KG-Scout : A Policy Driven Knowledge-Graph Retrieval framework to Mitigate Factual Inaccuracies of Large Language Model

by

SOURAV CHAKRABORTY

Under the supervision of

Dr. Md. Shad Akhtar

Submitted

in partial fulfilment of the requirements for the degree of

MASTER OF TECHNOLOGY

to

COMPUTER SCIENCE AND ENGINEERING

INDRAPRASTHA INSTITUTE OF INFORMATION TECHNOLOGY DELHI

NEW DELHI- 110020

August, 2025

THESIS CERTIFICATE

This is to certify that the thesis titled "***KG-Scout* : A Policy Driven Knowledge-Graph Retrieval framework to Mitigate Factual Inaccuracies of Large Language Model**", submitted by **Sourav Chakraborty**, to the Indraprastha Institute of Information Technology Delhi, for the award of the degree of Master of Technology, is an original research work carried out by him under my supervision. In my opinion, the thesis has reached the standards fulfilling the requirements of the regulations relating to the degree.

The results contained in this thesis have not been submitted in part or full to any other university or institute for the award of any degree/diploma.

August, 2025

Dr. Md. Shad Akhtar
Assistant Professor
Department of Computer Science and Engineering
Indraprastha Institute of Information Technology Delhi
New Delhi, 110020

ACKNOWLEDGEMENTS

I would like to express my heartfelt gratitude to all those who have supported and inspired me throughout this research journey.

First and foremost, I am deeply thankful to my advisor, Dr. Md. Shad Akhtar, for his invaluable guidance, consistent support, and insightful feedback. His mentorship has not only shaped the direction of this thesis but has also significantly contributed to my academic and professional development.

I am also grateful to the members of the *FLaMe Research Lab* for fostering a collaborative and intellectually stimulating environment. The discussions, technical guidance, and encouragement from the group have played a big role in refining my ideas.

I am deeply grateful to my parents for their unconditional love, encouragement, and belief in me throughout this journey. Their unwavering support has been a source of strength and motivation at every stage.

Lastly, I would like to acknowledge myself—for the resilience, perseverance, and dedication it took to see this work through. This journey has been a meaningful blend of academic growth and personal transformation.

Sourav Chakraborty

MT23099

ABSTRACT

Large Language Models (LLMs) have rapidly advanced their ability to answer questions and perform complex reasoning tasks. However, they often generate factual inaccuracies and hallucinations because they lack access or have limited access to up-to-date factual knowledge. To mitigate this, researchers often augment LLMs with factual information from external sources, such as knowledge graphs (KGs). However, most existing KG-based RAG systems suffer from a key limitation: triplet retrieval from KGs is either based on simplistic distance metrics, heuristics, or tightly coupled with reasoning, making optimizing both retrieval and reasoning challenging. To mitigate these, we propose *KG-Scout* , a reinforcement learning (RL)-based policy network that decouples retrieval from reasoning, enabling the selection of triplets that are both semantically aligned with the query and structurally important in the KG. Our approach operates in two key stages: (1) extracting a subgraph using topic entities and computing Personalized PageRank (PPR) scores for nodes, and (2) employing our policy network to select the most valuable triplets from this set based on their learned relevance scoring. To enhance the efficiency of this process, we first perform an initial filtering of candidate triplets using cosine similarity with the query before the policy network considers them. Using the retrieved results, smaller pretrained LLMs such as LLAMA-3.1-8b outperform several complex LLM-based baselines on WebQSP and CWQ benchmarks.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

The impressive performance of large language models (LLMs) across various NLP tasks has established them as powerful tools for complex problem-solving(Brown *et al.*, 2020; Kojima *et al.*, 2022). However, their reliability is often compromised by a tendency to generate factually incorrect (Augenstein *et al.*, 2024) or outdated information(Dhingra *et al.*, 2022). Retrieval Augmented Generation (RAG) (Lewis *et al.*, 2020) has emerged as a leading strategy to mitigate these issues by grounding LLM outputs in external, up-to-date knowledge sources (Shuster *et al.*, 2021; Borgeaud *et al.*, 2022).

One highly structured way to store this external information is through knowledge graphs (KGs). While KGs offer a rich source of factual data for RAG systems, a significant challenge remains: efficiently identifying the most relevant facts or triplets required to answer a specific query. Retrieving too much information can be computationally expensive and introduce irrelevant noise, which can degrade the performance of the RAG system. Conversely, retrieving too little information can lead to incomplete or incorrect answers, especially for complex questions requiring multi-hop reasoning.

Current approaches to this problem are often limited. Some methods rely on simple heuristics (Jiang *et al.*, 2023; He *et al.*, 2024) or basic similarity measures (Yang *et al.*; Yu *et al.*, 2022), which struggle with questions which require reasoning over multiple triplets. Other techniques employ iterative LLM calls to navigate the KG, a process that is both computationally costly and slow (Sun *et al.*, 2023; Wang *et al.*, 2023).

To overcome these limitations, we introduce *KG-Scout* , a novel framework that reframes the problem of relevant triplet selection as a reinforcement learning task. Instead of using rigid heuristics or expensive LLM calls, *KG-Scout* employs a lightweight policy network to intelligently search for high-quality triplets from the KG. This policy network is trained to consider multiple facets of relevance simultaneously, including the semantic similarity between the query and the triplet, the structural connectivity within the KG, and the coherence of the reasoning path.

Our core insight is that an effective triplet selection strategy must be adaptive. Simple factual questions may primarily require a strong semantic match, whereas complex multi-hop queries demand a deep understanding of structural pathways. *KG-Scout* achieves this adaptability through a dual-tower architecture that processes these different signals—semantic content, relational structure, and structural signals like PPR scores—in parallel. These distinct streams are then fused using a learned gating mechanism that dynamically weighs the importance of each information source based on the nature of the input query. By doing so,*KG-Scout* offers a more nuanced and efficient approach to grounding LLM responses with evidence from knowledge graphs. We evaluate *KG-Scout* on two Knowledge graph question-answer benchmarks, WebQSP and CWQ .

Even without fine-tuning our retriever LLM which reasons over the triplets and finds answer, we observed that smaller LLMs such as Llama-3.1-8b can achieve competitive performances on the CWQ dataset and outperform several baselines on the WebQSP dataset. The light weight nature of the retriever makes it a promising candidate for use as a KG triplet retriever to mitigate factual inaccuracies in LLM.

This thesis makes the following contributions:

- We formulate the task of selecting reasoning triplets from a knowledge graph as a reinforcement learning problem, allowing the model to adaptively choose high-quality triplets based on semantic relevance, structural connectivity, and the question.
- We introduce *KG-Scout* , a lightweight and efficient dual-tower policy network that combines semantic and structural signals using a learned gating mechanism. **Work submitted in *K-CAP'25***
- Our evaluation of *KG-Scout* on two KGQA datasets, WebQSP and CWQ , shows improvement in performance metrics over several competing baselines.

CHAPTER 2

Preliminaries and Related works

This chapter lays the groundwork necessary for understanding the proposed research by introducing key concepts and reviewing relevant prior works. It begins by defining core terminologies within the domain of knowledge graphs and question answering. Subsequently, it transitions into a discussion of existing literature, highlighting both recent advancements and challenges that motivate the contributions of this thesis.

2.1 Preliminaries

This section provides a concise overview of the fundamental concepts that are essential for comprehending the technical contributions presented in this thesis.

2.1.1 Knowledge Graphs (KGs)

Knowledge Graphs are a structured way to represent factual knowledge, organized as a network of interconnected entities. They are formally defined as a set of triplets, $\mathcal{G} = \{(e, r, e') \mid e, e' \in \mathcal{E},\ r \in \mathcal{R}\}$ where $\mathcal{E}$ represent set of all the entities and $\mathcal{R}$ represents set of all the relations. Each triplet (e, r, e') represents a fact, stating that a head entity e is connected to a tail entity e' via a directed relation r. This structured format provides a clear and unambiguous way to store complex information, making it an excellent source for grounding information.

2.1.2 Personalized Page Rank (PPR)

Personalized PageRank (Haveliwala, 2002) is an algorithm used to measure the importance of nodes within a graph from a specific starting point. In the context of KGQA, PPR is a powerful technique for identifying relevant candidate entities. The process involves:

- Seed Initialization: The algorithm is initialized by identifying the entities mentioned in the question as seed entities.

- Random Walks: The PPR score is then computed through a series of random walks that restart at the seed entities with a certain probability.
- Score Propagation: As the random walks traverse the KG, the relevance score is propagated from the seed entities to their neighbors.

Nodes that are structurally close to the seed entities receive higher PPR scores, marking them as important candidates that are likely to be part of the reasoning path and potentially lead to the final answer.

2.1.3 Knowledge Graph Question Answering (KGQA)

Knowledge Graph Question Answering (KGQA) is the task of finding answers to natural language questions by querying a KG. Given a query q, the goal is to identify a set of supporting facts (triplets) from the KG and use them to derive the correct answer entity, a, which is a member of the KG's entity set $\mathcal{E}$. KGQA can be categorized by the complexity of the questions it handles. Simple questions can be answered with a single triplet, whereas complex questions require reasoning over multiple triplets, forming a multi-hop path. The challenge in KGQA, especially for large-scale and densely connected KGs, lies in efficiently and accurately selecting the most relevant triplets that form the reasoning path to the answer.

Recent research has explored integrating Large Language Models (LLMs) with KGs to enhance KGQA. By providing the LLM with relevant KG triplets as a context, we can leverage its reasoning capabilities while mitigating its tendency for factual inaccuracies or "hallucinations." The LLM can then synthesize the provided facts to generate a coherent and factually grounded answer.

2.2 Related works

This section reviews the active and evolving field of research that integrates Knowledge Graphs (KGs) with Large Language Models (LLMs). The primary motivation for this integration is to overcome the inherent limitations of LLMs, specifically their susceptibility to factual inaccuracies and their tendency to generate incorrect information. We'll explore the recent methodologies researchers have developed to leverage the structured and verifiable knowledge found in KGs to enhance LLM capabilities. These approaches can be broadly categorized by their distinct strategies for retrieving relevant information from the KG and how the LLM subsequently uses this data to produce grounded answers.

Different systems employ distinct strategies for retrieving knowledge from graphs. Some researchers make the LLM to act as an active agent, dynamically exploring the KG. For instance, Think-on-Graph(Sun *et al.*, 2023) treats the LLM as an agent that interactively navigates the KG through an iterative beam search, discovering and evaluating reasoning paths composed of connected facts. Other approaches rely on combinatorial optimizations or fixed heuristics for retrieval. G-Retriever(He *et al.*, 2024) introduces a novel Retrieval-Augmented Generation (RAG) method for textual graphs, formulating subgraph retrieval as a Prize-Collecting Steiner Tree optimization problem to extract relevant, connected subgraphs efficiently. DeCAF(Yu *et al.*, 2022) uses sparse and dense retrieval techniques to get get most relevant queries from KG. Similarly, StructGPT(Jiang *et al.*, 2023) utilizes "specialized interfaces" to collect relevant evidence from structured data, employing "Extract_Neighbor_Relations" and "Extract_Triples" procedures where the LLM orchestrates these pre-defined procedures to gather information. EWEK-QA(Dehghan *et al.*, 2024) prioritizes efficiency by incorporating KG triples directly into its pipeline, explicitly avoiding LLM calls during the retrieval phase, which points to using fixed, non-LLM-driven mechanisms.

Once retrieved, this information is then used for reasoning and generation: Think-on-Graph directly uses the sequential reasoning paths to guide the LLM's answer formulation, G-Retriever transforms the retrieved subgraph into a textual format and uses a "graph token" as a soft prompt to guide the LLM's generation, helping to prevent factual errors and StructGPT iteratively refines its reasoning based on the collected and linearized evidence. These highlights the ongoing researches to find the relevant information from KG.

CHAPTER 3

Methodology

"The purpose of search is to find something you don't know you were looking for."

- Gemini

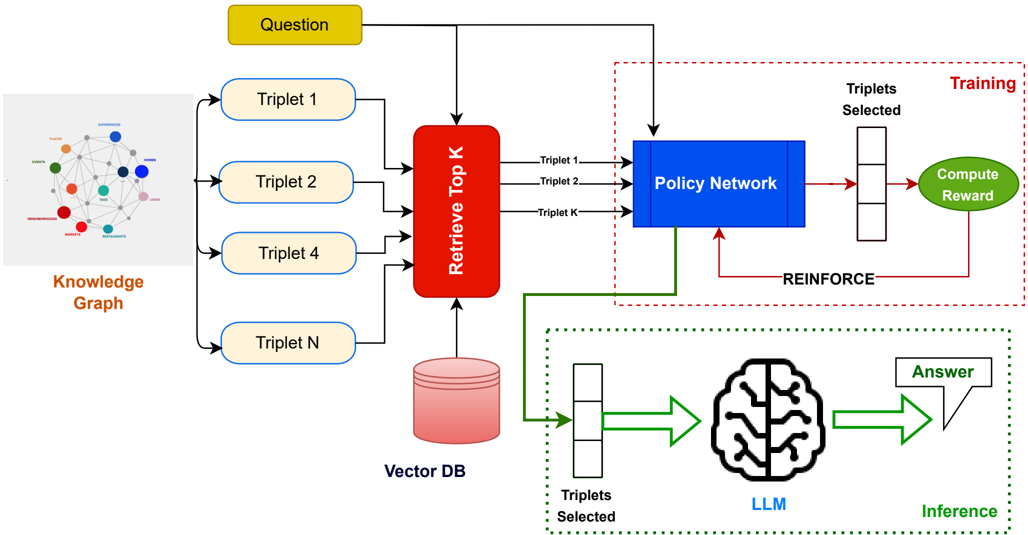

Figure 3.1: **Architecture of *KG-Scout***

This chapter provides a detailed exposition of the design and implementation of *KG-Scout* , our novel framework for mitigating factual inaccuracies of LLM by precisely identifying and selecting pertinent triplets from Knowledge Graphs (KGs). We start by presenting an high-level view of the methodology, explaining the relationship between our initial semantic pre-filtering step and the core reinforcement learning based selection mechanism. Then we present the the prompting strategy used for finding answers via LLM. Subsequently, we formalize the triplet selection problem as a contextual bandit, defining its constituent elements: the state space, action space, and reward function. The chapter then thoroughly details the architecture of our policy network, highlighting its dual-tower design and adaptive gating mechanisms, which are central to its ability to reason over both semantic and structural cues.

3.1 Overview of *KG-Scout*

As shown in Figure 3.1, we start by ranking the knowledge graph triplets based on how closely their meanings match the question, using cosine distance to measure their similarity. This step helps filter out less relevant triplets at the beginning, so we can focus on the ones more likely to be useful based on their semantic similarity to the question. Although cosine similarity captures the semantic closeness, it does not fully account for the structural patterns of KGs or the multi-hop reasoning required to reach an answer. Therefore, we use this sorted set as input to the policy network, which is trained to refine this ranking by learning to select the triplets most relevant to answer using a task-specific reward signal. This approach balances computational efficiency and learning effectiveness by narrowing the search space early on while still allowing the model to learn nuanced path-selection strategies beyond what simple similarity can provide. We use LLM to find answer a by including the linearized list of triplets in the prompt. This helps the LLM to refer to these triplets derived from KG and derive the answer a using the triplets, thereby minimizing the factual inaccuracies.

3.2 Prompting Strategy

Prompt Template

You are a knowledge graph question answering system. Given a question and relevant linearized knowledge triplets, provide correct answers in JSON format supported by the triplets and provide a brief reason for the answer based on the triplets.

Instructions:

- Provide your final answer in JSON format using this placeholder: `{"ans":["your answer 1","your answer 2"]}`. If there is insufficient information to answer the question, return `{"ans": ["answer not available"]}`.
- Ensure the answer does not contain duplicate entries.
- Keep your reasoning brief and focused.
- Your answer must not contradict any information presented in the provided triplets.
- If the answer is directly supported by the triplets, use the triplets to justify your answer.
- If the answer is not explicitly found in the triplets, you may use your own factual knowledge but only if it is consistent with the information in the triplets.

Example 1:

Triplets:

- Lou Seal, sports mascot team, San Francisco Giants
- San Francisco Giants, sports sports team championships, 2012 World Series
- Crazy Crab, sports mascot team, San Francisco Giants
- ...

Question: What year did the team with mascot named Lou Seal win the World Series?

Answer: Answer in JSON format:

`{"ans" : ["2014 (2014 World Series)", "2012 (2012 World Series)", "2010 (2010 World Series)"]}`

Reason:

- Identify the team with the mascot Lou Seal.
- Find the years that team won the World Series.
- From the triplets:
  - Lou Seal is the mascot of the San Francisco Giants.
  - San Francisco Giants won the World Series in 2010, 2012, and 2014.

Therefore, the team with mascot Lou Seal won the World Series in 2010, 2012, and 2014.

Now consider the below Triplets and answer the Question carefully.

Question: `<User Question>`

Triplets: `<Triplets provided by` *`KG-Scout`* `>`

3.3 Task Formulation and Methodology Outline

Given a KG, $G = \{(e, r, e') \mid e, e' \in \mathcal{E},\ r \in \mathcal{R}\}$ whose information is presented as a set of triplets, the task is to find a subset of these triplets or, more formally, a subset of the graph $\mathcal{G} \subset G$ which are most relevant to the question q and which may help to arrive at the answer a. The entities of question denoted by $\mathcal{E}q$ are already assumed to be present. In the subsequent sections we will see the formulation of triplet selection as a reinforcement learning problem and various terminologies associated with rl. Subsequently we will deep dive into the details architectural details of *KG-Scout* and its training.

The central task addressed in this work is to identify a relevant subgraph $\mathcal{G} \subset G$ from a given knowledge graph (KG) that can be used to answer a specific question q. The KG is formally defined as a set of triplets, $G = \{(e, r, e') \mid e, e' \in \mathcal{E},\ r \in \mathcal{R}\}$. We assume that the entities mentioned in the question, denoted by $\mathcal{E}q$ are successfully extracted. The following sections will introduce our formulation of this triplet selection problem as a reinforcement learning task, defining the associated terminologies. Subsequently, we will deep dive into our model's architecture and the methodology for its training.

3.3.1 A Reinforcement Learning Approach via Contextual Bandits

We formulate the triplet selection problem as a Contextual Bandit problem. This can be conceptualized as a single-step Markov Decision Process (MDP), denoted by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{R})$. Our goal is to train a policy π_θ , parameterized by a neural network, to select the most relevant triplets (actions) given the question (context). This framework allows us to optimize the triplet selection process by maximizing a reward signal that is directly correlated with the quality of the final answer. The following sections provide a detailed exposition of this approach.

3.3.2 Description of State and Action

State $\mathcal{S}$:

The state $s_t \in \mathcal{S}$ captures the complete context in the decision step t, where

$$s_t = \mathbf{q}, E_q, \mathbf{E}_{\text{triplets}}, \mathbf{E}_{\text{relations}}, E_a, \mathbf{G}$$

where $\mathbf{q} \in \mathbb{R}^d$ represents the embedding of the question with $d = 384$ dimensions. $E_q = e_1^q, e_2^q, \ldots, e_{|E_q|}^q$: represents the entities extracted from the question. $\mathbf{E}_{\text{triplets}} \in \mathbb{R}^{N \times d}$ represents the embeddings of N linearized candidate triplets. $\mathbf{E}_{\text{relations}} \in \mathbb{R}^{N \times d}$ represents the embeddings of relation phrases from the corresponding triplets. $E_a = e_1^a, e_2^a, \ldots, e_{|E_a|}^a$ represents the ground truth answer entities. $\mathbf{G} \in \mathbb{R}^{N \times 2}$: represents the PPR scores of the head and tail entities of each candidate triplet.

We concatenate the triplet (e, r, e') using a space character to linearize it. We also replace all special characters of the relation r with space characters. Let us consider the example ("Earth", "revolves around", "Sun"). The linearized form becomes "Earth revolves.around Sun", which is then encoded to produce $\mathbf{E}_{\text{triplets}}[i]$. Similarly, the encoding of the relation phrase "revolves around" generates $\mathbf{R}_{\text{relations}}[i]$. This combined representation allows the policy network π_θ to weigh the importance of both the relation and the entities while generating a relevance score for a triplet.

Action $\mathcal{A}$

The policy network π_θ performs batch selection rather than sequential decision making. Let us consider N candidate triplets; then the action space consists of selecting a subset of k triplets from N available triplets. Formally, let $\mathcal{A}$ be the action space and $\{t_1, t_2, \cdots t_N\}$ be the number of candidate triplets, then $A = \{S \subset \{t_1, t_2, \cdots t_N\} | \, |S| = k\}$, where each action corresponds to selecting k distinct triplets simultaneously. The selection process uses categorical sampling without replacements

$$\pi_\theta(a_t|s_t) = \text{Categorical}(\text{softmax}(\pi_\theta(s_t)/\tau))$$

where $\pi_\theta(s_t) \in \mathbb{R}^N$ produces relevance scores for all candidate triplets, and τ is a learnable temperature parameter that controls the exploration-exploitation trade-off during sampling.

3.3.3 Description of Reward

We have designed a reward function $\mathcal{R}$ that comprehensively assesses the quality of the triplets selected by the policy network. The functions construct a directed graph $\mathcal{G} = (V, E)$ from the selected triplets and compute a weighted combination of the evaluation components. Given a set of triplets $\mathcal{T} = \{(s_i, r_i, o_i)\}_{i=1}^N$, a set of question entities $\mathcal{E}_q = \{e_q^1, \ldots, e_q^m\}$ provided, and a set of answer entities $\mathcal{E}_a = \{e_a^1, \ldots, e_a^n\}$ for a given question q, we evaluate the triplets using four major criteria i.e., presence of answer entities, connectivity between the question and answer entities, path efficiency and path coverage. The overall reward is calculated as:

$$\mathcal{R} = w_{\text{pres}} \cdot \mathcal{R}_{\text{pres}} + w_{\text{conn}} \cdot \mathcal{R}_{\text{conn}} + w_{\text{eff}} \cdot \mathcal{R}_{\text{eff}} + w_{\text{cov}} \cdot \mathcal{R}_{\text{cov}}$$

where, $w_{\text{pres}}, w_{\text{conn}}, w_{\text{eff}}, w_{\text{cov}}$ are the scalar weights (which are set to 2, 4, 1, and 3 respectively in our experiments). We choose these weights so that finding a reasoning path has the maximum weightage in reward, followed by the presence of answer entities, coverage, and efficiency, respectively.

Answer-Presence Score ($\mathcal{R}_{\text{pres}}$)

This component calculates the fraction of answer entities that are present in the triplets selected by the policy network π_θ. If $\mathcal{V}_\mathcal{G}$ denotes the set of all entities present in $\mathcal{G}$, and let E_a represent the set of answer entities, then :

$$\mathcal{R}_{\text{presence}} = \frac{|\mathcal{E}_a \cap \mathcal{V}_\mathcal{G}|}{|\mathcal{E}_a|}$$

This approach encourages the policy network π_θ to retrieve triplets containing as many ground-truth answer entities as possible.

Graded Connectivity Score ($\mathcal{R}_{\text{conn}}$)

This component captures whether there is a path between a question entity and an answer entity, and how long that path is. Consider a pair that contains a question entity e_q and an answer entity e_a . If we represent the path length between those entities as $d(e_q, e_a)$, then the connectivity score is calculated as follows:

$$\mathcal{R}_{\text{connectivity}} = \max_{e_q \in \mathcal{E}_q,\, e_a \in \mathcal{E}_a} \left[\max\left(0, 1 - \lambda_{\text{lin}} \cdot (d(e_q, e_a) - 1)\right)\right]$$

where λ_{lin} is the linear decay hyperparameter (set to 0.2 in our experiments). The above approach encourages the policy network π_θ to select triplets whose combination creates a reasoning path between (e_q, e_a) of the shortest length and discourages long paths.

Efficiency Score ($\mathcal{R}_{\text{eff}}$)

The efficiency score rewards short and efficient paths. Consider a question entity e_q and an answer entity e_a; then the efficiency score is calculated as the inverse of the path length between e_q and e_a.

$$\mathcal{R}_{\text{eff}} = \max_{e_q \in \mathcal{E}_q,\, e_a \in \mathcal{E}_a} \left[\frac{1}{1 + d(e_q, e_a)} \right]$$

This reward component favors direct and minimal hop connections within the graph.

Path Coverage Score ($\mathcal{R}_{\text{cov}}$)

The path coverage score measures how well the selected triplets 'cover' the true reasoning paths between the question and answer entities. For each shortest path $P: \ e_q = v_0 \to v_1 \to \cdots \to v_k = e_a$ of length k, we count the number of hops along P for which we have a corresponding selected triplet (v_i, r, v_{i+1}). Formally, we define coverage P as:

$$\text{coverage}(P) \ = \ \frac{\big|\{\, i : 0 \leq i < k,\ (v_i \to v_{i+1}) \in \mathcal{E}_{\mathcal{T}} \}\big|}{k}.$$

where,

$$\mathcal{E}_{\mathcal{T}} = \{\, s_i \to o_i \mid (s_i, r_i, o_i) \in \mathcal{T} \}.$$

We then max over all the shortest paths P between e_q and e_a:

$$\mathcal{R}_{\text{cov}}(e_q, e_a) \ = \ \max_{P \,\in\, \mathcal{P}(e_q, e_a)} \text{coverage}(P)$$

A higher $\mathcal{R}_{\text{cov}}$ indicates that a larger fraction of each reasoning path is supported by the selected triplets.

3.3.4 Policy Network

The policy network as shown in Figure 3.2 accepts the embedding of the question as input, a set of encoded candidate triplets, their associated relation embeddings, and structural graph-based features generated using Personalized Page Rank. The network incorporates two attention modules to model the interaction between the question and the candidate triplets: one operating over the triplet representations, and the other over the relation embeddings. Let $\mathbf{q} \in \mathbb{R}^d$ represent the question generated through a frozen

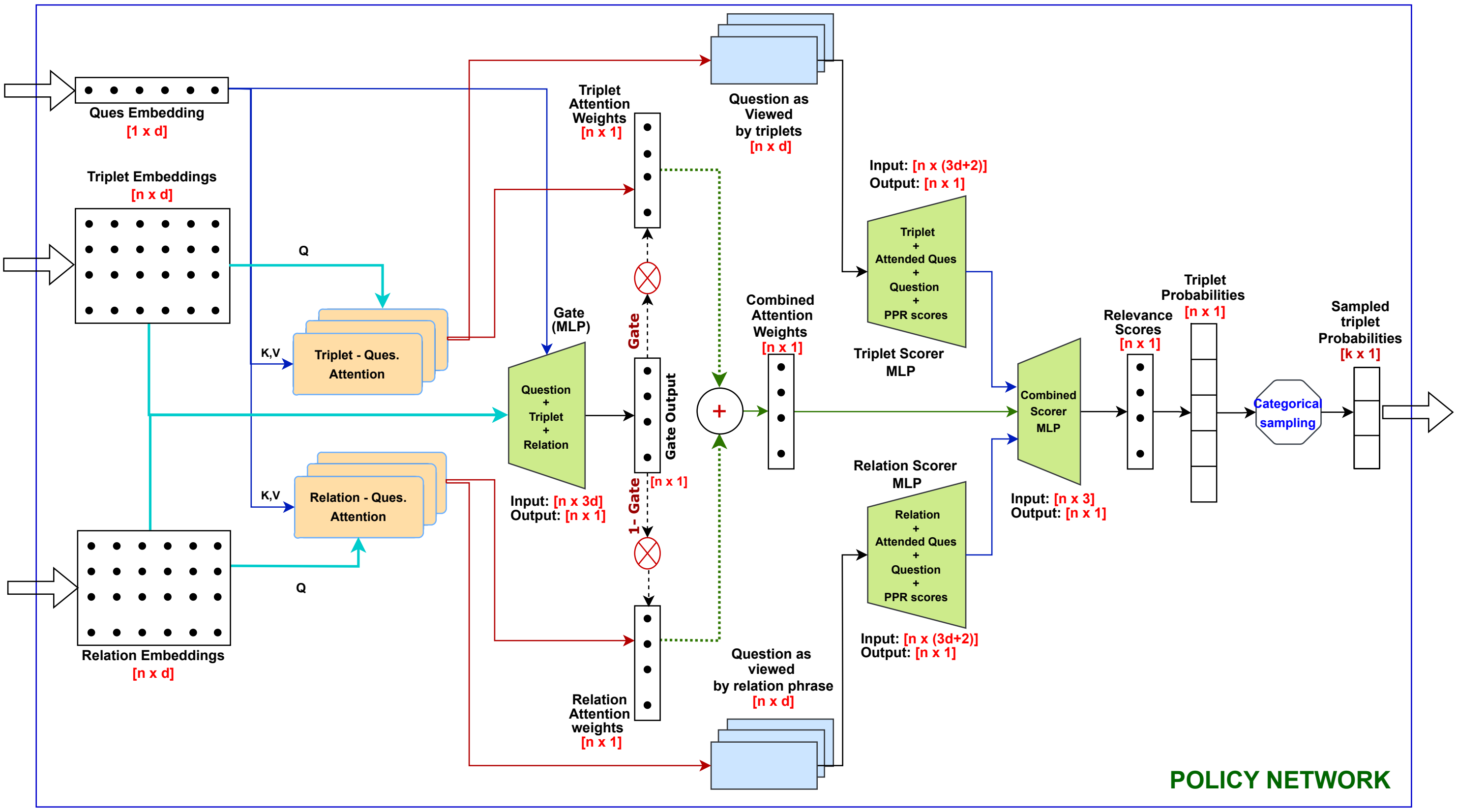

Figure 3.2: **Architecture of the Policy Network.**

encoder Sentence Bert. The matrix of linearized triplet embeddings is denoted by $\mathbf{T} \in \mathbb{R}^{nxd}$. The matrix of corresponding relation embeddings is $\mathbf{R} \in \mathbb{R}^{nxd}$, where n is the number of candidate triplets. $\mathbf{T}$ and $\mathbf{R}$ are passed through two multi-head attention blocks parallel to the question vector $\mathbf{q}$ serving as the value and $\mathbf{T}$ and $\mathbf{R}$ serving as the key and value in the respective blocks. Both attention blocks output the view of the question as seen by triplets $\mathbf{Q}_{trip} \in \mathbb{R}^{nxd}$ and as seen by relation phrases $\mathbf{Q}_{rel} \in \mathbb{R}^{nxd}$, respectively.

The model incorporates a dynamic gating mechanism to modulate the influence of triplet-centric versus relation-centric cues. A feedforward network GateMLP : $\mathbb{R}^{3d} \to \mathbb{R}$ receives a concatenation of $\mathbf{T}_i$, $\mathbf{R}_i$, and $\mathbf{q}$ for each candidate $\mathbf{i}$, and outputs a scalar gate $\mathbf{g}_i \in [0, 1]$. This value serves as a soft weight that indicates how much the final score should emphasize the structural coherence of the path versus its semantic alignment with the question.

The network implements a two-tower scoring scheme. The first tower, referred to as the triplet-centric tower, aggregates the candidate's original triplet embedding $\mathbf{T}_i$, the representation of the question as seen by $\mathbf{T}_i$ *i.e.* $\mathbf{Q}_{trip_i}$, the question embedding $\mathbf{q}$, and graph feature scores *i.e.* **PPR** scores into a input vector $x_i^{(A)} \in \mathbb{R}^{3d+2}$ which is scored by a multi-layer perceptron (MLP). The second tower, relation-centric, uses an analogous structure, but replaces the triplet features with $\mathbf{R}_i$ and$\mathbf{Q}_{rel_i}$. Each tower outputs a scalar score: $s_i^{(A)}$ and $s_i^{(B)}$, respectively, where i is the i^{th} candidate and A and B are the attention blocks.

These scores are combined with the gate $\mathbf{g}_i$ into a 3-dimensional vector denoted by $z_i = [s_i^{(A)}, s_i^{(B)}, g_i] \in \mathbb{R}^3$, which is passed through a shallow MLP (the combiner network) to produce a final scalar score $\hat{s}_i \in \mathbb{R}$ for each candidate path. This score brings together how well a triplet's content matches the question (semantic alignment), how well it connects to the knowledge graph (structural connectivity), and how much weight the model gives each using a learned gating mechanism. Triplets with higher scores are more likely to contribute to meaningful reasoning paths by balancing relevance and structure. These scores are subsequently converted to selection probabilities through temperature-controlled softmax, enabling the model to sample diverse yet high-quality triplets answering a query.

3.3.5 Training

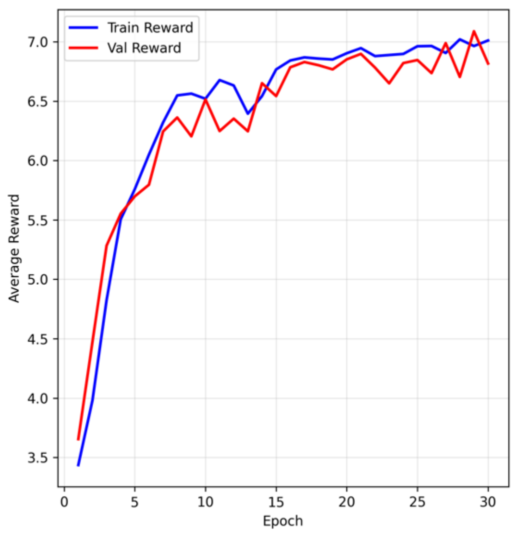

Figure 3.3: Average training and validation rewards as a function of training epochs on the WebQSP , showing progressive improvement.

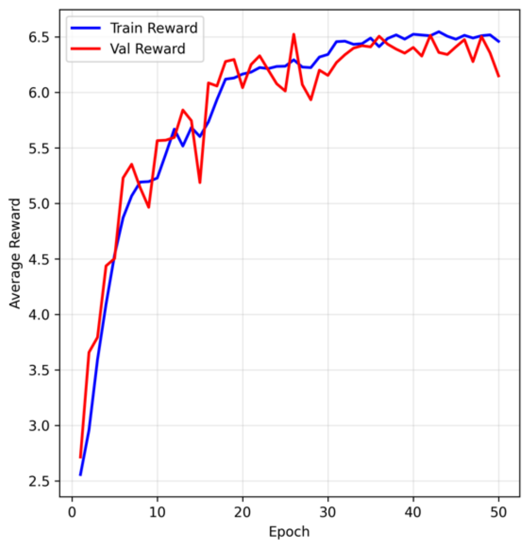

Figure 3.4: Average training and validation rewards as a function of training epochs on the CWQ , showing progressive improvement.

The model converts the final path scores $\hat{s}_i$ into a policy by scaling them with a learnable temperature parameter $\tau > 0$, and then applies the softmax function. This process produces a probability distribution over the candidate triplets.

$$\pi_\theta(a_i \mid s_t) = \frac{\exp(\hat{s}_i/\tau)}{\sum_{j=1}^{n} \exp(\hat{s}_j/\tau)}.$$

For the policy network described above, we want to look for parameters θ, which maximize the expected reward. We train the policy using the REINFORCE(Williams, 1992) algorithm. To reduce the variance of the gradients,a baseline is incorporated during training. The update is given by:

$$\nabla_\theta J(\theta) = \mathbb{E}_{a\sim\pi_\theta}\left[(R - b)\nabla_\theta \log \pi_\theta(a \mid s_t)\right]$$

where R is the reward signal obtained from the environment, and b is the baseline used.

CHAPTER 4

Experiments and Evaluations

Following the detailed explanation of our method in the last chapter, this chapter focuses on the empirical validation of our proposed framework. We begin by describing the datasets used to evaluate *KG-Scout* along with some important statistics. Then, we'll cover the experimental setup, including the other models we compared our system to, the specific ways we measured success, and the settings for our experiments. After that, we'll share the main results and analyze them, proving that our model performs better than existing methods.

4.1 Datasets

We evaluate the retrieval capability of *KG-Scout* on two KGQA datasets, WebQSP (Yih *et al.*, 2016) and CWQ (Talmor and Berant, 2018). Both of these datasets use FreeBase (Bollacker *et al.*, 2008) KG as background knowledge. During training, we exclude samples whose answer entities are absent from the KG triplets. At test time, however, *KG-Scout* is evaluated on the full test sets of both datasets regardless of the availability of the answer. We train *KG-Scout* on a very small subset of the CWQ dataset. We randomly sample 3000 data points from CWQ that have an answer entity in the associated triplets. This is done to save training time. Refer to Table 4.1 for details on sample counts.

Samples	**WebQSP**		**CWQ**	
	Train	**Test**	**Train**	**Test**
Total Samples	3098	1639	27734	3531
Samples Used	2717	1639	3000	3531

Table 4.1: **Dataset statistics used for evaluating *KG-Scout***

4.2 Implementation Details

We use the LLAMA-3.1-8B (Grattafiori *et al.*, 2024) model as a reasoner for our experiments. For generating embeddings of questions, linearized triplets, and relation phrases, we use the Sentence-Bert(Reimers and Gurevych, 2019) *all-MiniLM-L6-v2* model. We implement all graph-related operations using the NetworkX library, where each triplet (s, r, o) adds a directed edge from s to o with relation r. All entity and relation strings are normalized to lowercase to ensure consistency. To compute PPR scores, we use the built-in implementation provided by the NetworkX library. We train *KG-Scout* for 30 epochs in WebQSP and 50 epochs in CWQ . During training, we calculate the cosine similarity between the question and each triplet in the knowledge graph. Based on this, we select the top–$k = 1000$ most semantically similar triplets and feed them into *KG-Scout* , which then ranks and selects the top-$k = 100$ (k = 100 aligns with the baseline DeCAF, KD-CoT and cosine similarity). triplets for further processing. We follow the same procedure at test time: we retrieve the top-100 triplets relevant to the input question using from *KG-Scout* . These triplets and the question are then passed to an LLM to get the final answer. The temperature parameter is set to 0.1 to reduce the diversity of generation. To generate answers, we adopt in-context learning (ICL(Brown *et al.*, 2020)) to guide LLM to find answers. Prompting details are mentioned in 3.2

4.3 Performance Comparison

We compare our approach with six baselines that use LLMs to find triplets from KG to answer. We have used Hits and Macro F1 as an evaluation metric. The baseline have claimed to have reported Hits@1 but inadvertently they actually compute Hit, that measures whether LLM can generate atleast one correct answer. The results shown in Table 4.2 indicate that our method achieves the best performance on the WebQSP dataset. Compared to the G-retriever (He *et al.*, 2024) that uses combinatorial optimization to find the most suitable subgraph, our method improves the Hit ratio by 9.37% on the WebQSP dataset. Compared to DeCAF (without logical forms) which uses a combination of BM25(Robertson *et al.*, 2009) and DPR(Karpukhin *et al.*, 2020) for KG retrieval, our method improves the Hit ratio $\leq 8.78\%$ and F1 by $\leq 25.8\%$ on WebQSP and $\leq 4.9\%$ on the hit ratio of CWQ .

	Method	k	WEBQSP		CWQ	
			Hit	Macro-F1	Hit	Macro-F1
w/o KG	Llama-3.1-8B	0	66.50	45.93	40.07	33.42
	Qwen-2.5-7b	0	48.96	32.16	17.13	13.37
	Deepseek-R1-Llama-8b	0	50.80	32.13	30.27	25.83
w/ KG	StructGPT (Jiang *et al.*, 2023)	itr.	72.6	-	-	-
	ToG (Sun *et al.*, 2023)	itr.	57.6	-	**68.9**	-
	KD-CoT (Wang *et al.*, 2023)	100	68.6	52.5	55.7	-
	DeCAF(ans only) (Yu *et al.*, 2022)	100	74.2	49.5	47.9	-
	G-Retriever (He *et al.*, 2024)	-	73.79	-	-	-
	EWEK-QA w/ KG (Dehghan *et al.*, 2024)	itr.	59.9	-	40.1	-
	Cosine + Llama-3.1-8B	100	75.98	58.17	45.62	34.63
	Cosine + Qwen-2.5-7b	100	65.97	51.68	34.83	28.28
	Cosine + Deepseek-R1-Lama-8b	100	64.43	47.33	37.92	31.40
	KG-Scout + Llama-3.1-8B	100	**80.71**	**62.30**	50.07	**36.89**
	KG-Scout + Qwen-2.5-8B	100	71.50	56.36	35.63	28.86,
	KG-Scout + Deepseek-R1-Llama-8b	100	68.80	50.14	41.74	34.54

Table 4.2: **Comparison on WEBQSP and CWQ Datasets**.

In the CWQ data set, which is more challenging due to the inclusion of more than 3 hop questions, *KG-Scout* achieves competitive performance with KD-CoT and ToG.

Evaluation for different K values

We conducted a study Table 4.3 to evaluate the impact of varying the retrieval size k on performance, while keeping the reasoner model fixed as LLaMA-3.1-8B across all settings. The experiments were carried out for $k \in \{30, 50, 100, 150\}$ on both WEBQSP and CWQ datasets, comparing *KG-Scout* with a Cosine similarity baseline. Across all k values, *KG-Scout* consistently outperforms Cosine retrieval on both WEBQSP and CWQ datasets in terms of Hit and Macro-F1 scores. The performance gains are highest for k=100, with WEBQSP showing up to a +6.22 and +7.13 improvement in Hit and Macro-F1 respectively and CWQ achieving a maximum of +9.75 and +6.52 in Hit and Macro-F1 respectively. While the margins narrow for smaller k, *KG-Scout* maintains a consistent advantage, highlighting its robustness across varying retrieval set sizes. We also see that the gains using *KG-Scout* are significant on CWQ as compared to WebQSP . This can be attributed to that fact that CWQ includes comparatively larger portion of question which requires multiple hops to answer.

Method	k	WEBQSP		CWQ	
		Hit	Macro-F1	Hit	Macro-F1
KG-Scout	150	80.16	61.17	48.80	36.32
Cosine	150	76.17	57.46	46.45	34.71
Δ		↑5.23%	↑6.45%	↑5.05%	↑4.63%
KG-Scout	100	80.71	62.32	50.07	36.89
Cosine	100	75.98	58.17	45.62	34.63
Δ		↑6.22%	↑7.13%	↑9.75%	↑6.52%
KG-Scout	50	75.61	58.39	45.28	34.66
Cosine	50	75.25	57.30	40.41	31.20
Δ		↑0.47%	↑1.90%	↑4.87%	↑3.46%
KG-Scout	30	75.00	57.77	38.96	30.36
Cosine	30	73.83	56.19	36.70	28.82
Δ		↑1.58%	↑2.81%	↑6.16%	↑5.34%

Table 4.3: **Comparison of *KG-Scout* and Cosine retrieval for different K values.** LLama-3.1-8B is used as reasoner LLM in all the cases.

Retriever Evaluation

Dataset	Metrics	$K = 30$			$K = 50$			$K = 100$			$K = 150$		
		KG-Scout	Cosine	Δ	*KG-Scout*	Cosine	Δ	*KG-Scout*	Cosine	Δ	*KG-Scout*	Cosine	Δ
WebQSP	Ans present	74.42%	62.76%	↑18.57%	75.34%	65.57%	↑14.9%	81.20%	70.15%	↑15.75%	84.19%	74.11%	↑13.6%
	Path exists	68.19%	61.48%	↑11.73%	71.61%	64.59%	↑10.87%	77.11%	69.47%	↑10.99%	81.68%	73.26%	↑11.49%
CWQ	Ans present	49.11%	37.23%	↑31.90%	59.34%	42.84%	↑38.51%	64.42%	51.40%	↑25.33%	63.68%	56.37%	↑12.96%
	Path exists	44.63%	35.44%	↑25.93%	54.18%	41.51%	↑30.52%	61.44%	50.41%	↑21.88%	63.68%	55.57%	↑14.05%

Table 4.4: **Comparison of *KG-Scout* , Cosine, and Δ for different K values.**

We evaluate the effectiveness of triplet selection by comparing *KG-Scout* with a cosine similarity–based ranking baseline on two benchmark datasets: WEBQSP and CWQ. As shown in Table 4.4, *KG-Scout* consistently selects triplets that are more relevant and informative, leading to higher answer entity presence and reasoning path coverage. This behavior is observed across multiple values of $k \in \{30, 50, 100, 150\}$. Specifically for k=100, *KG-Scout* achieves relative improvements of 15.7% in answer entity presence and 10.99 % in reasoning path existence over the cosine similarity baseline. On CWQ, these improvements increase to 25.3 % and 21.88 %, respectively. These results demonstrate that *KG-Scout* produces triplets that more effectively support large language models in constructing complete reasoning chains and retrieving accurate answers.

Generalization Study

To rigorously evaluate the generalization capability of *KG-Scout* on entirely unseen data, we conducted cross-dataset experiments. Specifically, we trained *KG-Scout* on WebQSP and evaluated its performance on CWQ , and, in a complementary setting, trained on CWQ and tested on WebQSP . This setup allows us to examine how well the model transfers knowledge across datasets with potentially different distributions and characteristics.

Method	**k**	**WEBQSP**		**CWQ**	
		Hit	**Macro-F1**	**Hit**	**Macro-F1**
KG-Scout *	100	78.56	61.47	46.22	35.53

Table 4.5: **Cross-dataset generalization performance of *KG-Scout* *.** Here, * indicates that the metrics reported for WebQSP are obtained from a model trained on CWQ , and vice versa. LLama-3.1-8b is used as reasoner LLM.

Dataset	**Metrics**	$K = 100$		
		***KG-Scout* ***	***KG-Scout***	**cosine**
WebQSP	Ans present	77.23%	81.20%	70.15%
	Path exists	74.60%	77.11%	69.47%
CWQ	Ans present	59.97%	64.42%	51.40%
	Path exists	49.76%	61.44%	50.41%

Table 4.6: **Retrieval performance on cross-dataset.** Here, * indicates that the metrics reported for WebQSP are obtained from a model trained on CWQ , and vice versa.

From Table 4.5, we observe that *KG-Scout* achieves a Hit score of 78.56 and Macro-F1 of 61.47 when trained on CWQ and evaluated on WebQSP . Conversely, when trained on WebQSP and tested on CWQ , it achieves a Hit score of 46.22 and Macro-F1 of 35.53. When compared against the in-domain retrieval results reported in Table 4.3, it is clear that cross-dataset performance degrades slightly which is expected. For instance, KG-Scout with $k = 100$ achieves 80.71 Hit, 62.32 Macro-F1 on WebQSP and 50.07 Hit, 36.89 Macro-F1 on CWQ in the in-domain setting, both of which are higher than the corresponding cross-dataset scores of *KG-Scout* . This implies that KG-Scout is able to generalize across datasets. As seen in the retrieval performance in Table 4.6, a more noticeable decline happens in CWQ dataset. In CWQ , many data points require reasoning over paths longer than 3 hops to reach the answer entity, whereas in WebQSP , the answer entities are at most 2 hops away. As a result, a model trained on CWQ generalizes well to WebQSP , but a model trained on WebQSP struggles to generalize effectively on CWQ .

4.4 Discussions and Conclusion

Advantages

The empirical results demonstrate that our *KG-Scout* + Reasoner framework achieves robust performance on both the WEBQSP and CWQ datasets, notably outperforming several recent approaches, particularly on WEBQSP. Compared to methods such as KD-CoT, StructGPT, and Think-on-Graph (ToG), our approach presents distinct advantages in terms of efficiency, generalization with limited supervision, and scalability.

A primary strength of our framework stems from its lightweight design. While methods like KD-CoT and StructGPT necessitate the training of large-scale retrievers, complex multi-stage pipelines, or iterative LLM-driven interfaces requiring extensive supervision, *KG-Scout* employs a simple yet highly effective two-stage strategy. It first prunes the search space by filtering knowledge graph triplets using semantic similarity and then refines this selection through a compact policy network. It is particularly noteworthy that our policy network was trained on only 3,000 examples, randomly sampled from the 27,000 available in the CWQ dataset. This highlights our model's exceptional ability to generalize from minimal supervised data, in stark contrast to competitors like KD-CoT and StructGPT which typically rely on full training datasets and significantly larger models. Consequently, *KG-Scout* is highly scalable and cost-effective, making it a viable solution for low-resource environments.

Furthermore, by providing the LLM with linearized KG triplets, our model is able to maintain strong factual consistency while leveraging the LLM's inherent reasoning capabilities. Our results on WEBQSP (Hit@1: 80.71%, Macro-F1: 62.30%) surpass those of KD-CoT, StructGPT, and DeCAF (without a logical form). This substantiates the claim that a carefully curated KG context, even within a relatively simple architectural framework, can yield competitive or superior performance.

While ToG achieves higher scores on CWQ, this is accomplished at the cost of significant computational overhead. ToG performs an iterative beam search across KG paths, requiring multiple LLM queries per instance during inference. In contrast, *KG-Scout* performs all retrieval and reasoning in a single pass, making it considerably more suitable for real-world applications where latency and cost are critical factors. Additionally, while ToG is a training-free framework, its reliance on the quality of KG traversal may not always align with the semantic nuances of natural language questions. *KG-Scout* , by explicitly incorporating both semantic and structural cues into its policy network, offers a balanced and both effective and efficient alternative.

Limitations

Despite its strengths, our current design is not without limitations. Even though KG-Scout successfully retrieves triplets that contain the answer entities and establish coherent reasoning chains, the reasoning LLM occasionally fails to effectively interpret this information, leading to incorrect answers. These instances are examined in section 4.5. Since KG-Scout requires retrieving the top 1,000 triplets ranked by cosine similarity, it becomes necessary to store the embeddings of the triplets in a vector database to support efficient inference. However, this approach may incur high storage costs.

Conclusion

This paper presents *KG-Scout* , a novel framework for knowledge graph data retrieval designed to perform targeted extraction of relevant facts. By providing downstream LLMs with this carefully selected data, our approach effectively mitigates the issue of factual hallucinations. The retrieval model itself is lightweight and operates without making costly LLM calls. We demonstrate the effectiveness of our retrieval model on two benchmark datasets by integrating it with an LLM for answer generation. Our results show that *KG-Scout* , when combined with a small-sized LLM, achieves performance that is either comparable to or superior to existing approaches, while also being more efficient.

4.5 Analysis of Reasoning Performance

This section presents a qualitative analysis of the reasoning model's performance by examining specific examples. We will highlight instances where the model successfully extracts correct answers from the provided triplets, as well as cases where it fails to do so despite the presence of the answer entity.

4.5.1 Example of Failed Triplet Interpretation by LLM

Question: Who inspired the person who said, "Her voice is full of money"?

Answer: [*edith wharton, sherwood anderson, r.d.blackmore, sara murphy, shane leslie, john keats, oswald spengler, t.s.eliot*]

Cosine Triplets

1. F. Scott Fitzgerald, people person quotations, Her voice is full of money.
2. Her voice is full of money., common topic notable types, Quotation
3. Her voice is full of money., media common quotation author, F. Scott Fitzgerald
4. Her voice is full of money., media common quotation subjects, Riches
5. Riches, media common quotation subject quotations about this subject, Her voice is full of money.
6. The Great Gatsby, media common quotation source quotations, Her voice is full of money.
7. Her voice is full of money., media common quotation source, The Great Gatsby
8. Her voice is full of money., media common quotation spoken by character, Jay Gatsby
9. Jay Gatsby, fictional universe fictional character quotations, Her voice is full of money.
10. Her voice is full of money., common topic notable for, g.12577x3qf
11. Riches, media common quotation subject quotations about this subject, He must have killed a lot of men to have made so much money.
12. Riches, media common quotation subject quotations about this subject, He who knows he has enough is rich.
13. Riches, media common quotation subject quotations about this subject, Who is rich? He that rejoices in his portion.
14. Riches, media common quotation subject quotations about this subject, A certain kind of rich man afflicted with the symptoms of moral dandyism sooner or later comes to the conclusion that it isn't enough merely to make money. He feels obliged to hold views, to espouse causes and elect Presidents, to explain to a trembling world how and why the world went wrong. The spectacle is nearly always comic.",
15. Riches, media common quotation subject quotations about this subject, Among the very rich you will never find a really generous man, even by accident. They may give their money away, but they will never give themselves away; they are egoistic, secretive, dry as old bones. To be smart enough to get all that money you must be dull enough to want it.
16. Who is rich? He that rejoices in his portion., media common quotation subjects, Riches
17. Who is rich? He that rejoices in his portion., common topic notable types, Quotation
18. Riches, media common quotation subject quotations about this subject, The rich man is always sold to the institution which makes him rich. Absolutely speaking, the more money, the less virtue.

19. Riches, media common quotation subject quotations about this subject, If you want to feel rich, just count all of the things you have that money can't buy.",
20. Riches, media common quotation subject quotations about this subject, For just as poets love their own works, and fathers their own children, in the same way those who have created a fortune value their money, not merely for its uses, like other persons, but because it is their own production. This makes them moreover disagreeable companions, because they will praise nothing but riches.
21. He who knows he has enough is rich., media common quotation subjects, Riches
22. He who knows he has enough is rich., common topic notable types, Quotation
23. Riches, media common quotation subject quotations about this subject, He who is contented is rich.
24. Riches, media common quotation subject quotations about this subject, If Heaven had looked upon riches to be a valuable thing, it would not have given them to such a scoundrel.
25. Even the rich are hungry for love, for being cared for, for being wanted, for having someone to call their own., media common quotation subjects, Riches
26. Riches, media common quotation subject quotations about this subject, For a person to build a rich and rewarding life for himself, there are certain qualities and bits of knowledge that he needs to acquire. There are also things, harmful attitudes, superstitions, and emotions that he needs to chip away. A person needs to chip away everything that doesn't look like the person he or she most wants to become.",
27. The rich man is always sold to the institution which makes him rich. Absolutely speaking, the more money, the less virtue., common topic notable types, Quotation
28. Riches, media common quotation subject quotations about this subject, Every man thinks God is on his side. The rich and powerful know he is.
29. Riches, media common quotation subject quotations about this subject, The rich were dull and they drank too much or they played too much backgammon. They were dull and they were repetitious. He remembered poor Julian and his romantic awe of them and how he had started a story once that began, The very rich are different from you and me. And how someone had said to Julian, Yes, they have more money.
30. Riches, media common quotation subject quotations about this subject, No, not rich. I am a poor man with money, which is not the same thing.

KG-Scout Triplets

1. Her voice is full of money., media common quotation author, F. Scott Fitzgerald
2. The great Gatsby, book book edition author editor, F. Scott Fitzgerald
3. Jay Gatsby, fictional universe fictional character quotations, Her voice is full of money.
4. F. Scott Fitzgerald, fictional universe person in fiction representations in fiction, F. Scott Fitzgerald
5. F. Scott Fitzgerald, fictional universe fictional character creator fictional characters created, Daisy Buchanan
6. F. Scott Fitzgerald, influence influence node influenced, John Cheever
7. What people are ashamed of, usually makes a good story., media common quotation author, F. Scott Fitzgerald
8. F. Scott Fitzgerald, influence influence node influenced, J. D. Salinger
9. Her voice is full of money., media common quotation subjects, Riches
10. The Great Gatsby, media common quotation source quotations, No amount of fire or freshness can challenge what a man will store up in his ghostly heart.
11. Nick Carraway, fictional universe fictional character quotations, Americans, while occasionally willing to be serfs, have always been obstinate about being peasantry.
12. F. Scott Fitzgerald, influence influence node influenced, Richard Yates
13. Her voice is full of money., media common quotation spoken by character, Jay Gatsby
14. Her voice is full of money., media common quotation source, The Great Gatsby
15. F. Scott Fitzgerald, influence influence node influenced, Richard Hofstadter

16. F. Scott Fitzgerald, people deceased person place of death, Hollywood
17. Nick Carraway, fictional universe fictional character quotations, He smiled understandingly—much more than understandingly. It was one of those rare smiles with a quality of eternal reassurance in it, that you may come across four or five times in life. It faced–or seemed to face–the whole external world for an instant, and then concentrated on you with an irresistible prejudice in your favor. It understood you just as far as you wanted to be understood, believed in you as you would like to believe in yourself, and assured you that it had precisely the impression of you that, at your best, you hoped to convey.
18. Daisy Buchanan, fictional universe fictional character character created by, F. Scott Fitzgerald
19. F. Scott Fitzgerald, book book subject works, F. Scott Fitzgerald and the art of social fiction
20. F. Scott Fitzgerald, influence influence node influenced, T. S. Eliot
21. F. Scott Fitzgerald, influence influence node influenced by, Sherwood Anderson
22. Nick Carraway, fictional universe fictional character quotations, It occurred to me that there was no difference between men, in intelligence or race, so profound as the difference between the sick and the well.
23. Riches, media common quotation subject quotations about this subject, Her voice is full of money.
24. F. Scott Fitzgerald, influence influence node influenced, Edna O'Brien",
25. F. Scott Fitzgerald, influence influence node influenced, William Goldman
26. Riches, media common quotation subject quotations about this subject, He who is contented is rich.
27. Everyone suspects himself of at least one of the cardinal virtues, and this is mine: I am one of the few honest people that I have ever known., media common quotation source, The Great Gatsby
28. F. Scott Fitzgerald, influence influence node influenced, Jonathan Lamas
29. The Curious Case of Benjamin Button, film film language, English Language
30. F. Scott Fitzgerald, film writer film, Marie Antoinette'

KG-Scout + Llama Reasoning

prediction: `ans: [F. Scott Fitzgerald]`

Reason:To answer the question, we need to identify the person who said, Ḧer voice is full of moneyänd find out who inspired that person.From the triplets, we can see that the quotation Ḧer voice is full of moneyïs attributed to F. Scott Fitzgerald.Now, we need to find out who inspired F. Scott Fitzgerald.From the triplets, we can see that F. Scott Fitzgerald was influenced by several authors, including Edith Wharton, T. S. Eliot, and John Keats, but the triplet that directly answers the question is not available. However, we can infer that F. Scott Fitzgerald was influenced by various authors, but the specific author who inspired the person who said Ḧer voice is full of moneyïs not explicitly mentioned in the triplets.

Note: In this example we can see that although the triplets provided by*KG-Scout* contains the answer entities and they form a reasoning chain (triplets highlighted), still Llama model failed to correctly interpret the answer

4.5.2 Example of Successful Triplet Interpretation by LLM

Question: People from the country that contains Corfu speak what language?

Answer: [*albanian language, greek language*]

Cosine Triplets

1. Corfu, location location people born here, Hyacinthe Klosé
2. Corfu, location location people born here, Tasos Pappas
3. Corfu, location location people born here, g.1pz2txwwb
4. Corfu Island, location location people born here, Sotiris Trivizas
5. Corfu Island, location location people born here, Antzela Gerekou
6. Corfu Island, location location people born here, Sofoklis Dousmanis
7. Corfu, location location contains, Corfu
8. Corfu Island, location location people born here, Xenophon Stratigos
9. Corfu, location location people born here, Grigoris Chandrinos
10. Corfu, location location people born here, Spyridon Xyndas
11. Corfu Island, location location people born here, Viaros Kapodistrias
12. Corfu Island, location location people born here, Eugenios Voulgaris
13. Albanian language, language human language countries spoken in, Greece
14. Corfu, location location people born here, Spyridon Samaras
15. Corfu Island, location location people born here, Sakis Rouvas
16. Corfu Island, location location people born here, Viktor Dousmanis
17. Greek Language, language human language countries spoken in, Albania
18. Corfu, location location people born here, Nikos Kritikos
19. Corfu Island, location location people born here, Valeria Christodoulidou
20. Corfu Island, location location people born here, Andonios Liveralis
21. Corfu, location location people born here, g.1q5ccsg_x
22. Greek Language, language human language countries spoken in, Greece
23. Corfu Island, location location people born here, Thodoros Exarhos
24. Corfu, location location people born here, Evangelia Randou
25. Corfu Island, location location people born here, g.1yprty2gl
26. Corfu Island, location location people born here, Stamatis Voulgaris
27. Corfu Island, location location people born here, Vasso Geka
28. Corfu Island, location location people born here, Mihalis Desilas
29. Corfu Island, location location people born here, Alexandros Mouzakitis
30. Corfu Island, location location people born here, Safiye Sultan

KG-Scout Triplets

1. Corfu, location location people born here, Nikos Kritikos
2. Corfu, location location contains, Ionian University
3. Corfu, location location containedby, Corfu
4. Corfu, location location contains, Banknote Museum
5. Corfu, location location contains, Corfu
6. Corfu, location location contains, Archaeological Museum of Corfu
7. Greece, location country official language, Greek Language
8. Greece, location country languages spoken, Albanian language
9. Corfu, location location contains, Corfu Island
10. Corfu, location location containedby, Corfu Island
11. Corfu, location location people born here, Robert Reid, 1st Earl Loreburn
12. Corfu, location location people born here, Spyridon Samaras

13. Corfu, travel travel destination tourist attractions, Byzantine Museum of Antivouniotissa
14. Corfu, location location people born here, Louisa Pouli
15. Corfu, location location people born here, Hyacinthe Klosé
16. Corfu, location location contains, Museum of Asian art of Corfu
17. Greek Language, language human language countries spoken in, Greece
18. Corfu, travel travel destination tourist attractions, Archaeological Museum of Corfu
19. Corfu, location location containedby, Greece
20. Corfu, location location people born here, Fanny Marchiò
21. Corfu, location location containedby, Greece
22. Corfu, location location containedby, Ionian Islands
23. Corfu, travel travel destination tourist attractions, St. Spyridon Church
24. Corfu, travel travel destination tourist attractions, Temple of Artemis, Corfu
25. Corfu, sports sports team location teams, PAE Kerkyra
26. Albanian language, language human language countries spoken in, Greece
27. Corfu, location administrative division country, Greece
28. Corfu, location location people born here, Hector Kaloudis
29. Corfu, location location contains, Ionian University
30. Greece, location country languages spoken, Greek Language

KG-Scout + Llama Reasoning

prediction: `ans: [Greek Language]`

Reason:To answer the question, we need to: Identify the country that contains Corfu. Find the language spoken in that country. From the triplets, we can see that Corfu is contained by Greece. Now we need to find the language spoken in Greece.From the triplets, we can see that the official language of Greece is Greek Language.Therefore, people from the country that contains Corfu (Greece) speak Greek Language.

REFERENCES

1. **Augenstein, I.**, **T. Baldwin**, **M. Cha**, **T. Chakraborty**, **G. L. Ciampaglia**, **D. Corney**, **R. DiResta**, **E. Ferrara**, **S. Hale**, **A. Halevy**, *et al.* (2024). Factuality challenges in the era of large language models and opportunities for fact-checking. *Nature Machine Intelligence*, **6**(8), 852–863.

2. **Bollacker, K.**, **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.

3. **Borgeaud, S.**, **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.

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

5. **Dehghan, M.**, **M. A. Alomrani**, **S. Bagga**, **D. Alfonso-Hermelo**, **K. Bibi**, **A. Ghaddar**, **Y. Zhang**, **X. Li**, **J. Hao**, **Q. Liu**, *et al.* (2024). Ewek-qa: Enhanced web and efficient knowledge graph retrieval for citation-based question answering systems. *arXiv preprint arXiv:2406.10393*.

6. **Dhingra, B.**, **J. R. Cole**, **J. M. Eisenschlos**, **D. Gillick**, **J. Eisenstein**, and **W. W. Cohen** (2022). Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, **10**, 257–273.

7. **Grattafiori, A.**, **A. Dubey**, **A. Jauhri**, **A. Pandey**, **A. Kadian**, **A. Al-Dahle**, **A. Letman**, **A. Mathur**, *et al.* (2024). The llama 3 herd of models. URL `https://arxiv.org/abs/2407.21783`.

8. **Haveliwala, T. H.**, Topic-sensitive pagerank. *In Proceedings of the 11th international conference on World Wide Web*. 2002.

9. **He, X.**, **Y. Tian**, **Y. Sun**, **N. Chawla**, **T. Laurent**, **Y. LeCun**, **X. Bresson**, and **B. Hooi** (2024). G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *Advances in Neural Information Processing Systems*, **37**, 132876–132907.

10. **Jiang, J.**, **K. Zhou**, **Z. Dong**, **K. Ye**, **X. Zhao**, and **J.-R. Wen**, StructGPT: A general framework for large language model to reason over structured data. *In* **H. Bouamor**, **J. Pino**, and **K. Bali** (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Singapore, 2023. URL https://aclanthology.org/2023.emnlp-main.574/.

11. **Karpukhin, V.**, **B. Oguz**, **S. Min**, **P. S. Lewis**, **L. Wu**, **S. Edunov**, **D. Chen**, and **W.-t. Yih**, Dense passage retrieval for open-domain question answering. *In EMNLP (1)*. 2020.

12. **Kojima, T.**, **S. S. Gu**, **M. Reid**, **Y. Matsuo**, and **Y. Iwasawa** (2022). Large language models are zero-shot reasoners. *Advances in neural information processing systems*, **35**, 22199–22213.

13. **Lewis, P.**, **E. Perez**, **A. Piktus**, **F. Petroni**, **V. Karpukhin**, **N. Goyal**, **H. Küttler**, **M. Lewis**, **W.-t. Yih**, **T. Rocktäschel**, *et al.* (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, **33**, 9459–9474.

14. **Reimers, N.** and **I. Gurevych**, Sentence-bert: Sentence embeddings using siamese bert-networks. *In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2019. URL https://arxiv.org/abs/1908.10084.

15. **Robertson, S.**, **H. Zaragoza**, *et al.* (2009). The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, **3**(4), 333–389.

16. **Shuster, K.**, **S. Poff**, **M. Chen**, **D. Kiela**, and **J. Weston** (2021). Retrieval augmentation reduces hallucination in conversation. *arXiv preprint arXiv:2104.07567*.

17. **Sun, J.**, **C. Xu**, **L. Tang**, **S. Wang**, **C. Lin**, **Y. Gong**, **L. M. Ni**, **H.-Y. Shum**, and **J. Guo** (2023). Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. *arXiv preprint arXiv:2307.07697*.

18. **Talmor, A.** and **J. Berant** (2018). The web as a knowledge-base for answering complex questions. *arXiv preprint arXiv:1803.06643*.

19. **Wang, K.**, **F. Duan**, **S. Wang**, **P. Li**, **Y. Xian**, **C. Yin**, **W. Rong**, and **Z. Xiong** (2023). Knowledge-driven cot: Exploring faithful reasoning in llms for knowledge-intensive question answering. *arXiv preprint arXiv:2308.13259*.

20. **Williams, R. J.** (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, **8**(3), 229–256.

21. **Yang, R.**, **H. Liu**, **E. Marrese-Taylor**, **Q. Zeng**, **Y. Ke**, **W. Li**, **L. Cheng**, **Q. Chen**, **J. Caverlee**, **Y. Matsuo**, *et al.* (). Kg-rank: Enhancing large language models for medical qa with knowledge graphs and ranking techniques. arxiv 2024. *arXiv preprint arXiv:2403.05881*.

22. **Yih, W.-t.**, **M. Richardson**, **C. Meek**, **M.-W. Chang**, and **J. Suh**, The value of semantic parse labeling for knowledge base question answering. *In* **K. Erk** and **N. A. Smith** (eds.), *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Berlin, Germany, 2016. URL `https://aclanthology.org/P16-2033/`.

23. **Yu, D.**, **S. Zhang**, **P. Ng**, **H. Zhu**, **A. H. Li**, **J. Wang**, **Y. Hu**, **W. Wang**, **Z. Wang**, and **B. Xiang** (2022). Decaf: Joint decoding of answers and logical forms for question answering over knowledge bases. *arXiv preprint arXiv:2210.00063*.