots, S_N\}$. Here, $\mathbf{w_i}$ is the embedding vector of word i. Accordingly, the weight assigned to the 1-gram x_i^1 is:

$$t_{x_i^1} = \text{BM25}(w_i) \quad (7.8)$$

$$\text{BM25}(w_i) = \sum_{i=1}^{N} \text{IDF}(w_i) \cdot \frac{f(w_i, S_i) \cdot (k_1 + 1)}{f(w_i, S_i) + k_1 \cdot \left(1 - b + b \cdot \frac{|S_i|}{\text{avgdl}}\right)} \quad (7.9)$$

where $f(w_i, S_i)$ is the frequency of w_i in sentence S_i, $|S_i|$ is the sentence length of S_i in words, and avgdl is the average sentence length in the corpus $\mathcal{C}$. The corpus contains N ssentences and $n(w_i)$ indicates the number of sentences that include w_i. We adopt the standard parameter setting $k_1 = 1.2$ and $b = 0.75$. The inverse document frequency (IDF) is defined as:

$$\mathrm{IDF}(w_i) = \ln\left(\frac{N - n(w_i) + 0.5}{n(w_i) + 0.5}\right) \tag{7.10}$$

Complexity. Two primary computational steps determine the time complexity of our chunk retrieval method. First, we evaluate the pairwise JSD distances between the query and each candidate chunk. Given a query of h_1 tokens and a candidate chunk of h_2 tokens, where each token is encoded as a d-dimensional embedding, this step requires $O(h_1 h_2 d)$ operations. Second, we solve the optimal transport problem that matches the two token sets by minimizing the total JSD-based alignment cost. This minimum-cost flow computation has complexity $O((h_1 + h_2)^3 \log(h_1 + h_2))$. Therefore, the total time complexity is:

$$O(h_1 h_2 d) \; + \; O\big((h_1{+}h_2)^3 \log(h_1{+}h_2)\big). \tag{7.11}$$

Because our chunk retriever computes exact optimal transport by solving a minimum-cost flow problem between the query and each candidate chunk, its runtime becomes expensive: the computation scales cubically with the number of tokens in the compared texts. As a result, directly applying this method to very large corpora is not practical. In this thesis, we retain the exact optimal transport formulation rather than adopting faster approximations to prioritize accuracy over speed. To keep inference time within a reasonable range, we mitigate the cost by segmenting long Wikipedia documents into shorter passages, with each chunk limited to 512 tokens. This preprocessing step enhances computational feasibility while preserving robust performancer for text similarity estimation.

7.3.2.2 Knowledge Graph Retrieval

In addition to retrieving textual evidence, we introduce a knowledge graph retrieval module to strengthen the system's reasoning ability. The core purpose of this module is to integrate external structured knowledge into the question answering pipeline through a set of graph-oriented retrieval and ranking steps. Concretely, the workflow includes four main stages: (i) extracting and aligning entities, (ii) retrieving relational triplets, (iii) transforming the retrieved knowledge into a passage-style format, and (iv) pruning irrelevant information. Figure 7.3 provides an overview of the overall procedure. Each stage plays a role in steadily narrowing the gap between natural language questions and structured relational paths needed for reliable answer prediction.

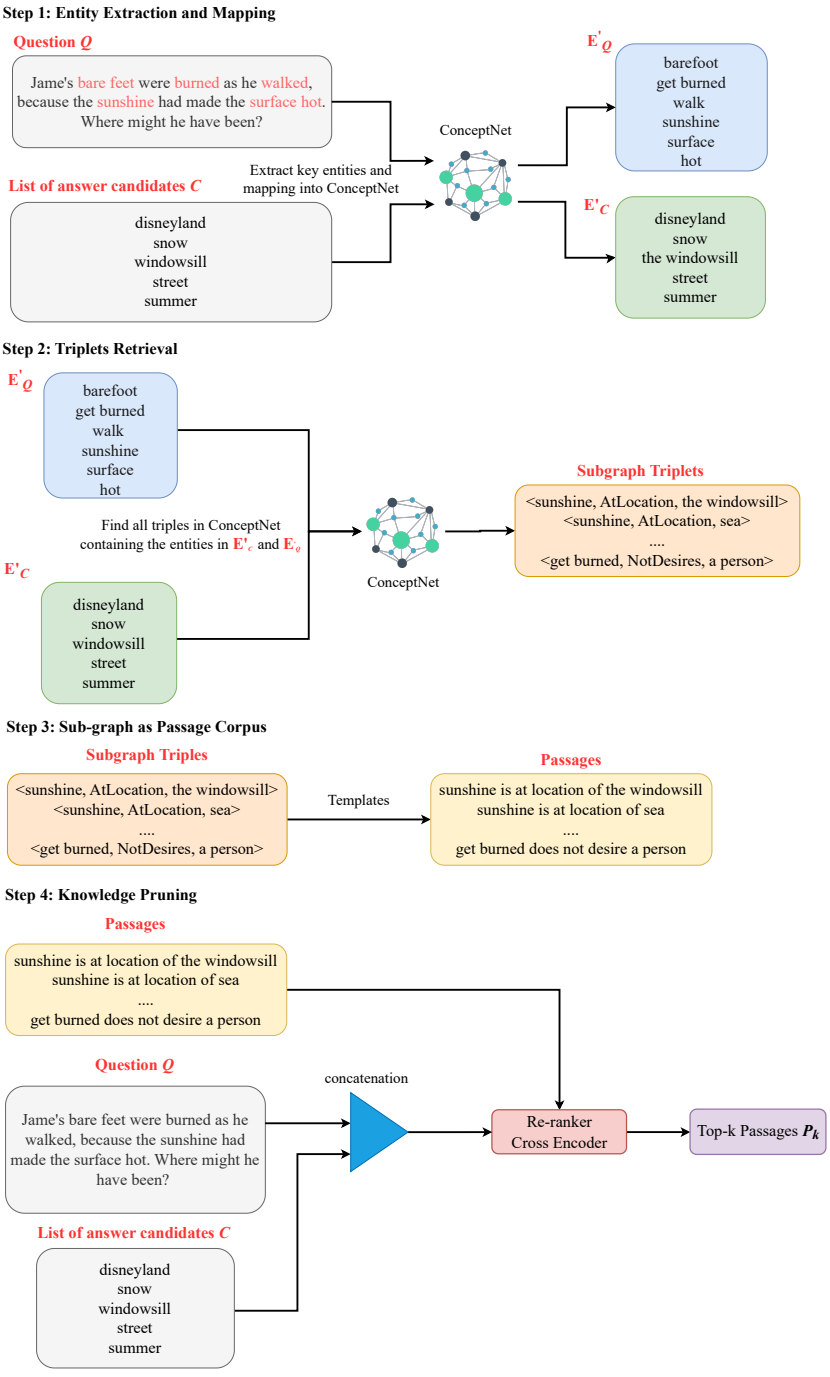


Figure 7.3: An illustration of the knowledge graph retrieval stage.

External Knowledge Graph We define the external knowledge source as a graph $G = (V, E)$, where V is the set of entities and E denotes directed relations among them. In this thesis, we use ConceptNet[2], a large-scale commonsense knowledge graph with more than 799k nodes and around 2.5 million edges.

In ConceptNet, each fact is represented as a triple (h, r, t), where h and t correspond to the head and tail entities, and r specifies the relation type. For instance, `(a window, AtLocation, any house)` expresses a location-oriented relationship between two physical concepts.

Entity Extraction and Mapping Given a natural language question q with its candidate answer set C, we first identify salient concepts using KeyBERT [193], a transformer-based approach for keyword extraction. This step produces two entity sets: E_q, which captures the main concepts from the question, and E_C, which contains representative terms extracted from the answer options.

Next, we align entities by linking each element in E_q and E_C to the corresponding entries in ConceptNet. This linking procedure can rely on exact lexical matching as well as fuzzy matching to increase coverage when surface forms differ. After alignment, we obtain the refined mapped sets: E'_q, the subset of question-side entities successfully linked to the graph, and E'_C, the set of mapped entities derived from the candidate answers.

Triplets Retrieval At this stage, given two entity lists E'_q and E'_C, we collect all ConceptNet triples that are connected to any entity appearing in either list. We define the retrieved triple set as $T = \{(e, r, e'), (e', r', e) | e \in E'_q \cup E'_C\}$. Concretely, for every entity $e \in E'_q \cup E'_C$, we construct its 1-hop neighborhood by collecting all triples where e appears as the subject or the object. For instance, when e is `sunshine`, this procedure can return triples such as (`sunshine`, `AtLocation`, `sea`) or (`sunshine`, `Causes`, `warmth`). By restricting retrieval to a 1-hop subgraph, we keep the most immediate relational context for each entity, which provides useful coverage while minimizing irrelevant expansion.

Sub-graph as Passage Corpus The retrieved subgraph T is then treated as a collection of textual "passages" P, where each element $p \in P$ corresponds to a triple with head entity h, relation r, and tail entity t. To make these structured triples usable in text-based reasoning, we convert each triple into a natural-language sentence using predefined templates as presented in Table

[2] https://home.ttic.edu/~kgimpel/commonsense.html

7.1. Formally, for each relation r, we first associate it with a phrase r_p, and then build a sentence d by concatenating h, r_p, and t in that order. For example, `<sunshine, AtLocation, the windowsill>` is verbalized as "sunshine is at location of the windowsill". After this verbalization step, the graph-structured set T becomes a textual corpus D, thereby facilitating downstream retrieval and reasoning.

Knowledge Pruning To suppress irrelevant content and retain only useful evidence, we introduce a pruning step that removes sentences whose relevance to the question–answer context is low. Concretely, for each sentence $d \in D$, we concatenate the questions and options: $s_i = q \oplus c_i$ and the sentence d as an input of the pretrained cross-encoder model [29]. The model produces a semantic relevance score that indicates how well the sentence aligns with the corresponding question–option pair. We then use these scores to re-rank all candidate sentences and select the top s sentences with the highest similarity values. This filtering process ensures that only the most informative content from the graph is retained, which both reduces retrieval overhead and helps the subsequent reasoning stage concentrate on the most relevant content.

7.3.3 Stage 2: Multiple Reasoning Chains

For every answer option $c_i \in C$, we generate a corresponding reasoning chain, denoted as rc_i. The construction of rc_i is conditioned on the question q, the specific candidate c_i, and the top-ranked textual passages with the graph triples returned by the hybrid retrieval module. Conceptually, each reasoning chain provides a step-by-step inference path that link q to c_i by integrating evidence from both retrieved text and the knowledge graph.

To obtain these chains, we provide a language model with a structured input that includes q, c_i, and the retrieved evidence. The model then outputs an explanation rc_i that supports the plausibility of candidate c_i. These explanations typically contain intermediate sub-facts, commonsense elements, and explicit logical transitions, making the inference process more interpretable. In contrast to standard pipelines that condense reasoning into only a final answer selection, our approach produces one reasoning chain per candidate. This design enables the subsequent decision stage to compare candidates using the plausibility, coherence, and factual support of their respective explanations.

We denote the complete set of reasoning chains as $RC = \{rc_1, rc_2, \ldots, rc_n\}$, where each rc_i is associated with candidate c_i. To systematically elicit these explanations, we propose prompts that explicitly instruct the model to reason step-by-step as shown below:

Table 7.1: Relation templates for ConceptNet are used in USCRaKE. We adapt these templates from [2]

Relation Type	**Template Phrase**
Antonym	is the antonym of
AtLocation	is at location of
CapableOf	is capable of
Causes	causes
CreatedBy	is created by
IsA	is a kind of
Desires	desires
HasSubevent	has subevent
PartOf	is part of
HasContext	has context
HasProperty	has property
MadeOf	is made of
NotCapableOf	is not capable of
NotDesires	does not desire
ReceivesAction	is
RelatedTo	is related to
UsedFor	is used for
LocatedNear	is located near
CausesDesire	causes the desire of
MotivatedByGoal	is motivated by the goal of
DistinctFrom	is distinct from
HasFirstSubevent	has the first subevent
HasLastSubevent	has the last subevent
HasPrerequisite	has the prerequisite of
Entails	entails
MannerOf	a manner of
InstanceOf	an instance of
DefinedAs	is defined as
HasA	has a
SimilarTo	is similar to
Synonym	is the synonym of

```
You are a helpful reasoning assistant. Your task is to explain
    whether a given candidate answer is likely to be correct by
    reasoning step-by-step from the question and supporting evidence
```

```
    .

You are given:
- A question: {question}
- A candidate answer: {candidate}
- Retrieved evidence: {evidence}

Using the information above, generate a step-by-step chain of
    reasoning that links the question to the candidate. The
    explanation should integrate relevant facts from both textual
    and graph-based evidence and show how they relate to the
    candidate answer.

Guidelines:
- Focus only on building the reasoning chain from question to
    candidate.
- Do not state whether the candidate is correct or incorrect.
- Make sure each explanation is logically complete and includes all
    key facts from the question and evidence.

Begin your reasoning below:
```

To illustrate Stage 2 in our pipeline, we provide a concrete example in which the LLM is asked to select the most plausible option by assessing the quality of evidence-grounded reasoning rather than relying on answer frequency. Given the question “Which vitamin is supplied from only animal source?" and four candidate vitamins (C, B12, D, K), the prompt supplies a compact evidence set containing both supportive and potentially distracting statements. In particular, the evidences emphasize that vitamin B12 occurs naturally in animal-derived foods and is not reliably present in plants unless fortified, while an additional Wikipedia-style statement about vitamin A functions as a distractor by being animal-related but not satisfying the “only animal source" constraint. The evaluator must therefore judge logical coherence and factual grounding across the provided chains to output the best-supported answer choice.

```
You are a helpful reasoning assistant. Your task is to explain
    whether a given candidate answer is likely to be correct by
    reasoning step-by-step from the question and supporting evidence
    .

You are given:
- A question: Which vitamin is supplied from only animal source?
```

```
- A candidate answer: Vitamin B12
- Retrieved evidence:
[E1] Vitamin B12 is naturally present in foods of animal origin such
     as meat, fish, eggs, milk.
[E2] Plants do not naturally contain reliable amounts of vitamin B12
    ; plant-based foods provide B12 mainly when fortified or
    contaminated by microorganisms.
[E3] Vitamin B12 is synthesized by microorganisms and accumulates in
     animal tissues through the food chain, making animal-derived
    foods the primary natural dietary source.
[E4] Vitamin B12 is related to animal products.
[E5] Vitamin A is found in animal-derived foods such as liver, fish
    oils, egg yolk, and dairy products, and it can also be obtained
    from plant sources as provitamin A carotenoids

Using the information above, generate a step-by-step chain of
    reasoning that links the question to the candidate. The
    explanation should integrate relevant facts from both textual
    and graph-based evidence and show how they relate to the
    candidate answer.

Guidelines:
- Focus only on building the reasoning chain from question to
    candidate.
- Do not state whether the candidate is correct or incorrect.
- Make sure each explanation is logically complete and includes all
    key facts from the question and evidence.

Begin your reasoning below:
```

7.3.4 Stage 3: Answer Generator

Given the reasoning chains produced for each candidate, the final decision stage selects the most plausible option by comparing the overall explanatory strength across these chains. Concretely, each chain is written as a step-by-step explanation that links the input question q to a particular candidate c_i, using both retrieved textual and graph-based evidence.

We concatenate all chains into a single structured prompt, where each line corresponds to the explanation associated with one candidate. The model is then instructed to solve the question step by step so that it explicitly examines and contrasts the reasoning paths. Based on this comparison, it determines which candidate is supported by the strongest rationale and uses

that rationale to produce the final answer prediction.

Formally, we define the Answer Generator as a function f_{AG} that takes the question q, the candidate set C, and the corresponding collection of reasoning chains $RC = \{rc_1, rc_2, \ldots, rc_n\}$, and returns a predicted answer c^*:

$$c^* = f_{\text{AG}}(q, C, RC) \tag{7.12}$$

To reliably elicit the final prediction, we use the following prompt template, which encourages introspective reasoning grounded in the provided chains:

```
You are a reasoning evaluator. Your task is to analyze multiple
    reasoning chains generated for each candidate answer to a given
    question and select the most plausible answer based on logical
    support.

You are given:
- A multiple-choice question: {question}
- A set of candidate answers: {candidate_list}
- A collection of reasoning chains for each candidate. Each chain is
     a step-by-step explanation connecting the question to that
    candidate using retrieved evidence.

Below are the reasoning chains:

{reasoning_chains_formatted}

Your task:
1. Carefully read the reasoning chains for each candidate.
2. Identify which candidate answer is most strongly supported across
     its chains.
3. Compare the logical coherence, factual grounding, and
    informativeness of the chains.
4. Choose the best answer.

Guidelines:
- Do not rely on the majority count. Based on your choice about the
    quality and consistency of reasoning.
- Focus on factual accuracy, logical flow, and connection to the
    original question.

Your final output should be in the following format:

Answer: <final_answer_choice>
```

To illustrate the Answer Generate stage in our pipeline, we provide a concrete example in which the model is asked to select the final answer by comparing several candidate-specific reasoning chains rather than answering the question directly. The question asks which vitamin is supplied only from animal sources, and the supporting evidence consistently characterizes vitamin B12 as naturally present in animal-derived foods while not being reliably available in plants unless through fortification or microbial contamination. Accordingly, the reasoning chains for option B (Vitamin B12) are both better grounded and more logically coherent than those for the remaining options, which are not supported by any explicit statements in the evidence set. This example highlights how the evaluator prioritizes factual grounding and internal consistency of reasoning, reducing the risk of choosing an option based on surface plausibility or majority voting.

```
You are a reasoning evaluator. Your task is to analyze multiple
    reasoning chains generated for each candidate answer to a given
    question and select the most plausible answer based on logical
    support.

You are given:
- A multiple-choice question: Which vitamin is supplied from only
    animal source?
- A set of candidate answers:
  A) Vitamin C
  B) Vitamin B12
  C) Vitamin D
  D) Vitamin K
- A collection of reasoning chains for each candidate. Each chain is
     a step-by-step explanation connecting the question to that
    candidate using retrieved evidence.

Below are the reasoning chains:

Chain 1
1) The question asks which vitamin is supplied only from an animal
    source.
2) Evidence states that vitamin B12 is naturally present in foods of
     animal origin (meat, fish, eggs, milk).
3) Evidence also states that plants do not naturally contain
    reliable amounts of B12; plant-based foods provide B12 mainly
    when fortified or contaminated by microorganisms.
4) Therefore, among the options, vitamin B12 best matches the
    criterion of being supplied only from animal sources (naturally)
```

```
    .

Chain 2
1) To determine a vitamin supplied only from animal sources, we
    check whether it is naturally available in plants.
2) Vitamin B12 is synthesized by microorganisms and accumulates in
    animal tissues through the food chain, making animal-derived
    foods the primary natural dietary source.
3) Plants do not naturally contain reliable amounts of B12.
4) Thus, vitamin B12 is the most strongly supported answer for "only
     animal source" in a natural dietary sense.

Chain 3
1) The key requirement is exclusivity to animal-derived foods.
2) Evidence explicitly links vitamin B12 to animal products.
3) Additionally, evidence lists animal-origin foods as natural
    sources of B12.
4) Since the other options are not supported as "only animal source"
     by the provided evidence, vitamin B12 is the best-supported
    choice.

Chain 4
1) The question asks which vitamin is supplied only from animal
    source.
2) The provided evidence discusses vitamin B12 and vitamin A, but
    does not mention vitamin C as being supplied only from animal
    sources.
3) Since there is no passage support for vitamin C in the evidence
    set, this candidate is weakly grounded compared with vitamin B12
    . (No supporting evidence for A)

Chain 5
1) The question asks which vitamin is supplied only from animal
    source.
2) The evidence set provides support that vitamin B12 is naturally
    present in animal foods and not reliably present in plants.
3) The evidence set does not provide any statements about vitamin D
    or vitamin K sources, nor does it claim either is only from
    animal sources.
4) Therefore, vitamin D and vitamin K are not supported by the given
     evidence, making them less plausible than vitamin B12. (No
    supporting evidence for C/D)
```

```
Your task:
1. Carefully read the reasoning chains for each candidate.
2. Identify which candidate answer is most strongly supported across
     its chains.
3. Compare the logical coherence, factual grounding, and
    informativeness of the chains.
4. Choose the best answer.

Guidelines:
- Do not rely on the majority count. Based on your choice about the
    quality and consistency of reasoning.
- Focus on factual accuracy, logical flow, and connection to the
    original question.

Your final output should be in the following format:

Answer: <final_answer_choice>
```

7.4 Experiment Settings

7.4.1 Implementation Details

To determine the final configuration, we search the number of retained chunks (top-k) and the number of retained triples (top-s) over the following candidate ranges:

- $k \in \{5, 10, 15, 20\}$
- $s \in \{5, 10, 15, 20\}$

Across the explored settings, we observe that $k = 15$ and $s = 10$ deliver the best overall performance, and we therefore adopt these values in all reported results. We encode chunks with the BGE model [126] to obtain 1024-dimensional semantic embeddings and index them using FAISS [216] to support efficient retrieval. All experiments are conducted on an NVIDIA A100 GPU with 40GB memory. For the answer generation component, we use open-source language models as the backbone of the framework, specifically Llama 3.3-70B-Instruct [127] (referred to as Llama3.3-70B), as well as Qwen2.5-14B-Instruct and Qwen2.5-72B-Instruct [128] (denoted as Qwen2.5-14B and Qwen2.5-72B, respectively), all running with 4-bit quantization [129]. We set the decoding temperature to 0 to keep the generation deterministic. Finally, for a fair and consistent comparison with prior state-of-the-art methods, including from QA-GNN [2], DRAGON [219],

and GREASELM [220] to MixLoRA [221], which report results exclusively in terms of accuracy, we also use accuracy (%) as the evaluation metric.

7.4.2 Data Preparation

Table 7.2 presents the evaluation benchmarks used in our experiments. In particular, we evaluate our proposed framework on six diverse multiple-choice QA datasets, covering both general-domain and specialized settings: CommonsenseQA (CSQA) [222], OpenBookQA (OBQA) [223], RiddleSense (Riddle) [224], Physical Interaction QA (PIQA) [225], MedMCQA [195], and ARC_C [226].

Table 7.2: Overview of dataset statistics. The symbol "–" denotes dataset splits that were either unavailable or not used in our experiments.

Task	**Dev**	**Test**
CommonsenQA official split	1,221	-
OpenBookQA	-	500
RiddleSenseQA	1,021	-
PIQA	1,838	-
MedMCQA	4,183	-
ARC_C	-	1,172

CommonsenseQA (CSQA) CommonsenseQA is a multiple-choice question answering benchmark centered on general commonsense reasoning, and it is constructed using ConceptNet [163]. The dataset contains 12,247 questions, each accompanied by five answer options. Since the CommonsenseQA test split is not publicly released, we follow [222] and run all comparisons with baseline methods as well as ablation analyses on the development set of the official split.

OpenBookQA (OBQA) OpenBookQA is a four-choice multiple-choice QA task that targets reasoning with elementary-level science knowledge. It includes 5,957 questions, and we report the accuracy of our system on the official test set provided by [223].

RiddleSense (Riddle) RiddleSense is a dataset designed to measure higher-level commonsense reasoning through riddle-style questions. This benchmark introduces a a distinctive multiple-choice question-answering challenge that requires understanding figurative expressions, counterfactual reasoning, and other advanced natural language understanding capabilities. Because the RiddleSense test set is not publicly available, we conduct evaluation on its development set as released in [224].

Physical Interaction QA (PIQA) The PIQA dataset is designed to address the challenging task of reasoning about physical commonsense in natural language. As PIQA does not release its test set, we use the validation split from [225], which contains 1,838 two-choice questions focused on physical intuition.

MedMCQA MedMCQA is a benchmark for answering real-world medical entrance examination questions, covering a wide range of medical topics drawn from the AIIMS & NEET PG entrance exams. In this work, since ground-truth labels for the test set are not provided, we adopt the development split from the original MedMCQA, consisting of 4,183 medical questions, for evaluating our framework.

AI2's Reasoning Challenge (ARC) AI2's Reasoning Challenge (ARC) evaluates reasoning on science multiple-choice questions spanning grades 3 through 9. ARC is organized into two parts, Easy and Challenge, where the Challenge set contains harder problems that demand stronger reasoning. The ARC_C subset follows a four-option format and includes 2,590 questions collected from standardized science examinations. We use the official test set introduced in [226].

7.4.3 Method Selected for Comparison

To assess the empirical effectiveness of our framework, we include several state-of-the-art (SOTA) reasoning QA systems (sorted by the timeline below).

- **QA-GNN** [2]: an end-to-end question answering approach that combines pretrained language models with knowledge graphs to strengthen multi-hop reasoning. It builds a unified graph that links the QA context with KG entities, and then applies graph neural networks to perform joint reasoning over both sources.
- **DRAGON** [219]: a self-supervised pretraining framework that learns a unified language–knowledge representation by integrating textual corpora with knowledge graphs.
- **GREASELM** [220]: a hybrid model that integrates pretrained language models with graph neural networks, aiming to enhance question answering.
- **GrapeQA** [227]: a framework for commonsense question-answering that integrates pretrained language models with knowledge graphs .
- **Med-PaLM 2** [198]: introduces Med-PaLM 2, an advanced large language model developed specifically to improve medical question answering performance.

- **MEDITRON 70B** [197]: Released in November 2023, MEDITRON is an open-source, monolingual biomedical large language model obtained by further pre-training LLaMA 2 on an additional 45 billion English tokens to boost domain specialization. It provides two parameter scales (7B and 70B); however, to keep the comparison aligned with prior biomedical baselines, we mainly report results for the 7B variant in this study.
- **MVP-Tuning** [228]: a retrieval-augmented tuning strategy that leverages similar question–answer pairs from the training data and uses a single prompt-tuned pretrained language model to fuse retrieved knowledge with the input text.
- **GPT-4** [3]: The authors introduced GPT-4, a large-scale multimodal model that can accept both text and image inputs while producing text outputs as its primary response.
- **PESC** [229]: The authors propose Parameter-Efficient Sparsity Crafting (PESC), which converts dense models into sparse ones by exploiting a Mixture of Experts architecture.
- **G-SAP** [230]: A framework that improves commonsense question answering by incorporating graph-structured signals into prompt learning.
- **MoSLoRA** [231]: Authors proposed the Mixture-of-Subspaces Low-Rank Adaptation (MoSLoRA) framework, which strengthens commonsense reasoning by decomposing model weights into multiple low-rank subspaces and learning an optimal combination of these subspaces.
- **Graph Reasoning for Question Answering with Triplet Retrieval** [30]: Since the paper does not introduce an acronym, we propose **'GRQATR'** (Graph Reasoning for Question Answering with Triplet Retrieval) in this study, as it captures the central idea of retrieving relevant knowledge graph triplets to support graph-based question answering reasoning.
- **SHAKTI** [232]: SHAKTI presents a 2.5 billion parameter language model tailored for resource-limited settings (e.g., edge devices). Despite its compact size, it achieves competitive commonsense reasoning results according to the benchmarks reported by the authors.
- **FineMedLM-o1** [233]: This framework improves dialogue and deep reasoning ability by training on high-quality synthetic medical data and long-form reasoning traces via Supervised Fine-Tuning (SFT) and Direct Preference Optimization (DPO).
- **MedS**^{3} [234]: a self-improving medical language model proposed for comprehensive clinical use, with an emphasis on deep reasoning capability.
- **MixLoRA** [221]: the authors introduce MixLoRA, a parameter-

efficient mixture-of-experts design that inserts multiple LoRA experts into the feed-forward layers of a frozen dense pretrained model.

- **Hippocrates** [235]: Hippocrates applies LoRA during the Domain-Adaptive Pre-training stage to effectively integrate medical-domain knowledge while aiming to preserve the model's general-purpose reasoning ability.
- **ISP^2** [236]: the authors propose Iterative Summarization Pre-Prompting (ISP^2), a technique designed to enhance LLM reasoning, particularly in cases essential information is implicit or not directly stated.
- **ERA-CoT** [237]: This framework is designed to enhance LLMs' comprehension of context by identifying relationships between entities while also facilitating reasoning across various tasks using the Chain-of-Thought (CoT) approach.
- **ERA-CoT** [237]: a framework that improves LLM's comprehension of context by explicitly modeling relationships among entities and supports reasoning across tasks using Chain-of-Thought.
- **RE2** [238]: RE2 is a prompting strategy proposed to improve the reasoning performance of Large Language Models by targeting input-side understanding. In contrast to approaches that mainly induce reasoning in the output stage, RE2 improves comprehension by instructing the model to read and process the question twice before generating an answer.

Finally, due to significant time constraints and limited computational resources, we could not reproduce all baseline models. Instead, we reran three representative approaches: QA-GNN [2], GREASELM [220], and PESC [229] in parallel with our proposed framework, and we denote these reproduced scores using an asterisk $*$ in Section 7.5.1. For all other baselines, we directly use the officially reported numbers from their original papers; these results were produced on the same benchmark datasets that have the same number of questions as in our experiments. Hence, while the reported improvement percentages should be interpreted with this limitation, the comparison is still informative and provides a reasonably fair indication of the strengths of our proposed method.

7.5 Experiment Results

7.5.1 Main Results

7.5.1.1 Results on Commonsense Reasoning QA Datasets

We perform extensive experiments on several standard MCQA benchmarks to evaluate the effectiveness of our retrieval-augmented reasoning framework. Table 7.3 reports results on three representative commonsense reasoning datasets: OBQA, CSQA, and ARC_C. In contrast to prior studies that often rely on fine-tuned language models or simple retrieval augmentation, our approach combines chunk-level retrieval with triple-level retrieval, enabling more informative and context-sensitive inputs for inference. When paired with open-source large language models, this design achieve state-of-the-art results across several of these benchmarks.

Using Qwen2.5-72B, USCRaKE attains 95.4 accuracy on OBQA, improving upon the strongest baseline (MVP-Tuning) by 4.1 points, and reaches 92.79 on CSQA, outperforming G-SAP by 1.44 points. The advantage of our retrieval pipeline also extends to smaller models: with Qwen2.5-14B and Llama3.3-70B, we obtain OBQA accuracies of 92.2 and 94.8, respectively. While GPT-4 still provides the best result on ARC_C, our method with open-source LLMs remains strongly competitive, reducing the performance gap and obtaining 93.69 on this challenging benchmark.

Table 7.3: Comparison of model performance on the commonsense reasoning benchmarks OBQA, CSQA, and ARC_C. Accuracy scores for baseline models (reported from original publications) and our implementations using Qwen2.5 and Llama3.3 are reported. Bold highlights the best-performing results, and underlined values indicate the highest baseline performance. Results reproduced in our experiments are marked with $*$. The symbol "-" indicates results that were not reported in the SOTA. Blue values indicate improvements achieved by USCRAKE over the best baseline, whereas red values indicate a performance decrease of USCRAKE compared to that baseline.

model \ dataset	**OBQA**	**CSQA**	**ARC_C**
Our implementation			
USCRaKE (Qwen2.5-14B)	$92.20_{\uparrow 0.9}$	$85.53_{\downarrow 5.82}$	$91.49_{\downarrow 4.81}$
USCRaKE (Qwen2.5-72B)	$\mathbf{95.40}_{\uparrow 4.1}$	$\mathbf{92.79}_{\uparrow 1.44}$	$93.69_{\downarrow 2.61}$
USCRaKE (Llama3.3-70B)	$94.80_{\uparrow 2.9}$	$89.24_{\downarrow 2.11}$	$91.53_{\downarrow 4.77}$
Baseline (test)			
QA-GNN [2]	82.75 $*$	-	-
DRAGON [219]	72.00	-	-
GREASELM [220]	84.77 $*$	-	-
GrapeQA [227]	90.00	-	-
MVP-Tuning [228]	<u>91.30</u>	83.29	-
GRQATR [30]	74.93	-	-
GPT-4 [3]	-	-	**<u>96.30</u>**
PESC [229]	-	-	65.20
G-SAP [230]	84.52	<u>91.35</u>	-
MoSLoRA [231]	86.80	-	81.50
MixLoRA [221]	86.90	-	79.90
ISP^2 [236]	-	81.00	-
ERA-CoT [237]	-	83.20	-
RE2 [238]	-	73.38	84.47

The empirical results indicate that dual-source retrieval can substantially improve LLM performance without relying on proprietary models or supervised fine-tuning, thereby establishing strong open-source baselines for open-domain MCQA. Overall, the findings highlight that combining textual retrieval with knowledge-based retrieval is a key factor in advancing the state of open-domain multiple-choice question answering.

7.5.1.2 Results on Other QA Datasets

To assess robustness and adaptability, we further evaluate the method on three heterogeneous reasoning benchmarks: Riddle, MedMCQA, and PiQA, as summarized in Table 7.4. On Riddle, USCRaKE with Qwen2.5-72B reaches 85.01 accuracy, outperforming the best baseline (GRQATR) by more than 10 points. On PiQA, the same setting achieves 96.24, exceeding G-SAP by 5.27 points and setting a new high-water mark. In the biomedical setting, Llama3.3-70B obtains 74.69 on MedMCQA, surpassing Med-Palm 2 by 2.39 points, which evidences that the approach transfers effectively to specialized domains.

USCRaKE also remains strong with smaller backbones. In particular, Qwen2.5-14B obtains 78.82 on Riddle and 92.21 on PiQA, and in both cases it consistently surpasses all evaluated baselines. These improvements across both general and domain-specific tasks show that retrieval-augmented reasoning not only boosts absolute performance but also provides reliable, scalable MCQA with open-source models. We associate these gains with our hybrid retrieval mechanism, which effectively combines evidence from complementary sources to support accurate and interpretable reasoning on complex questions.

Table 7.4: Accuracy of various models across diverse reasoning tasks, including Riddle, MedMCQA, and PiQA datasets. Baseline results (reported from original publications) and improvements from our approaches using Qwen2.5 and Llama3.3 are shown, with bold indicating the best performance. Results reproduced by us are marked with $*$. The symbol "-" indicates results that were not reported in the SOTA. Blue values indicate performance improvements, while red values indicate performance decrease of USCRAKE compared to the best baseline.

model \ dataset	**Riddle**	**MedMCQA**	**PiQA**
Our implementation			
USCRaKE (Qwen2.5-14B)	78.82$_{\uparrow 3.89}$	65.34$_{\downarrow 5.96}$	92.21$_{\uparrow 1.24}$
USCRaKE (Qwen2.5-72B)	**85.01**$_{\uparrow 10.08}$	71.65$_{\downarrow 0.65}$	**96.24**$_{\uparrow 5.27}$
USCRaKE (Llama3.3-70B)	83.61$_{\uparrow 8.68}$	**74.69**$_{\uparrow 2.39}$	92.82$_{\uparrow 1.85}$
Baseline (test)			
DRAGON [219]	71.30	-	81.10
GREASELM [220]	64.88 $*$	-	78.22 $*$
Med-Palm 2 [198]	-	<u>72.30</u>	-
MEDITRON 70B [197]	-	66.00	-
MVP-Tuning [228]	64.54	-	78.94
GRQATR [30]	<u>74.93</u>	-	-
G-SAP [230]	-	-	<u>90.97</u>
MoSLoRA [231]	-	-	89.70
SHAKTI [232]	-	-	86.20
FineMedLM-o1 [233]	-	65.26	-
MedS^3 [234]	-	65.20	-
MixLoRA [221]	-	-	87.80
PESC [229]	-	-	82.09 $*$
Hippocrates [235]	-	54.30	

The consistent performance improvements observed across a range of datasets are explained by the synergistic structure of our dual-source retrieval pipeline. Specifically, the pipeline retrieves evidence from two levels in parallel: chunk-level textual passages and 1-hop knowledge graph triples. By integrating these complementary forms of evidence, our hybrid retrieval mechanism not only constructs richer reasoning chains but also enables open-source LLMs to select answers more accurately without relying on in-context learning tricks or fine-tuning.

In addition, the enhanced chunk retrieval module contributes significantly

to the system's accuracy. Rather than returning broadly relevant or weakly associated passages, it is better at identifying the most informative chunks that align with both the question and the answer choices, thereby increasing the relevance of the textual evidence presented to the model. When combined with structured knowledge from the graph, the model gains access to both contextual surface-level contexts and deeper background information. These design elements enhance the method's reliability, broaden its applicability across diverse question types, and improve the interpretability of its reasoning process.

7.5.2 Ablation Study

To analyze the individual contribution of each retrieval component, we conduct ablation studies focused on both the chunk retriever and the knowledge graph modules.

7.5.2.1 Ablation Study on CommonsenseQA

For the chunk retrieval module, we evaluate our JSD-based optimal transport method against three alternative measures: MoverScore, Euclidean distance, and cosine similarity, evaluated on the main commonsense reasoning datasets. As presented in Table 7.5, our retriever consistently achieves the highest accuracy across all evaluated datasets and model scales. For instance, under Qwen2.5-14B, the JSD-based method obtains 90.6 on OBQA, exceeding MoverScore by 2.2 points and Euclidean distance by 6.6 points. When using larger backbones, the gains increase further, reaching up to 7.2 points relative to the baseline similarity metrics. Overall, these findings indicate that using Jensen-Shannon Divergence to define the retrieval cost leads to more semantically aligned evidence for downstream reasoning and establishes a new retrieval-augmented QA benchmark.

Table 7.5: Accuracy of different chunk retrieval methods: our proposed Jensen-Shannon Divergence, MoverScore, Euclidean distance, and cosine similarity—on commonsense reasoning benchmarks (OBQA, CSQA, ARC_C) using different models (Qwen2.5-72B, Qwen2.5-14B, and Llama3.3-70B). Best results per dataset are highlighted in bold. Blue numbers denote performance improvement of each chunk retrieval compared to the metric with the lowest score, and top results are marked in bold.

model \ dataset		OBQA	CSQA	ARC_C
Qwen2.5-14B				
USCRaKE (	Chunk retrieval with **Jensen-Shannon Divergence**)	**90.60**$_{\uparrow 6.6}$	**83.88**$_{\uparrow 4.13}$	**90.41**$_{\uparrow 3.79}$
USCRaKE (	Chunk retrieval with *MoverScore* [239])	88.40$_{\uparrow 4.4}$	82.44$_{\uparrow 2.69}$	89.87$_{\uparrow 3.25}$
USCRaKE (	Chunk retrieval with *Euclidean distance*)	84.00	79.75	87.15$_{\uparrow 0.53}$
USCRaKE (	Chunk retrieval with *cosine similarity*)	86.20$_{\uparrow 2.2}$	82.21$_{\uparrow 2.46}$	86.62
Qwen2.5-72B				
USCRaKE (	Chunk retrieval with **Jensen-Shannon Divergence**)	**93.60**$_{\uparrow 6.4}$	**90.81**$_{\uparrow 4.45}$	**93.25**$_{\uparrow 3.16}$
USCRaKE (	Chunk retrieval with *MoverScore* [239])	90.60$_{\uparrow 3.4}$	88.12$_{\uparrow 1.76}$	92.67$_{\uparrow 2.96}$
USCRaKE (	Chunk retrieval with *Euclidean distance*)	87.20	86.36	89.71
USCRaKE (	Chunk retrieval with *cosine similarity*)	91.80$_{\uparrow 4.6}$	89.53$_{\uparrow 3.17}$	91.93$_{\uparrow 2.22}$
Llama3.3-70B				
USCRaKE (	Chunk retrieval with **Jensen-Shannon Divergence**)	**92.60**$_{\uparrow 7.2}$	**87.98**$_{\uparrow 5.19}$	**91.01**$_{\uparrow 4.5}$
USCRaKE (	Chunk retrieval with *MoverScore* [239])	90.20$_{\uparrow 5.2}$	86.03$_{\uparrow 3.24}$	90.52$_{\uparrow 4.01}$
USCRaKE (	Chunk retrieval with *Euclidean distance*)	85.00	82.79	86.51
USCRaKE (	Chunk retrieval with *cosine similarity*)	89.00$_{\uparrow 4}$	85.47$_{\uparrow 2.68}$	88.05$_{\uparrow 1.54}$

7.5.2.2 Ablation Study on other QA Datasets

To further evaluate the generalizability of our retrieval approach, we extend the ablation analysis beyond commonsense reasoning to three additional benchmarks: Riddle, MedMCQA, and PiQA. Table 7.6 provides a comprehensive comparison between our Jensen-Shannon divergence-based optimal transport retriever and three commonly used alternatives—MoverScore [239],

Euclidean distance, and cosine similarity—across multiple open-source large language models, namely Qwen2.5-14B, Qwen2.5-72B, and Llama3.3-70B.

Table 7.6: Accuracy comparison of various chunk retrieval strategies: our proposed Jensen-Shannon Divergence-based method, MoverScore, Euclidean distance, and cosine similarity—on three distinct reasoning benchmarks: Riddle, MedMCQA, and PiQA. Blue values denote performance improvement of each chunk retrieval compared to the metric with the lowest score, while the best results are highlighted in bold.

model \ dataset		**Riddle**	**MedMCQA**	**PiQA**
Qwen2.5-14B				
USCRaKE (	Chunk retrieval with **Jensen-Shannon Divergence**)	**76.90**$_{\uparrow 3.89}$	**62.10**$_{\uparrow 3.39}$	**90.32**$_{\uparrow 3.97}$
USCRaKE (	Chunk retrieval with *MoverScore* [239])	73.57$_{\uparrow 0.56}$	59.07$_{\uparrow 0.36}$	89.85$_{\uparrow 3.5}$
USCRaKE (	Chunk retrieval with *Euclidean distance*)	73.01	58.71	86.35
USCRaKE (	Chunk retrieval with *cosine similarity*)	74.21$_{\uparrow 1.2}$	59.54$_{\uparrow 0.83}$	87.33$_{\uparrow 0.98}$
Qwen2.5-72B				
USCRaKE (	Chunk retrieval with **Jensen-Shannon Divergence**)	**83.78**$_{\uparrow 4.49}$	**68.41**$_{\uparrow 4.19}$	**95.29**$_{\uparrow 6.16}$
USCRaKE (	Chunk retrieval with *MoverScore* [239])	82.34$_{\uparrow 3.05}$	66.31$_{\uparrow 2.09}$	93.78$_{\uparrow 4.65}$
USCRaKE (	Chunk retrieval with *Euclidean distance*)	79.29	64.22	89.13
USCRaKE (	Chunk retrieval with *cosine similarity*)	81.41$_{\uparrow 2.12}$	65.17$_{\uparrow 0.95}$	92.02$_{\uparrow 2.89}$
Llama3.3-70B				
USCRaKE (	Chunk retrieval with **Jensen-Shannon Divergence**)	**82.79**$_{\uparrow 4.41}$	**71.15**$_{\uparrow 5.32}$	**91.56**$_{\uparrow 4.55}$
USCRaKE (	Chunk retrieval with *MoverScore* [239])	80.65$_{\uparrow 2.27}$	68.81$_{\uparrow 2.98}$	89.34$_{\uparrow 2.33}$
USCRaKE (	Chunk retrieval with *Euclidean distance*)	78.38	65.83	87.01
USCRaKE (	Chunk retrieval with *cosine similarity*)	82.23$_{\uparrow 3.85}$	69.35$_{\uparrow 3.52}$	89.27$_{\uparrow 2.26}$

Across all datasets and model scales, our JSD-based retriever delivers the best accuracy among the compared retrieval strategies. Concretely, on Riddle, Qwen2.5-72B with our retriever reaches 83.78, yielding a +4.49 point gain over the strongest Euclidean distance baseline. The same configuration on PiQA achieves 95.29, exceeding Euclidean distance by 6.16 points. The advantage is also evident in the biomedical setting: Llama3.3-70B obtains

71.15 on MedMCQA, which is 5.32 points higher than the Euclidean baseline. Notably, the improvements are not limited to large models—using the smaller Qwen2.5-14B, our retriever still improves results on all three datasets, indicating that the scalability of the approach.

Overall, these findings show that our JSD-based retriever generalizes reliably across domains and question styles, and it clearly surpasses conventional distance-based retrieval. By more accurately capturing the semantic relationships between questions, answer choices, and candidate chunks, our strategy supplies open-source LLMs with higher-quality, more relevant contextual evidence—thereby enhancing performance in a variety of challenging reasoning scenarios. These consistent gains directly substantiate our core contribution: in unsupervised retrieval settings, the JSD-based approach better captures distributional mismatches between query and candidate chunks than standard similarity measures like cosine or Euclidean distance.

7.5.2.3 Analysis: Effect of Knowledge Graph retrieval

Table 7.7 presents the performance of our knowledge graph retrieval module evaluated on six diverse reasoning benchmarks using three open-source LLMs: Qwen2.5-14B, Qwen2.5-72B, and Llama3.3-70B. The results consistently demonstrate that integrating knowledge graph evidence leads to accuracy improvements across all datasets and all model sizes. Specifically, with Qwen2.5-72B, our method achieves the top performance on four of the six benchmarks: OBQA (91.0), CSQA (88.05), ARC_C (91.36), and PiQA (93.35). On Riddle and MedMCQA, Llama3.3-70B shows particularly strong results, attaining 80.84 and 69.98, respectively, and notably exceeds Qwen2.5-72B on MedMCQA by over five points. This suggests that the advantages of structured retrieval are preserved and even amplified as model capacity increases, particularly for tasks that require relational or multi-hop reasoning. Importantly, Qwen2.5-14B is also robust, reaching 88.0 on OBQA and 89.43 on PiQA, indicating that graph-derived context is beneficial even for moderate-sized models.

Consistent gains across domains—including general commonsense (CSQA, ARC_C), biomedical (MedMCQA), and abstract reasoning (Riddle) - demonstrate the versatility of our approach. By incorporating ConceptNet triples, the model gains access to explicit relational structures that text-only retrieval may overlook, thereby enriching the evidence available for answer selection. Overall, these findings confirm that structured knowledge graph retrieval significantly strengthens the reasoning capabilities of open-source LLMs, enabling them to perform more effectively and adaptably across a broad range of multiple-choice question-answering tasks. Moreover, the

results reinforce that such structured retrieval not only enhances contextual grounding but also empowers models to handle complex relational questions that would otherwise be inaccessible using text alone.

Table 7.7: Accuracy comparison of our knowledge graph retrieval framework across six diverse reasoning datasets (OBQA, CSQA, ARC_C, Riddle, MedMCQA, and PiQA) using different open-source LLMs (Qwen2.5-14B, Qwen2.5-72B, and Llama3.3-70B).

model \ dataset	**OBQA**	**CSQA**	**ARC_C**	**Riddle**	**MedMCQA**	**PiQA**
USCRaKE (Qwen2.5-14B)	88.00	81.92	88.56	74.15	59.12	89.43
USCRaKE (Qwen2.5-72B)	91.00	88.05	91.36	81.23	64.67	93.35
USCRaKE (Llama3.3-70B)	90.60	84.15	89.37	80.84	69.98	89.83

Our ablation results highlight the central role of retrieval in final performance. Compared to standard similarity measures, our enhanced chunk retriever consistently yields more relevant and better-aligned textual evidence, thereby improving the model's ability to perform accurate reasoning. Combined with structured triples from the knowledge graph, the dual-retrieval setup enables robust generalization across commonsense and domain-specific questions. These findings validate the importance of using a hybrid and semantically grounded retrieval design in improving reasoning quality.

7.5.3 Computational Cost

To assess the computational costs of our framework across different LLMs, this subsection reports the end-to-end runtime on six datasets (Table 7.8. A consistent pattern emerges: retrieval is the dominant cost, contributing for roughly 90% of total runtime. In contrast, knowledge-graph reasoning contributes about 8–10%, and the answer generator remains below 6%, even when using 70B-parameter models. Total runtime increases with dataset size, largely because the framework must compute optimal-transport distances with a JSD ground metric across a large retrieval pool. Although the three generators have different decoding throughput, their influence on end-to-end time is limited once retrieval is included. For instance, moving from a 14B model to a 70B model raises overall latency by only 5–8%, since generation constitutes only a small portion of the pipeline. Accordingly, the "Average Time" column—defined as the mean across generator settings—changes only modestly with model scale, because retrieval dominates the overall budget. These observations imply that meaningful acceleration depends mainly on retrieval-related hyperparameters, including the FAISS candidate pool size

R and the degree of parallelization used for chunk retrieval scoring. Finally, MedMCQA exhibits a much larger absolute runtime because it contains substantially more questions than the other benchmarks. However, after normalizing by minutes per question, the runtime is similar across datasets (16–19 min/Q), suggesting that the framework scales approximately linearly with dataset size rather than introducing disproportionate overhead.

Table 7.8: End-to-end runtime per dataset. Values are reported in hours (h), with minutes per question (min/Q) shown in the rightmost column.

Dataset	Chunk retrieval	Knowledge Graph Reasoning	Answer Generator			Average Time	min/Q
			Qwen2.5-14B	Llama3.3-70B	Qwen2.5-70B		
OBQA	132h	16h	1.4h	9.7h	8.3h	153h	17.36min
CSQA	289h	38h	3.4h	23.7h	20.4h	343h	16.8min
ARC_C	305h	36h	3.3h	22.8h	19.5h	360h	17.6min
Riddle	293h	33h	2.8h	19.8h	17.0h	339h	19.8min
MedMCQA	984h	74h	11.6h	81.4h	69.7h	1016h	16.2min
PiQA	480h	50h	5.1h	35.7h	30.7h	553h	18.6min

7.5.4 Effect of FAISS candidate-pool size R

We investigate how the coarse retrieval pool size R influences USCRaKE by evaluating 100 randomly sampled questions from OBQA and CSQA. Figure 7.4 shows that increasing the FAISS pool from **15k** to **20k** leads to steady improvements in accuracy on both datasets. On OBQA, accuracy increases from 86.19% to 87.51%, corresponding to an improvement of $+\mathbf{1.32}$ points. On CSQA, accuracy rises from 83.98% to 84.74%, corresponding to an improvement of $+\mathbf{0.76}$ points.

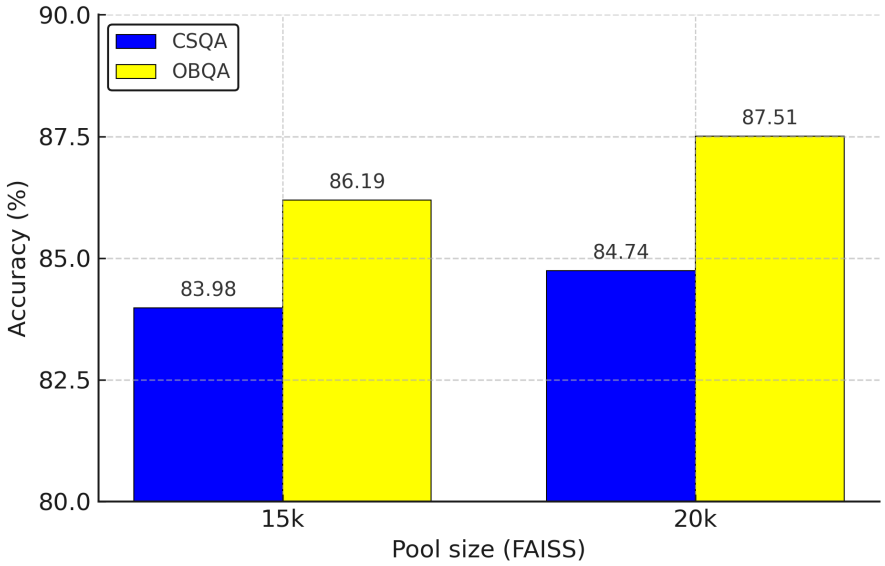


Figure 7.4: Comparative analysis of different pool size R on the two datasets CSQA and OBQA.

These improvements arise because a larger candidate pool increases the likelihood of capturing relevant evidence chunks, which our fine-grained semantic matcher can then promote into the final top-ranked set. In practice, this suggests that setting $\mathbf{R = 20k}$ offers a favorable choice when maximizing accuracy is the main objective, while acknowledging that it introduces additional retrieval overhead.

7.5.5 Analysis of top-k and top-s

Figure 7.5 presents the effect of varying the chunk budget k and the triple budget s on accuracy. On both OBQA and CSQA, performance improves steadily when k increasing from 5 to 15, but then drops slightly at $k = 20$, showing that larger chunk pools eventually introduce noise that offsets recall gains. A similar trend is observed for the triple budget: moving from $s = 5$ to $s = 10$ yields clear improvements, whereas higher values ($s \geq 15$) provide little benefit and may even reduce accuracy. The best results are consistently achieved with $k = 15$ and $s = 10$, where accuracy reaches 95.6% on OBQA and 92.8% on CSQA. Overall, these findings demonstrate that moderate retrieval budgets effectively balance comprehensive evidence coverage against noise suppression, while also mitigating the unnecessary computational overhead associated with excessively large retrieval candidates.

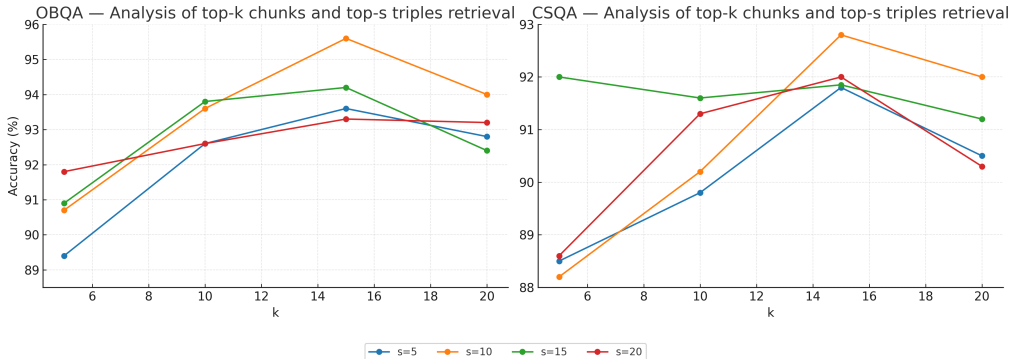


Figure 7.5: Comparative analysis with different top-k and top-s on the two datasets OBQA and CSQA.

7.6 Error Analysis

To better understand the limitations of our hybrid retrieval framework, we analyze two representative failure cases. These examples illustrate how abstract or metaphorically phrased queries can challenge both chunk-level and knowledge graph retrieval, especially when such queries lack dense lexical anchors or exhibit ambiguous semantics.

Question and Options	Chunk Retrieval	Triple Retrieval
Question: What do people aim to do at work? **Options**: A. complete job B. learn from each other C. kill animals D. wear hats E. talk to each other **Gold Answer**: complete job **Predicted Answer**: learn from each other	Wage labo usually referred to as paid work, paid employmen In exchange for the money paid as wages the work product generally becomes the undifferentiated property of the employer. The business's actions and decisions should be primarily ethical before it happens to become an ethical or even legal issue. Misuse of company's times and resources costs a company billions of dollars on a yearly basis. This misuse is from late arrivals, leaving early, long lunch breaks, inappropriate sick days. A person's personal code of ethics encompasses many different qualities such as integrity, honesty, communication, respect, compassion, and common goals. Among the many people management strategies that companies employ are a "soft" approach that regards employees as a source of creative energy and participants in workplace decision making. Many companies are assessing the environmental factors that can lead employees to engage in unethical conduct. A competitive business environment may call for unethical behavior	("worker", "CapableOf", "complete job") ("work", "MotivatedByGoal", "earn money") ("student", "CapableOf", "learn from teacher") ("teacher", "CapableOf", "educate students") ("people", "CapableOf", "talk to each other")

Figure 7.6: Error Analysis for Abstract Goal-Oriented Query

Figure 7.6 presents the system output for the question: *"What do people aim to do at work?"*. The gold answer is **complete job**, whereas the system predicts **learn from each other**. This question is highly abstract and goal-oriented, but it does not contain strong lexical indicators of the intended objective. Consequently, most of the top-ranked chunk passages emphasize general workplace activities (e.g., communication and teamwork) rather than the concrete goal of finishing tasks. Among the top twenty retrieved passages, only two explicitly mention completing a job, suggesting that the chunk retriever tends to prefer thematically related evidence even when it is not aligned with the question's core intent. A similar pattern appears in KG retrieval: although relevant triples are retrieved (e.g., *("worker", "CapableOf", "complete job")*), they are dominated by triples that stress social interaction or learning, such as *("student", "CapableOf", "learn from teacher")* and *("people", "CapableOf", "talk to each other")*. As a result, the retrieved evidence broadly matches the workplace setting but does not sufficiently emphasize the principal objective, revealing a limitation in intent alignment for abstract queries.

Question and Options	Chunk Retrieval	Triple Retrieval
Question: I saw a strange creature long, hard, and straight, thrusting into a round, dark opening preparing to discharge its load of lives puffing and squealing noises accompanied it, then a final screech as it slowed and stopped **Options**: A. eye B. space C. gas station D. finalist E. train **Gold Answer**: train **Predicted Answer**: space	Most anecdotal sightings of Bigfoot describe the creatures allegedly observed as solitary, although some reports have described groups being allegedly observed together. Many alleged sightings are reported to occur at night leading some cryptozoologists to hypothesize that Bigfoot may possess nocturnal tendencies. The bunyip has been described by natives as amphibious, nocturnal, reclusive, and inhabiting lakes, rivers, and swamps. Physical descriptions of bunyips vary widely, with some likening it to a starfish or a large aquatic predator, feeding on unsuspecting prey. Miners say that anyone can hear her knocking through the galleries just in the next corridor, even if there is no knowledge of that place in the rock. She is producing that strange sound to search for gold to show them, or to help them not get lost. Many descriptions of the amphisbaena say its eyes glow like candles or lightning, and it makes peculiar sounds as it moves. Some accounts depict the creature as snake-like, propelling itself forward through difficult terrain. Trilobites are always found with their heads directed towards the opening of the tube, suggesting they reversed in; the absence of any moulted carapaces suggests that moulting was not their primary reason for seeking shelter. The animals moved against the current, at the rate of about seven miles an hour.	("passenger", "PartOf", "train") ("hole", "RelatedTo", "space") ("dark", LocatedNear, "space") ("round", IsA, "planet") ("long", RelatedTo, "train")

Figure 7.7: Error Analysis for Figurative Metaphorical Query

The second example, shown in Figure 7.7, involves a metaphorically expressed question: *"I saw a strange creature long, hard, and straight, thrusting into a round, dark opening preparing to discharge its load of lives..."* The correct answer is **train**, but the system outputs **space**. In this case, the query relies heavily on figurative imagery rather than explicit reference terms, which makes retrieval especially susceptible to misleading overlaps. At the chunk level, the retriever returns passages about nocturnal creatures (e.g., Bigfoot), largely triggered by surface matches with words such as "creature", "dark", and "nocturnal", while failing to recover evidence related to transportation. The KG retriever exhibits the same tendency, producing triples such as *("hole", "RelatedTo", "space")* and *("dark", "LocatedNear", "space")*, driven by prominent but contextually unhelpful terms in the question. Although some relevant KG evidence exists (e.g., *("passenger", "PartOf", "train")*), it is too limited to shift the overall evidence distribution toward the correct interpretation. This failure illustrates how metaphorical phrasing, when paired with sparse lexical anchoring, can cause retrieval to

overweight coincidental associations instead of the intended concept.

In summary, these two cases underscore a recurring limitation in our hybrid retrieval system: for queries that are abstract, ambiguous, or figurative, both chunk-level and KG-based retrievers tend to prioritize surface-level similarity at the expense of task-specific intent. In the abstract case, the system conflates surrounding workplace context (e.g., teamwork and communication) with the latent goal (e.g., task completion). In the metaphorical case, retrieval is primarily driven by shallow word overlap, producing evidence that appears related at the term level but is pragmatically misaligned with the correct answer. These difficulties are closely tied to the distributional nature of such queries, including sparse lexical cues, broad expressions, and limited concrete descriptors. While the framework performs well on literal and well-grounded questions, it is less reliable when ambiguity and figurative language dominate the input. Without additional reranking or verification, the system cannot consistently disambiguate these cases. This analysis motivates future extensions toward goal-aware retrieval, pragmatic reasoning, and stronger verification strategies that better align retrieved evidence with the true intent of the query.

7.7 Conclusion

We present USCRaKE, a unified retrieval-and-reasoning framework designed to improve language model performance by combining fine-grained semantic chunk retrieval, knowledge graph reasoning, and structured answer generation. The main novelty is a fully unsupervised dual-evidence pipeline: Optimal Transport with Jensen-Shannon Divergence delivers precise text grounding, while verbalized knowledge graph triples provide essential structured cues. A key contribution is an unsupervised semantic retrieval module built on Optimal Transport, where the transport cost is defined via Jensen-Shannon Divergence within the Word Mover's Distance framework. This formulation supports semantically aligned evidence retrieval without requiring annotated supervision or training a dense retriever.

To further enhance retrieval quality, the framework integrates 1-hop subgraphs from ConceptNet to capture relational paths between question and context entities. Evidence from both chunk-based and graph-based sources is then re-ranked with a cross-encoder, which reduces noise and improves overall coherence. On top of this evidence, the reasoning component produces contrastive, candidate-specific rationales, explicitly encouraging the model to differentiate among answer choices. The final prediction is obtained by selecting and aggregating the most plausible and coherent explanation, which

improves factual accuracy and helps mitigate hallucinations.

Experimental results indicate that this hybrid combination of unsupervised semantic retrieval, structured knowledge graph reasoning, and contrastive explanation generation yields substantial gains in robustness, grounding, and interpretability for complex question answering tasks. Overall, USCRaKE offers a promising direction for retrieval-augmented language modeling in both general-domain and specialized-domain settings.

Despite these advantages, the framework remains less effective on questions that are highly abstract, ambiguous, or dependent on implicit knowledge. Future work could investigate retrieval mechanisms that are more sensitive to the underlying intent of the question, rather than relying primarily on surface-level similarity. Exploring approaches that incorporate pragmatic and contextual understanding, as well as mechanisms for verifying the factual consistency of retrieved evidence, could further enhance the reliability of the system.

Chapter Summary

This chapter presented USCRaKE, an unsupervised framework for evidence selection in retrieval-augmented reasoning, designed to reduce reliance on labeled data while improving the faithfulness of retrieved context. The core motivation is that many RAG pipelines still retrieve evidence using lexical matching or pointwise vector similarity, which can capture topical relatedness but often fails to preserve the distribution-level semantics needed for multi-hop inference. To address the limitation, USCRaKE formulates chunk retrieval as an Optimal Transport problem and uses Jensen–Shannon divergence as the ground cost, enabling retrieval that compares distributions of contextual tokens rather than a single pooled embedding. This design makes the retriever more sensitive to subtle semantic shifts and better suited for selecting precise supporting evidence in a fully unsupervised manner. The chapter also described the complete pipeline from chunk segmentation and embedding construction to Optimal Transport-based scoring and final ranking, emphasizing that the method can be integrated with open models without task-specific fine-tuning. Empirical results across general, biomedical, and scientific benchmarks showed that USCRaKE consistently improves evidence quality and downstream accuracy compared with conventional similarity-based retrieval. Beyond accuracy, the framework strengthens interpretability by making the retrieval decision more principled and easier to diagnose when errors occur. Overall, USCRaKE establishes a retrieval foundation for evidence-based reasoning in this thesis by prioritizing verifiable,

semantically aligned context selection.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

This dissertation develops a unified, *unsupervised*, and *explainable* framework for evidence-based reasoning across **text**, **knowledge graphs**, and **tables**. The core claim established in the dissertation is that trustworthy use of LLMs in scientific and biomedical settings requires not only fluent answers but also (i) *inspectable evidence selection* and (ii) *verifiable reasoning traces*. To meet this requirement without task-specific supervision, the thesis follows three design principles: (1) make model knowledge *explicit* when possible, (2) retrieve evidence precisely, and (3) produce reasoning artifacts that humans can *inspect and verify*.

Concretely, the framework contributes a *family of unsupervised evidence selectors* that can be plugged into different reasoning settings. For knowledge extraction, we cast tuple selection as an Optimal Transport view with cosine-based matching to retain contextually consistent relations. For text retrieval, we use Optimal Transport with Jensen–Shannon divergence to compare representations in a stable and distribution-aware manner. For graph reasoning, we select question-relevant multi-hop paths from a biomedical knowledge graph using unsupervised graph attention and then verbalize them as compact evidence for an LLM. For tables, we perform KG-free evidence compression through spherical k-means column selection and generate structured, counterfactual explanations with fine-grained citations. Together, these components support both question answering and fact verification while keeping decisions auditable.

The main contributions of this dissertation are summarized as follows:

- **Chapter 3: UCRET — Unsupervised Column Relevance Extraction for Table-Based Fact Verification.** We propose **UCRET**, a KG-free framework for table-based fact verification that performs unsupervised evidence compression via spherical k-means column selection. It forms a collapsed evidence table and produces label-conditioned explanations (SUPPORTS, REFUTES, NOT ENOUGH

INFO) with fine-grained, citation-grounded rationales.

- **Chapter 4: UCRET-JS — Unsupervised Column Relevance Extraction for Table-Based Question Answering.** We introduce **UCRET-JS**, which strengthens semantic column selection by replacing cosine-based similarity with a Jensen–Shannon-divergence alternative over contextual token distributions. This improves robustness for option-driven table question answering while preserving the same unsupervised and explainable design.
- **Chapter 5: K-Bloom — Unsupervised Knowledge Extraction.** We propose **K-Bloom**, which converts implicit knowledge in pretrained language models into an explicit and reusable knowledge graph by harvesting high-precision tuples. The method adopts an Optimal-Transport perspective with cosine-based matching to reduce noisy or inconsistent extractions and to produce an interpretable symbolic resource.
- **Chapter 6: UGAT-MedQA — Unsupervised Graph Attention for Biomedical Question Answering.** We develop **UGAT-MedQA**, which applies unsupervised graph attention to select question-relevant nodes and salient multi-hop paths from the extracted biomedical KG. The selected subgraph is verbalized into compact evidence chains to support step-by-step biomedical question answering without node-level supervision.
- **Chapter 7: USCRaKe — Unsupervised Semantic Retrieval for Text with Lightweight Knowledge Graph Evidence.** We present **USCRaKe**, an unsupervised retrieval-and-reasoning framework for multiple-choice question answering. It introduces semantic chunk retrieval based on Optimal Transport with Jensen–Shannon divergence and integrates lightweight one-hop ConceptNet evidence to support answer selection with interpretable rationales.

Overall, this thesis shows that evidence-based reasoning does not require heavy supervision to be practical. By focusing on unsupervised evidence selection and verifiable reasoning artifacts, the proposed framework improves transparency and supports reliable analysis in scientific and biomedical settings, where users must inspect both evidence and reasoning rather than only accept an answer.

8.2 Future Work

Although the proposed framework is fully unsupervised and modular, there remain several promising directions to improve reliability and coverage:

- **Unsupervised Knowledge Extraction.** Future work should reduce hallucinated or inconsistent tuples by adding stronger consistency checks during extraction. Another open challenge is how to evaluate knowledge coverage in a principled way. Since coverage is difficult to measure directly, future work can explore improved evaluation protocols using entity matching, contextual validation, and alignment with external knowledge bases when available.
- **Unsupervised Chunk Retrieval.** A key next step is to design retrieval strategies that better reflect the intent of the question, not only surface similarity. Incorporating lightweight verification signals for retrieved evidence could further improve reliability, especially in settings where retrieved context contains distractors or partially relevant information.
- **Unsupervised Graph Attention for Biomedical Reasoning.** Future work can increase KG coverage by integrating multiple complementary medical knowledge graphs and developing automatic KG updating methods to reduce missing links and outdated facts. The verbalized reasoning chains produced by our approach also provide a natural interface for future clinical decision support settings, where transparency and traceability are essential.
- **Unsupervised Table Evidence Selection.** For tables, future work can extend column selection to jointly identify informative rows (e.g., bidirectional selection over rows and columns) to better handle large or noisy tables. Additional improvements may include more robust table parsing and normalization to support heterogeneous real-world schemas and domain-specific formatting.

References

[1] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, 2013.

[2] M. Yasunaga, H. Ren, A. Bosselut, P. Liang, and J. Leskovec, "Qa-gnn: Reasoning with language models and knowledge graphs for question answering," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021, pp. 535–546.

[3] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[4] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann *et al.*, "Palm: Scaling language modeling with pathways," *Journal of Machine Learning Research*, vol. 24, no. 240, pp. 1–113, 2023.

[5] Z. Zhao, W. Fan, J. Li, Y. Liu, X. Mei, Y. Wang, Z. Wen, F. Wang, X. Zhao, J. Tang *et al.*, "Recommender systems in the era of large language models (llms)," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 11, pp. 6889–6907, 2024.

[6] J. Li, Y. Liu, W. Fan, X.-Y. Wei, H. Liu, J. Tang, and Q. Li, "Empowering molecule discovery for molecule-caption translation with large language models: A chatgpt perspective," *IEEE transactions on knowledge and data engineering*, vol. 36, no. 11, pp. 6071–6083, 2024.

[7] Y. Ding, Y. Ma, W. Fan, Y. Yao, T.-S. Chua, and Q. Li, "Fashionregen: Llm-empowered fashion report generation," in *Companion Proceedings of the ACM Web Conference 2024*, 2024, pp. 991–994.

[8] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong *et al.*, "A survey of large language models," *arXiv e-prints*, pp. arXiv–2303, 2023.

[9] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.

[10] S. Borgeaud, A. Mensch, J. Hoffmann, T. Cai, E. Rutherford, K. Millican, G. B. Van Den Driessche, J.-B. Lespiau, B. Damoc, A. Clark *et al.*, "Improving language models by retrieving from trillions of tokens," in *International conference on machine learning*. PMLR, 2022, pp. 2206–2240.

[11] O. Ram, Y. Levine, I. Dalmedigos, D. Muhlgay, A. Shashua, K. Leyton-Brown, and Y. Shoham, "In-context retrieval-augmented language models," *Transactions of the Association for Computational Linguistics*, vol. 11, pp. 1316–1331, 2023.

[12] F. Petroni, P. Lewis, A. Piktus, T. Rocktäschel, Y. Wu, A. H. Miller, and S. Riedel, "How context affects language models' factual predictions," *arXiv preprint arXiv:2005.04611*, 2020.

[13] K. Shi, X. Sun, Q. Li, and G. Xu, "Compressing long context for enhancing rag with amr-based concept distillation," *arXiv preprint arXiv:2405.03085*, 2024.

[14] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. Chang, "Retrieval augmented language model pre-training," in *International conference on machine learning*. PMLR, 2020, pp. 3929–3938.

[15] Q. He, G. Huang, Q. Cui, L. Li, and L. Liu, "Fast and accurate neural machine translation with translation memory," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 3170–3180.

[16] U. Khandelwal, O. Levy, D. Jurafsky, L. Zettlemoyer, and M. Lewis, "Generalization through memorization: Nearest neighbor language models," *arXiv preprint arXiv:1911.00172*, 2019.

[17] J. Xu, J. M. Crego, and J. Senellart, "Boosting neural machine translation with similar translations," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 1580–1590.

[18] S. Liu, W. Nie, C. Wang, J. Lu, Z. Qiao, L. Liu, J. Tang, C. Xiao, and A. Anandkumar, "Multi-modal molecule structure–text model for text-based retrieval and editing," *Nature Machine Intelligence*, vol. 5, no. 12, pp. 1447–1457, 2023.

[19] J. Wu, C.-C. Chang, T. Yu, Z. He, J. Wang, Y. Hou, and J. McAuley, "Coral: collaborative retrieval-augmented large language models improve long-tail recommendation," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 3391–3401.

[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "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)*, J. Burstein, C. Doran, and T. Solorio, Eds. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[21] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[22] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020.

[23] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier *et al.*, "Knowledge graphs," *ACM Computing Surveys (Csur)*, vol. 54, no. 4, pp. 1–37, 2021.

[24] Google, "Introducing the Knowledge Graph: things, not strings," https://blog.google/products/search/introducing-knowledge-graph-things-not/, 2012, [Online; accessed 16-May-2012].

[25] H. Wang, M. Zhao, X. Xie, W. Li, and M. Guo, "Knowledge graph convolutional networks for recommender systems," in *The world wide web conference*, 2019, pp. 3307–3313.

[26] X. Huang, J. Zhang, D. Li, and P. Li, "Knowledge graph embedding based question answering," in *Proceedings of the twelfth ACM international conference on web search and data mining*, 2019, pp. 105–113.

[27] W. Jin, B. Zhao, H. Yu, X. Tao, R. Yin, and G. Liu, "Improving embedded knowledge graph multi-hop question answering by introducing relational chain reasoning," *Data Mining and Knowledge Discovery*, vol. 37, no. 1, pp. 255–288, 2023.

[28] S. Hao, B. Tan, K. Tang, B. Ni, X. Shao, H. Zhang, E. Xing, and Z. Hu, "BertNet: Harvesting knowledge graphs with arbitrary relations from pretrained language models," in *Findings of the Association for Computational Linguistics: ACL 2023*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 5000–5015. [Online]. Available: https://aclanthology.org/2023.findings-acl.309

[29] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," 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)*, 2019, pp. 3982–3992.

[30] S. Li, Y. Gao, H. Jiang, Q. Yin, Z. Li, X. Yan, C. Zhang, and B. Yin, "Graph reasoning for question answering with triplet retrieval," in *Findings of the Association for Computational Linguistics: ACL 2023*, 2023, pp. 3366–3375.

[31] Y.-S. Chuang, W. Fang, S.-W. Li, W.-t. Yih, and J. Glass, "Expand, rerank, and retrieve: Query reranking for open-domain question answering," in *Findings of the Association for Computational Linguistics: ACL 2023*, 2023, pp. 12 131–12 147.

[32] A. Gunawardana and G. Shani, "A survey of accuracy evaluation metrics of recommendation tasks," *Journal of Machine Learning Research*, vol. 10, pp. 2935–2962, 2009.

[33] H. Steck, C. Ekanadham, and N. Kallus, "Is cosine-similarity of embeddings really about similarity?" in *Companion Proceedings of the ACM Web Conference 2024*, 2024, pp. 887–890.

[34] S. Robertson, H. Zaragoza *et al.*, "The probabilistic relevance framework: Bm25 and beyond," *Foundations and Trends® in Information Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.

[35] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, "Dense passage retrieval for open-domain question answering," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 6769–6781.

[36] A. Abdallah, J. Mozafari, B. Piryani, M. Ali, and A. Jatowt, "From retrieval to generation: Comparing different approaches," *arXiv preprint arXiv:2502.20245*, 2025.

[37] W. Chen, H. Zha, Z. Chen, W. Xiong, H. Wang, and W. Y. Wang, "Hybridqa: A dataset of multi-hop question answering over tabular and textual data," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 1026–1036.

[38] R. Aly, Z. Guo, M. Schlichtkrull, J. Thorne, A. Vlachos, C. Christodoulopoulos, O. Cocarascu, and A. Mittal, "Feverous: Fact extraction and verification over unstructured and structured information," in *35th Conference on Neural Information Processing Systems, NeurIPS 2021*. Neural Information Processing Systems foundation, 2021.

[39] W. Chen, H. Wang, J. Chen, Y. Zhang, H. Wang, S. Li, X. Zhou, and W. Y. Wang, "Tabfact: A large-scale dataset for table-based fact verification," in *International Conference on Learning Representations*.

[40] W. Chen, M.-W. Chang, E. Schlinger, W. Y. Wang, and W. W. Cohen, "Open question answering over tables and text," in *International Conference on Learning Representations*.

[41] C. Li, W. Ye, and Y. Zhao, "Finmath: Injecting a tree-structured solver for question answering over financial reports," in *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, 2022, pp. 6147–6152.

[42] X. Lu, L. Pan, Y. Ma, P. Nakov, and M.-Y. Kan, "Tart: An open-source tool-augmented framework for explainable table-based reasoning," in *Findings of the Association for Computational Linguistics: NAACL 2025*, 2025, pp. 4323–4339.

[43] M. Zheng, X. Feng, Q. Si, Q. She, Z. Lin, W. Jiang, and W. Wang, "Multimodal table understanding," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024, pp. 9102–9124.

[44] Z. Wu and Y. Feng, "Protrix: Building models for planning and reasoning over tables with sentence context," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024, pp. 4378–4406.

[45] W. Zhou, M. Mesgar, A. Friedrich, and H. Adel, "Efficient multi-agent collaboration with tool use for online planning in complex table question answering," in *Findings of the Association for Computational Linguistics: NAACL 2025*, 2025, pp. 945–968.

[46] B. Zhou, Z. Gao, Z. Wang, B. Zhang, Y. Wang, Z. Chen, and H. Xie, "Syntab-llava: Enhancing multimodal table understanding with decoupled synthesis," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 24 796–24 806.

[47] J. Baek, A. F. Aji, and A. 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)*, 2023, pp. 78–106.

[48] X. He, Y. Tian, Y. Sun, N. Chawla, T. Laurent, Y. LeCun, X. Bresson, and B. Hooi, "G-retriever: Retrieval-augmented generation for textual graph understanding and question answering," *Advances in Neural Information Processing Systems*, vol. 37, pp. 132 876–132 907, 2024.

[49] L. Luo, Y.-F. Li, G. Haffari, and S. Pan, "Reasoning on graphs: Faithful and interpretable large language model reasoning," *arXiv preprint arXiv:2310.01061*, 2023.

[50] J. Sun, C. Xu, L. Tang, S. Wang, C. Lin, Y. Gong, L. Ni, H.-Y. Shum, and J. Guo, "Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph," in *The Twelfth International Conference on Learning Representations*.

[51] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, and A. Miller, "Language models as knowledge bases?" 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)*, 2019, pp. 2463–2473.

[52] M. Yang, K. Chen, S. Sun, Z. Han, L. Kong, and Q. Meng, "A pattern driven graph ranking approach to attribute extraction for knowledge graph," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1250–1259, 2021.

[53] H. Yu, H. Li, D. Mao, and Q. Cai, "A relationship extraction method for domain knowledge graph construction," *World Wide Web*, vol. 23, pp. 735–753, 2020.

[54] W. Deng, P. Guo, and J. Yang, "Medical entity extraction and knowledge graph construction," in *2019 16th International Computer Conference on Wavelet Active Media Technology and Information Processing*. IEEE, 2019, pp. 41–44.

[55] K. Bollacker, R. Cook, and P. Tufts, "Freebase: A shared database of structured general human knowledge," in *AAAI*, vol. 7, 2007, pp. 1962–1963.

[56] C. Fellbaum, "Wordnet: An electronic lexical database," *MIT Press google schola*, vol. 2, pp. 678–686, 1998.

[57] D. Vrandečić, "Wikidata: A new platform for collaborative data collection," in *Proceedings of the 21st international conference on world wide web*, 2012, pp. 1063–1064.

[58] A. Yates, M. Banko, M. Broadhead, M. J. Cafarella, O. Etzioni, and S. Soderland, "Textrunner: open information extraction on the web," in *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, 2007, pp. 25–26.

[59] A. Fader, S. Soderland, and O. Etzioni, "Identifying relations for open information extraction," in *Proceedings of the 2011 conference on empirical methods in natural language processing*, 2011, pp. 1535–1545.

[60] M. Schmitz, S. Soderland, R. Bart, O. Etzioni *et al.*, "Open language learning for information extraction," in *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, 2012, pp. 523–534.

[61] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: a core of semantic knowledge," in *Proceedings of the 16th international conference on World Wide Web*, 2007, pp. 697–706.

[62] M. Ghazvininejad, O. Levy, Y. Liu, and L. Zettlemoyer, "Mask-predict: Parallel decoding of conditional masked language models," 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)*, K. Inui, J. Jiang,

V. Ng, and X. Wan, Eds. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 6112–6121. [Online]. Available: https://aclanthology.org/D19-1633

[63] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with gpus," *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535–547, 2019.

[64] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "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*, 2020, pp. 7871–7880.

[65] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee *et al.*, "Natural questions: a benchmark for question answering research," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 453–466, 2019.

[66] K. Shuster, S. Poff, M. Chen, D. Kiela, and J. Weston, "Retrieval augmentation reduces hallucination in conversation," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021, pp. 3784–3803.

[67] B. Saha, U. Saha, and M. Z. Malik, "Advancing retrieval-augmented generation with inverted question matching for enhanced qa performance," *IEEE Access*, 2024.

[68] K. Juvekar and A. Purwar, "Cos-mix: cosine similarity and distance fusion for improved information retrieval," *arXiv preprint arXiv:2406.00638*, 2024.

[69] W. Shi, S. Min, M. Yasunaga, M. Seo, R. James, M. Lewis, L. Zettlemoyer, and W.-t. Yih, "Replug: Retrieval-augmented black-box language models," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2024, pp. 8364–8377.

[70] J. A. Li, Y. Li, G. Li, X. Hu, X. Xia, and Z. Jin, "Editsum: A retrieve-and-edit framework for source code summarization," in *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2021, pp. 155–166.

[71] M. Kulkarni, P. Tangarajan, K. Kim, and A. Trivedi, "Reinforcement learning for optimizing rag for domain chatbots," *arXiv preprint arXiv:2401.06800*, 2024.

[72] J. Chen, Y. Pan, Y. Li, T. Yao, H. Chao, and T. Mei, "Retrieval augmented convolutional encoder-decoder networks for video captioning," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 19, no. 1s, pp. 1–24, 2023.

[73] C. Villani *et al.*, *Optimal transport: old and new*. Springer, 2008, vol. 338.

[74] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," *Advances in neural information processing systems*, vol. 26, 2013.

[75] A. Miller, A. Fisch, J. Dodge, A.-H. Karimi, A. Bordes, and J. Weston, "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.

[76] G. He, Y. Lan, J. Jiang, W. X. Zhao, and J.-R. Wen, "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*, 2021, pp. 553–561.

[77] Y. Sun, L. Zhang, G. Cheng, and Y. Qu, "Sparqa: skeleton-based semantic parsing for complex questions over knowledge bases," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 05, 2020, pp. 8952–8959.

[78] Y. Lan and J. 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*, 2020, pp. 969–974.

[79] Y. Gu and Y. Su, "Arcaneqa: Dynamic program induction and contextualized encoding for knowledge base question answering," in *Proceedings of the 29th International Conference on Computational Linguistics*, 2022, pp. 1718–1731.

[80] X. Ye, S. Yavuz, K. Hashimoto, Y. Zhou, and C. Xiong, "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)*, 2022, pp. 6032–6043.

[81] H. Sun, B. Dhingra, M. Zaheer, K. Mazaitis, R. Salakhutdinov, and W. Cohen, "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*, 2018, pp. 4231–4242.

[82] H. Sun, T. Bedrax-Weiss, and W. 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)*, 2019, pp. 2380–2390.

[83] J. Zhang, X. Zhang, J. Yu, J. Tang, J. Tang, C. Li, and H. Chen, "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)*, 2022, pp. 5773–5784.

[84] J. Baek, A. F. Aji, J. Lehmann, and S. J. Hwang, "Direct fact retrieval from knowledge graphs without entity linking," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023, pp. 10 038–10 055.

[85] J. Jiang, K. Zhou, X. Zhao, and J.-R. Wen, "Unikgqa: Unified retrieval and reasoning for solving multi-hop question answering over knowledge graph," in *The Eleventh International Conference on Learning Representations*.

[86] J. Kim, Y. Kwon, Y. Jo, and E. Choi, "Kg-gpt: A general framework for reasoning on knowledge graphs using large language models," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023, pp. 9410–9421.

[87] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, Q. Guo, M. Wang *et al.*, "Retrieval-augmented generation for large language models: A survey," *CoRR*, 2023.

[88] T. Li, X. Ma, A. Zhuang, Y. Gu, Y. Su, and W. Chen, "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), 2023, pp. 6966–6980.

[89] K. Wang, F. Duan, S. Wang, P. Li, Y. Xian, C. Yin, W. Rong, and Z. Xiong, "Knowledge-driven cot: Exploring faithful reasoning in llms for knowledge-intensive question answering," *CoRR*, 2023.

[90] D. Yu, S. Zhang, P. Ng, H. Zhu, A. H. Li, J. Wang, Y. Hu, W. Y. Wang, Z. Wang, and B. Xiang, "Decaf: Joint decoding of answers and logical forms for question answering over knowledge bases," in *The Eleventh International Conference on Learning Representations*.

[91] H. Luo, E. Haihong, Z. Tang, S. Peng, Y. Guo, W. Zhang, C. Ma, G. Dong, M. Song, W. Lin *et al.*, "Chatkbqa: A generate-then-retrieve framework for knowledge base question answering with fine-tuned large language models," in *Findings of the Association for Computational Linguistics ACL 2024*, 2024, pp. 2039–2056.

[92] Y. Gu, S. Kase, M. Vanni, B. Sadler, P. Liang, X. Yan, and Y. Su, "Beyond iid: three levels of generalization for question answering on knowledge bases," in *Proceedings of the Web Conference 2021*, 2021, pp. 3477–3488.

[93] J. Pérez, M. Arenas, and C. Gutierrez, "Semantics and complexity of sparql," *ACM Transactions on Database Systems (TODS)*, vol. 34, no. 3, pp. 1–45, 2009.

[94] L. LUO, Y.-F. Li, R. Haf, and S. Pan, "Reasoning on graphs: Faithful and interpretable large language model reasoning," in *The Twelfth International Conference on Learning Representations*.

[95] S. Yerragunta, R. Prasath, and G. Girish, "Bayesian-error-informed contrastive learning for knowledge-based question answering systems," *Computers and Electrical Engineering*, vol. 123, p. 110142, 2025.

[96] J. Jiang, K. Zhou, K. Ye, X. Zhao, J.-R. Wen *et al.*, "Structgpt: A general framework for large language model to reason over structured data," in *The 2023 Conference on Empirical Methods in Natural Language Processing*.

[97] Y. Gu, X. Deng, and Y. Su, "Don't generate, discriminate: A proposal for grounding language models to real-world environments," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023, pp. 4928–4949.

[98] B. Jin, G. Liu, C. Han, M. Jiang, H. Ji, and J. Han, "Large language models on graphs: A comprehensive survey," *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[99] Y. Tian, H. Song, Z. Wang, H. Wang, Z. Hu, F. Wang, N. V. Chawla, and P. Xu, "Graph neural prompting with large language models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 17, 2024, pp. 19 080–19 088.

[100] X. Huang, K. Han, Y. Yang, D. Bao, Q. Tao, Z. Chai, and Q. Zhu, "Can gnn be good adapter for llms?" in *Proceedings of the ACM Web Conference 2024*, 2024, pp. 893–904.

[101] B. Jin, C. Xie, J. Zhang, K. K. Roy, Y. Zhang, Z. Li, R. Li, X. Tang, S. Wang, Y. Meng *et al.*, "Graph chain-of-thought: Augmenting large language models by reasoning on graphs," in *Findings of the Association for Computational Linguistics ACL 2024*, 2024, pp. 163–184.

[102] C. Mavromatis, P. Karypis, and G. Karypis, "Sempool: Simple, robust, and interpretable kg pooling for enhancing language models," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2024, pp. 154–166.

[103] Y. Wu, N. Hu, S. Bi, G. Qi, J. Ren, A. Xie, and W. Song, "Retrieve-rewrite-answer: A kg-to-text enhanced llms framework for knowledge graph question answering," 2023.

[104] T. Yu, R. Zhang, K. Yang, M. Yasunaga, D. Wang, Z. Li, J. Ma, I. Li, Q. Yao, S. Roman *et al.*, "Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

[105] P. Yin, Z. Lu, H. Li, and B. Kao, "Neural enquirer: learning to query tables in natural language," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 2016, pp. 2308–2314.

[106] X. Yang, F. Nie, Y. Feng, Q. Liu, Z. Chen, and X. Zhu, "Program enhanced fact verification with verbalization and graph attention network," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 7810–7825.

[107] W. Zhong, D. Tang, Z. Feng, N. Duan, M. Zhou, M. Gong, L. Shou, D. Jiang, J. Wang, and J. Yin, "Logicalfactchecker: Leveraging logical operations for fact checking with graph module network," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 6053–6065.

[108] J. Herzig, P. K. Nowak, T. Mueller, F. Piccinno, and J. Eisenschlos, "Tapas: Weakly supervised table parsing via pre-training," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 4320–4333.

[109] P. Yin, G. Neubig, W.-t. Yih, and S. Riedel, "Tabert: Pretraining for joint understanding of textual and tabular data," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 8413–8426.

[110] Y. Zhou, X. Liu, K. Zhou, and J. Wu, "Table-based fact verification with self-adaptive mixture of experts," *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 139–149, 2022.

[111] Y. Ye, B. Hui, M. Yang, B. Li, F. Huang, and Y. Li, "Large language models are versatile decomposers: Decomposing evidence and questions for table-based reasoning," in *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, 2023, pp. 174–184.

[112] Z. Wang, H. Zhang, C.-L. Li, J. M. Eisenschlos, V. Perot, Z. Wang, L. Miculicich, Y. Fujii, J. Shang, C.-Y. Lee *et al.*, "Chain-of-table: Evolving tables in the reasoning chain for table understanding," in *The Twelfth International Conference on Learning Representations*, 2024.

[113] Y. Zhang, J. Henkel, A. Floratou, J. Cahoon, S. Deep, and J. M. Patel, "Reactable: enhancing react for table question answering," *Proceedings of the VLDB Endowment*, vol. 17, no. 8, pp. 1981–1994, 2024.

[114] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "React: Synergizing reasoning and acting in language models," in *International Conference on Learning Representations (ICLR)*, 2023.

[115] R. Liu, Y. Zhang, B. Yang, Q. Shi, and L. Tian, "Robust and resource-efficient table-based fact verification through multi-aspect adversarial contrastive learning," *Information Processing & Management*, vol. 61, no. 6, p. 103853, 2024.

[116] M. J. Cafarella, A. Halevy, D. Z. Wang, E. Wu, and Y. Zhang, "Webtables: exploring the power of tables on the web," *Proceedings of the VLDB Endowment*, vol. 1, no. 1, pp. 538–549, 2008.

[117] P. Pasupat and P. Liang, "Compositional semantic parsing on semi-structured tables," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, 2015.

[118] W. Chen, "Large language models are few (1)-shot table reasoners," in *Findings of the Association for Computational Linguistics: EACL 2023*, 2023, pp. 1120–1130.

[119] Y. Zhao, L. Chen, A. Cohan, and C. Zhao, "Tapera: enhancing faithfulness and interpretability in long-form table qa by content planning and execution-based reasoning," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024, pp. 12 824–12 840.

[120] F. Zhu, Z. Liu, F. Feng, C. Wang, M. Li, and T. S. Chua, "Tat-llm: A specialized language model for discrete reasoning over financial tabular and textual data," in *Proceedings of the 5th ACM International Conference on AI in Finance*, 2024, pp. 310–318.

[121] F. Zhu, W. Lei, Y. Huang, C. Wang, S. Zhang, J. Lv, F. Feng, and T.-S. Chua, "Tat-qa: A question answering benchmark on a hybrid of tabular and textual content in finance," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 3277–3287.

[122] R. M. Byrne, "Precis of the rational imagination: How people create alternatives to reality," *Behavioral and Brain Sciences*, vol. 30, no. 5-6, pp. 439–453, 2007.

[123] ——, "Counterfactuals in explainable artificial intelligence (xai): Evidence from human reasoning." in *IJCAI*. California, CA, 2019, pp. 6276–6282.

[124] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *Artificial intelligence*, vol. 267, pp. 1–38, 2019.

[125] S. Dhuliawala, M. Komeili, J. Xu, R. Raileanu, X. Li, A. Celikyilmaz, and J. Weston, "Chain-of-verification reduces hallucination in large language models," in *Findings of the Association for Computational Linguistics ACL 2024*, 2024, pp. 3563–3578.

[126] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, and Z. Liu, "Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation," *CoRR*, 2024.

[127] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan *et al.*, "The llama 3 herd of models," *CoRR*, 2024.

[128] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei *et al.*, "Qwen2. 5 technical report," *arXiv preprint arXiv:2412.15115*, 2024.

[129] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: Efficient finetuning of quantized llms," *Advances in neural information processing systems*, vol. 36, pp. 10 088–10 115, 2023.

[130] M. Akhtar, O. Cocarascu, and E. Simperl, "Pubhealthtab: A public health table-based dataset for evidence-based fact checking," in *Findings of the Association for Computational Linguistics: NAACL 2022*, 2022, pp. 1–16.

[131] X. Lu, L. Pan, Q. Liu, P. Nakov, and M.-Y. Kan, "Scitab: A challenging benchmark for compositional reasoning and claim verification on scientific tables," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 7787–7813.

[132] J. A. Baktash and M. Dawodi, "Gpt-4: A review on advancements and opportunities in natural language processing," *arXiv preprint arXiv:2305.03195*, 2023.

[133] M. Zheng, Z. Feng, J. Wang, L. Wang, Z. Lin, Y. Hao, and W. Wang, "Tabledreamer: Progressive and weakness-guided data synthesis from scratch for table instruction tuning," *arXiv preprint arXiv:2506.08646*, 2025.

[134] S. K. Jauhar, P. Turney, and E. Hovy, "Tabmcq: A dataset of general knowledge tables and multiple-choice questions," *arXiv preprint arXiv:1602.03960*, 2016.

[135] J. Kim, M. Bae, S. Lee, J. Yoon, and H. J. Kim, "Tabflash: Efficient table understanding with progressive question conditioning and token focusing," *arXiv preprint arXiv:2511.13283*, 2025.

[136] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 2008, pp. 1247–1250.

[137] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[138] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: A large ontology from wikipedia and wordnet," *Journal of Web Semantics*, vol. 6, no. 3, pp. 203–217, 2008.

[139] J. Deng, C. Chen, X. Huang, W. Chen, and L. Cheng, "Research on the construction of event logic knowledge graph of supply chain management," *Advanced Engineering Informatics*, vol. 56, p. 101921, 2023.

[140] Q. Guo, F. Zhuang, C. Qin, H. Zhu, X. Xie, H. Xiong, and Q. He, "A survey on knowledge graph-based recommender systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 8, pp. 3549–3568, 2020.

[141] J. Liu, A. Liu, X. Lu, S. Welleck, P. West, R. L. Bras, Y. Choi, and H. Hajishirzi, "Generated knowledge prompting for commonsense reasoning," *arXiv preprint arXiv:2110.08387*, 2021.

[142] S. Deng, C. Wang, Z. Li, N. Zhang, Z. Dai, H. Chen, F. Xiong, M. Yan, Q. Chen, M. Chen *et al.*, "Construction and applications of billion-scale pre-trained multimodal business knowledge graph," in *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, 2023, pp. 2988–3002.

[143] X. Mao, H. Sun, X. Zhu, and J. Li, "Financial fraud detection using the related-party transaction knowledge graph," *Procedia Computer Science*, vol. 199, pp. 733–740, 2022.

[144] J. P. Chiu and E. Nichols, "Named entity recognition with bidirectional lstm-cnns," *Transactions of the association for computational linguistics*, vol. 4, pp. 357–370, 2016.

[145] G. Zhou, J. Su, J. Zhang, and M. Zhang, "Exploring various knowledge in relation extraction," in *Proceedings of the 43rd annual meeting of the association for computational linguistics (acl'05)*, 2005, pp. 427–434.

[146] L. Yu, F. Tian, P. Kuang, and F. Zhou, "Amplifying diversity and quality in commonsense knowledge graph completion (student abstract)," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 21, 2024, pp. 23 699–23 700.

[147] D. Alivanistos, S. B. Santamaría, M. Cochez, J.-C. Kalo, E. van Krieken, and T. Thanapalasingam, "Prompting as probing: Using language models for knowledge base construction," *arXiv preprint arXiv:2208.11057*, 2022.

[148] Z. Jiang, F. F. Xu, J. Araki, and G. Neubig, "How can we know what language models know?" *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 423–438, 2020.

[149] N. Poerner, U. Waltinger, and H. Schütze, "E-BERT: Efficient-yet-effective entity embeddings for BERT," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, T. Cohn, Y. He, and Y. Liu, Eds. Online: Association for Computational Linguistics, Nov. 2020, pp. 803–818. [Online]. Available: https://aclanthology.org/2020.findings-emnlp.71

[150] T. Shin, Y. Razeghi, R. L. Logan IV, E. Wallace, and S. Singh, "AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds. Online: Association for Computational Linguistics, Nov. 2020, pp. 4222–4235. [Online]. Available: https://aclanthology.org/2020.emnlp-main.346

[151] Y. Elazar, N. Kassner, S. Ravfogel, A. Feder, A. Ravichander, M. Mosbach, Y. Belinkov, H. Schütze, and Y. Goldberg, "Measuring causal effects of data statistics on language model'sfactual'predictions," *arXiv preprint arXiv:2207.14251*, 2022.

[152] Y. Elazar, N. Kassner, S. Ravfogel, A. Ravichander, E. Hovy, H. Schütze, and Y. Goldberg, "Measuring and improving consistency in pretrained language models," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 1012–1031, 2021.

[153] B. Cao, H. Lin, X. Han, L. Sun, L. Yan, M. Liao, T. Xue, and J. Xu, "Knowledgeable or educated guess? revisiting language models as knowledge bases," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds. Online: Association for Computational Linguistics, Aug. 2021, pp. 1860–1874. [Online]. Available: https://aclanthology.org/2021.acl-long.146

[154] N. Kandpal, H. Deng, A. Roberts, E. Wallace, and C. Raffel, "Large language models struggle to learn long-tail knowledge," in *International Conference on Machine Learning*. PMLR, 2023, pp. 15 696–15 707.

[155] M. Grootendorst. (2020) Keybert: Minimal keyword extraction with bert. [Online]. Available: https://zenodo.org/records/4461265

[156] N. Arefyev, B. Sheludko, A. Podolskiy, and A. Panchenko, "Always keep your target in mind: Studying semantics and improving performance of neural lexical substitution," in *Proceedings of the 28th International Conference on Computational Linguistics*, D. Scott, N. Bel, and C. Zong, Eds. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 1242–1255. [Online]. Available: https://aclanthology.org/2020.coling-main.107

[157] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," *arXiv preprint arXiv:1904.09675*, 2019.

[158] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[159] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger, "From word embeddings to document distances," in *International conference on machine learning*. PMLR, 2015, pp. 957–966.

[160] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *International journal of computer vision*, vol. 40, pp. 99–121, 2000.

[161] C. Wei, B. Wang, and C.-C. J. Kuo, "Synwmd: Syntax-aware word mover's distance for sentence similarity evaluation," *Pattern Recognition Letters*, vol. 170, pp. 48–55, 2023.

[162] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving language understanding by generative pre-training," 2018.

[163] R. Speer, J. Chin, and C. Havasi, "Conceptnet 5.5: An open multilingual graph of general knowledge," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, no. 1, 2017.

[164] D. Vrandečić and M. Krötzsch, "Wikidata: a free collaborative knowledgebase," *Communications of the ACM*, vol. 57, no. 10, pp. 78–85, 2014.

[165] P. Mendes, M. Jakob, and C. Bizer, "DBpedia: A multilingual cross-domain knowledge base," in *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, N. Calzolari, K. Choukri, T. Declerck, M. U. Doğan, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, and S. Piperidis, Eds. Istanbul, Turkey: European Language Resources Association (ELRA), May 2012, pp. 1813–1817.

[166] D.-T. Vo and E. Bagheri, "Open information extraction," *Encyclopedia with semantic computing and Robotic intelligence*, vol. 1, no. 01, p. 1630003, 2017.

[167] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *biometrics*, pp. 159–174, 1977.

[168] W. Yuan, G. Neubig, and P. Liu, "Bartscore: Evaluating generated text as text generation," *Advances in Neural Information Processing Systems*, vol. 34, pp. 27 263–27 277, 2021.

[169] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[170] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier *et al.*, "Mistral 7b," *arXiv preprint arXiv:2310.06825*, 2023.

[171] A. Bosselut, H. Rashkin, M. Sap, C. Malaviya, A. Celikyilmaz, and Y. Choi, "Comet: Commonsense transformers for automatic knowledge graph construction," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 4762–4779.

[172] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," *Advances in neural information processing systems*, vol. 35, pp. 27 730–27 744, 2022.

[173] R. Thoppilan, D. De Freitas, J. Hall, N. Shazeer, A. Kulshreshtha, H.-T. Cheng, A. Jin, T. Bos, L. Baker, Y. Du *et al.*, "Lamda: Language models for dialog applications," *arXiv preprint arXiv:2201.08239*, 2022.

[174] F. Petroni, A. Piktus, A. Fan, P. Lewis, M. Yazdani, N. De Cao, J. Thorne, Y. Jernite, V. Karpukhin, J. Maillard *et al.*, "Kilt: a benchmark for knowledge intensive language tasks," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021, pp. 2523–2544.

[175] A. Talmor and J. Berant, "The web as a knowledge-base for answering complex questions," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 641–651.

[176] H. Zhang, Y. Gong, X. He, D. Liu, D. Guo, J. Lv, and J. Guo, "Noisy pair corrector for dense retrieval," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023, pp. 11 439–11 451.

[177] H. Ren, W. Hu, and J. Leskovec, "Query2box: Reasoning over knowledge graphs in vector space using box embeddings," in *International Conference on Learning Representations (ICLR)*, 2020.

[178] H. Ren and J. Leskovec, "Beta embeddings for multi-hop logical reasoning in knowledge graphs," *Advances in Neural Information Processing Systems*, vol. 33, pp. 19 716–19 726, 2020.

[179] B. Y. Lin, X. Chen, J. Chen, and X. Ren, "Kagnet: Knowledge-aware graph networks for commonsense reasoning," 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)*, 2019, pp. 2829–2839.

[180] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," *Advances in neural information processing systems*, vol. 26, 2013.

[181] K. Guu, J. Miller, and P. Liang, "Traversing knowledge graphs in vector space," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 318–327.

[182] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, and X. Wu, "Unifying large language models and knowledge graphs: A roadmap," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 7, pp. 3580–3599, 2024.

[183] S. Min, D. Chen, L. Zettlemoyer, and H. Hajishirzi, "Knowledge guided text retrieval and reading for open domain question answering," *arXiv preprint arXiv:1911.03868*, 2019.

[184] M. R. A. H. Rony, R. Usbeck, and J. Lehmann, "Dialokg: Knowledge-structure aware task-oriented dialogue generation," in *Findings of the Association for Computational Linguistics: NAACL 2022*, 2022, pp. 2557–2571.

[185] Y. Feng, X. Chen, B. Y. Lin, P. Wang, J. Yan, and X. Ren, "Scalable multi-hop relational reasoning for knowledge-aware question answering," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 1295–1309.

[186] S. Lv, D. Guo, J. Xu, D. Tang, N. Duan, M. Gong, L. Shou, D. Jiang, G. Cao, and S. Hu, "Graph-based reasoning over heterogeneous external knowledge for commonsense question answering," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 05, 2020, pp. 8449–8456.

[187] T. Vo, S. T. Luu, and L.-M. Nguyen, "K-bloom: unleashing the power of pre-trained language models in extracting knowledge graph with predefined relations," *Knowledge and Information Systems*, pp. 1–35, 2025.

[188] O. Bodenreider, "The unified medical language system (umls): integrating biomedical terminology," *Nucleic acids research*, vol. 32, no. suppl_1, pp. D267–D270, 2004.

[189] D. S. Wishart, Y. D. Feunang, A. C. Guo, E. J. Lo, A. Marcu, J. R. Grant, T. Sajed, D. Johnson, C. Li, Z. Sayeeda *et al.*, "Drugbank 5.0: a major update to the drugbank database for 2018," *Nucleic acids research*, vol. 46, no. D1, pp. D1074–D1082, 2018.

[190] D. Khashabi, T. Khot, A. Sabharwal, and D. Roth, "Learning what is essential in questions," in *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, 2017, pp. 80–89.

[191] W. Zhong, D. Tang, N. Duan, M. Zhou, J. Wang, and J. Yin, "Improving question answering by commonsense-based pre-training," in *Natural Language Processing and Chinese Computing: 8th CCF International Conference, NLPCC 2019, Dunhuang, China, October 9–14, 2019, Proceedings, Part I 8*. Springer, 2019, pp. 16–28.

[192] X. Wang, P. Kapanipathi, R. Musa, M. Yu, K. Talamadupula, I. Abdelaziz, M. Chang, A. Fokoue, B. Makni, N. Mattei *et al.*, "Improving natural language inference using external knowledge in the science questions domain," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 7208–7215.

[193] M. Grootendorst, "Keybert: Minimal keyword extraction with bert." 2020. [Online]. Available: https://doi.org/10.5281/zenodo.4461265

[194] D. Jin, E. Pan, N. Oufattole, W.-H. Weng, H. Fang, and P. Szolovits, "What disease does this patient have? a large-scale open domain question answering dataset from medical exams," *Applied Sciences*, vol. 11, no. 14, p. 6421, 2021.

[195] A. Pal, L. K. Umapathi, and M. Sankarasubbu, "Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering," in *Conference on health, inference, and learning*. PMLR, 2022, pp. 248–260.

[196] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt, "Measuring massive multitask language understanding," *arXiv preprint arXiv:2009.03300*, 2020.

[197] Z. Chen, A. H. Cano, A. Romanou, A. Bonnet, K. Matoba, F. Salvi, M. Pagliardini, S. Fan, A. Köpf, A. Mohtashami *et al.*, "Meditron-70b: Scaling medical pretraining for large language models," *arXiv preprint arXiv:2311.16079*, 2023.

[198] K. Singhal, T. Tu, J. Gottweis, R. Sayres, E. Wulczyn, M. Amin, L. Hou, K. Clark, S. R. Pfohl, H. Cole-Lewis *et al.*, "Toward expert-level medical question answering with large language models," *Nature Medicine*, pp. 1–8, 2025.

[199] K. Zhang, S. Zeng, E. Hua, N. Ding, Z.-R. Chen, Z. Ma, H. Li, G. Cui, B. Qi, X. Zhu *et al.*, "Ultramedical: Building specialized generalists in biomedicine," *Advances in Neural Information Processing Systems*, vol. 37, pp. 26 045–26 081, 2024.

[200] Y. Wang, X. Ma, and W. Chen, "Augmenting black-box llms with medical textbooks for biomedical question answering," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024, pp. 1754–1770.

[201] S. S. Mullappilly, M. I. Kurpath, S. Pieri, S. Y. Alseiari, S. Cholakkal, K. Aldahmani, F. Khan, R. Anwer, S. Khan, T. Baldwin *et al.*, "Bimedix2: Bio-medical expert lmm for diverse medical modalities," *arXiv preprint arXiv:2412.07769*, 2024.

[202] J. Chen, Z. Cai, K. Ji, X. Wang, W. Liu, R. Wang, J. Hou, and B. Wang, "Huatuogpt-o1, towards medical complex reasoning with llms," *arXiv preprint arXiv:2412.18925*, 2024.

[203] M. S. Ankit Pal, "Openbiollms: Advancing open-source large language models for healthcare and life sciences," https://huggingface.co/aaditya/OpenBioLLM-Llama3-70B, 2024.

[204] V. Liévin, C. E. Hother, A. G. Motzfeldt, and O. Winther, "Can large language models reason about medical questions?" *Patterns*, vol. 5, no. 3, 2024.

[205] G. Xiong, Q. Jin, Z. Lu, and A. Zhang, "Benchmarking retrieval-augmented generation for medicine," in *Findings of the Association for Computational Linguistics ACL 2024*, 2024, pp. 6233–6251.

[206] K. Lu, Z. Liang, D. Pan, S. Zhang, X. Wu, W. Chen, Z. Zhou, G. Dong, B. Cui, and W. Zhang, "Med-r: Crafting trustworthy llm physicians through retrieval and reasoning of evidence-based medicine," *arXiv preprint arXiv:2501.11885*, 2025.

[207] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "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)*, 2019, pp. 4171–4186.

[208] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, 2019.

[209] B. Oguz, X. Chen, V. Karpukhin, S. Peshterliev, D. Okhonko, M. Schlichtkrull, S. Gupta, Y. Mehdad, and S. Yih, "Unik-qa: Unified representations of structured and unstructured knowledge for open-domain question answering," in *Findings of the Association for Computational Linguistics: NAACL 2022*, 2022, pp. 1535–1546.

[210] K. Ma, H. Cheng, X. Liu, E. Nyberg, and J. Gao, "Open domain question answering with a unified knowledge interface," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 1605–1620.

[211] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.

[212] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," *Advances in neural information processing systems*, vol. 35, pp. 22 199–22 213, 2022.

[213] X. Wang, J. Wei, D. Schuurmans, Q. V. Le, E. H. Chi, S. Narang, A. Chowdhery, and D. Zhou, "Self-consistency improves chain of thought reasoning in language models," in *The Eleventh International Conference on Learning Representations*.

[214] S. Kadavath, T. Conerly, A. Askell, T. Henighan, D. Drain, E. Perez, N. Schiefer, Z. Hatfield-Dodds, N. DasSarma, E. Tran-Johnson *et al.*, "Language models (mostly) know what they know," *CoRR*, 2022.

[215] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *International Conference on Learning Representations*, 2018.

[216] M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P.-E. Mazaré, M. Lomeli, L. Hosseini, and H. Jégou, "The faiss library," *CoRR*, 2024.

[217] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," in *International Conference on Learning Representations*.

[218] H. Yamagiwa, S. Yokoi, and H. Shimodaira, "Improving word mover's distance by leveraging self-attention matrix," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023, pp. 11 160–11 183.

[219] M. Yasunaga, A. Bosselut, H. Ren, X. Zhang, C. D. Manning, P. S. Liang, and J. Leskovec, "Deep bidirectional language-knowledge graph pretraining," *Advances in Neural Information Processing Systems*, vol. 35, pp. 37 309–37 323, 2022.

[220] X. Zhang, A. Bosselut, M. Yasunaga, H. Ren, P. Liang, C. Manning, and J. Leskovec, "Greaselm: Graph reasoning enhanced language models for question answering," in *International Conference on Representation Learning (ICLR)*, 2022.

[221] D. Li, Y. Ma, N. Wang, Z. Cheng, L. Duan, J. Zuo, C. Yang, and M. Tang, "Mixlora: Enhancing large language models fine-tuning with lora based mixture of experts," *CoRR*, 2024.

[222] A. Talmor, J. Herzig, N. Lourie, and J. Berant, "Commonsenseqa: A question answering challenge targeting commonsense knowledge," 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)*, 2019, pp. 4149–4158.

[223] T. Mihaylov, P. Clark, T. Khot, and A. Sabharwal, "Can a suit of armor conduct electricity? a new dataset for open book question answering," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 2381–2391.

[224] B. Y. Lin, Z. Wu, Y. Yang, D.-H. Lee, and X. Ren, "Riddlesense: Reasoning about riddle questions featuring linguistic creativity and commonsense knowledge," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2021, pp. 1504–1515.

[225] Y. Bisk, R. Zellers, J. Gao, Y. Choi *et al.*, "Piqa: Reasoning about physical commonsense in natural language," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 05, 2020, pp. 7432–7439.

[226] P. Clark, I. Cowhey, O. Etzioni, T. Khot, A. Sabharwal, C. Schoenick, and O. Tafjord, "Think you have solved question answering? try arc, the AI2 reasoning challenge," *CoRR*, 2018.

[227] D. Taunk, L. Khanna, S. V. P. K. Kandru, V. Varma, C. Sharma, and M. Tapaswi, "Grapeqa: Graph augmentation and pruning to enhance question-answering," in *Companion Proceedings of the ACM Web Conference 2023*, 2023, pp. 1138–1144.

[228] Y. Huang, Y. Li, Y. Xu, L. Zhang, R. Gan, J. Zhang, and L. Wang, "Mvp-tuning: Multi-view knowledge retrieval with prompt tuning for commonsense reasoning," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023, pp. 13 417–13 432.

[229] H. Wu, H. Zheng, Z. He, and B. Yu, "Parameter-efficient sparsity crafting from dense to mixture-of-experts for instruction tuning on general tasks," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024, pp. 737–749.

[230] R. Dai, Y. Tan, L. Mo, S. Liang, G. Huo, J. Luo, and Y. Cheng, "G-sap: Graph-based structure-aware prompt learning over heterogeneous knowledge for commonsense reasoning," in *Proceedings of the 2024 International Conference on Multimedia Retrieval*, 2024, pp. 1051–1060.

[231] T. Wu, J. Wang, Z. Zhao, and N. Wong, "Mixture-of-subspaces in low-rank adaptation," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024, pp. 7880–7899.

[232] S. A. G. Shakhadri, D. K. KR, and R. Aralimatti, "Shakti: A 2.5 billion parameter small language model optimized for edge ai and low-resource environments," in *IFIP International Conference on Artificial Intelligence Applications and Innovations*. Springer, 2025, pp. 434–447.

[233] H. Yu, T. Cheng, Y. Cheng, and R. Feng, "Finemedlm-o1: Enhancing the medical reasoning ability of LLM from supervised fine-tuning to test-time training," *CoRR*, 2025.

[234] S. Jiang, Y. Liao, Z. Chen, Y. Zhang, Y. Wang, and Y. Wang, "Meds: Towards medical small language models with self-evolved slow thinking," *CoRR*, 2025.

[235] E. C. Acikgoz, O. B. Ince, R. Bench, A. A. Boz, I. Kesen, A. Erdem, and E. Erdem, "Hippocrates: An open-source framework for advancing large language models in healthcare," *CoRR*, 2024.

[236] D.-H. Zhu, Y.-J. Xiong, J.-C. Zhang, X.-J. Xie, and C.-M. Xia, "Understanding before reasoning: Enhancing chain-of-thought with iterative summarization pre-prompting," *arXiv preprint arXiv:2501.04341*, 2025.

[237] Y. Liu, X. Peng, T. Du, J. Yin, W. Liu, and X. Zhang, "Era-cot: Improving chain-of-thought through entity relationship analysis," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024, pp. 8780–8794.

[238] X. Xu, C. Tao, T. Shen, C. Xu, H. Xu, G. Long, J.-G. Lou, and S. Ma, "Re-reading improves reasoning in large language models," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024, pp. 15 549–15 575.

[239] W. Zhao, M. Peyrard, F. Liu, Y. Gao, C. M. Meyer, and S. Eger, "Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance," 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)*, 2019, pp. 563–578.

Publications

[1] Vo, T., Luu, S.T. & Nguyen, LM. **K-Bloom: unleashing the power of pre-trained language models in extracting knowledge graph with predefined relations**. *Knowl Inf Syst 67, 4487–4521 (2025).* https://doi.org/10.1007/s10115-025-02345-1

[2] Vo, T., Luu, S.T. & Nguyen, L.M., 2025. **USCRaKE: Unsupervised semantic chunk retrieval and knowledge-enhanced reasoning for multiple-choice question answering**. *Neurocomputing*, p.131932. https://doi.org/10.1016/j.neucom.2025.131932

[3] Trung Vo, An Trieu, Vu Tran, Yuji Matsumoto & Le-Minh Nguyen. **CancerRAGent: Evidence-Linked and Safety-Guided Oncology Question Answering. ECIR** 2026

[4] Vo, T., Luu, S.T. & Nguyen, LM. **UGAT-MedQA: Unsupervised Graph Attention Network Empowered by LLMs for Medical Question Answering**. *Computers and Electrical Engineering*, (**1st Revision**). In: 2025

[5] Vo, T., Luu, S.T. & Nguyen, LM. **UCRET: Unsupervised Column Relevance Extraction for Table-Based Fact Verification via Structured and Explainable Large Language Model Reasoning**. *Computers and Electrical Engineering*, (**Under Review**). In: 2025

[6] Vo, T., Luu, S.T. & Nguyen, LM. **Chain-of-Hypothesis: Hypothesis-First Prompting with Re-Reading for Compact and Auditable Vietnamese Math Reasoning.**, *Information Processing and Management*, (**Under Review**). In: 2026

[7] Luu, S.T., Vo, T. & Nguyen, LM. **MCVE: multimodal claim verification and explanation framework for fact-checking system.** *Multimedia Systems 31, 242 (2025).* https://doi.org/10.1007/s00530-025-01804-7

[8] Luu, S.T., Vo, T. & Nguyen, LM. **M-RAV: Multimodal Retrieve-Augment-Verify Framework for Zero-shot Fact Verification**

System. *Information Processing and Management*, (**Under Review**). In: (2025)

[9] Luu, S.T., Trung Vo, Vu Tran, Tomoko Matsui, & Nguyen, LM. **Boosting large-language models for fact-checking: leveraging verbalized tabular data as evidence.** *International Journal of Data Science and Analytics*, (**Accepted**). In: (2026)

[10] Liyanaarachchi, S., Dinh, T., Yue, W., Vo, T., Nguyen, LM. (2026). **Sentiment Analysis of Travel Reviews in Kyoto Using LLMs**. In: Yoshikawa, M., Meng, X., Cao, Y., Xiao, C., Chen, W., Wang, Y. (eds) Advanced Data Mining and Applications. ADMA 2025. Lecture Notes in Computer Science(), vol 16199. Springer, Singapore. https://doi.org/10.1007/978-981-95-3459-3_20

[11] Luu, S.T., Vo, T., Nguyen, H., Tran, K.Q., Van Nguyen, K., Tran, V., Luu-Thuy Nguyen, N. and Nguyen, L.M., 2025. (2025, October). **VLSP 2025 MLQA-TSR Challenge: Vietnamese Multimodal Legal Question Answering on Traffic Sign Regulation**. In Proceedings of the 11th International Workshop on Vietnamese Language and Speech Processing (pp. 402-409)

[12] Nguyen, C., Tran, T., Le, K., Nguyen, H., Do, T., Pham, T., Luu, S.T., Vo, T. and Nguyen, L.M., 2024, May. **Pushing the boundaries of legal information processing with integration of large language models**. In JSAI International Symposium on Artificial Intelligence (pp. 167-182). Singapore: Springer Nature Singapore.

[13] LE, K.N., NGUYEN, M.N., NGUYEN, H., VO, T., LUU, S., BUI, Q.M., NOMURA, S. and Le NGUYEN, M., 2025. **Automated Web Application Testing: End-to-End Test Case Generation with Large Language Models and Screen Transition Graphs**. In 人工知能学会全国大会論文集 第 39 回 (2025) (pp. 3K4IS2a03-3K4IS2a03). 一般社団法人 人工知能学会.

[14] Do, D.T., Luu, S.T., Pham, T., Vo, T., Chu, N.H., Chu, Q.H., Nguyen, C., Nguyen, M., Trieu, A., Nguyen, D. and Tran, T., 2024, November. **A Summary of the ALQAC 2024 Competition**. In 2024 16th International Conference on Knowledge and System Engineering (KSE) (pp. 422-427). IEEE.

[15] Luu, S.T., Nguyen, H., Vo, T., Nguyen, LM. (2025). **ZeFaV: Boosting Large Language Models for Zero-Shot Fact Verification**.

In: *PRICAI 2024: Trends in Artificial Intelligence. PRICAI 2024. Lecture Notes in Computer Science, vol 15282. Springer, Singapore.* https://doi.org/10.1007/978-981-96-0119-6_28

[16] Nguyen, C., Luu, S.T., Tran, T., Trieu, A., Dang, A., Nguyen, D., Nguyen, H., Pham, T., Pham, T., Vo, T.T. and Dol, D.T., 2023, October. **A summary of the alqac 2023 competition**. In 2023 15th International Conference on Knowledge and Systems Engineering (KSE) (pp. 1-6). IEEE.

[17] Tran, V., Nguyen, H.T., Vo, T., Luu, S.T., Dang, H.A., Le, N.C., Le, T.T., Nguyen, M.T., Nguyen, T.S. and Nguyen, L.M., 2024. **VLSP 2023–LTER: A Summary of the Challenge on Legal Textual Entailment Recognition**. arXiv preprint arXiv:2403.03435.

Awards

[1] Achieved the highest performance on the Pilot Task (Legal Judgment Prediction for Japanese Tort Law) of the 12th Competition on Legal Information Extraction and Entailment (COLIEE 2025).

[2] The organizing committee for the Automated Legal Question Answering Competition (ALQAC), held as part of the Knowledge and Systems Engineering (KSE) conference in 2023, 2024, and 2025.

[3] Organizing committee of the challenge on Multimodal Legal QA on Traffic Sign Rules on VLSP 2025.

[4] Be selected as the recipient of the 2025 JSAI Annual Conference Award.