HOLMES: Hyper-Relational Knowledge Graphs for Multi-hop Question Answering using LLMs

Pranoy Panda [*1]   **Ankush Agarwal**[1]   **Chaitanya Devaguptapu**[1]
Manohar Kaul[1]   **Prathosh A P**[1, 2]

[1]Fujitsu Research India
[2]Indian Institute of Science, Bengaluru
{pranoy.panda, ankush.agarwal, manohar.kaul}@fujitsu.com
email@chaitanya.one, prathoshap@gmail.com

Abstract

Given unstructured text, Large Language Models (LLMs) are adept at answering simple (single-hop) questions. However, as the complexity of the questions increase, the performance of LLMs degrade. We believe this is due to the overhead associated with understanding the complex question followed by filtering and aggregating unstructured information in the raw text. Recent methods try to reduce this burden by integrating structured knowledge triples into the raw text, aiming to provide a structured overview that simplifies information processing. However, this simplistic approach is query-agnostic and the extracted facts are ambiguous as they lack context. To address these drawbacks and to enable LLMs to answer complex (multi-hop) questions with ease, we propose to use a knowledge graph (KG) that is *context-aware* and is distilled to contain query-relevant information. The use of our compressed distilled KG as input to the LLM results in our method utilizing up to 67% fewer tokens to represent the query relevant information present in the supporting documents, compared to the state-of-the-art (SoTA) method. Our experiments show consistent improvements over the SoTA across several metrics (EM, F1, BERTScore, and Human Eval) on two popular benchmark datasets (HotpotQA and MuSiQue).

1 Introduction

Multi-Hop Question Answering (MHQA) is a field that presents unique challenges in the realm of natural language processing. To illustrate the challenges of MHQA, consider extracting information from data arising during board meetings. While current technologies (such as LLMs) are proficient at addressing simple (single-hop) questions, such as "*How many board meetings were held in the last twelve months?*", they falter when confronted with
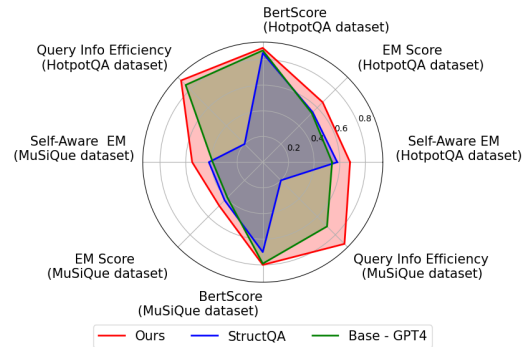

Figure 1: **Multi-Dimensional Improvements**: Our method (with GPT-4 as reader LLM) achieves SoTA results on several datasets and multiple Multi-hop QA metrics. *EM*: Exact-Match with the gold answer, *Self-Aware EM*: Confidence-aware EM, *BertScore* (Zhang et al., 2019): Semantic similarity between predicted and gold answer; *Query Info Efficiency*: Efficiency of representing query-relevant information in the supporting documents - inversely proportional to the input token count for the reader LLM.

complex (multi-hop) questions. An example of a multi-hop question is "*For the board meeting with the most divided votes in the last twelve months, what was the agenda, who voted against it, and by what margin did it pass or fail?*". Answering this question requires a series of interconnected steps: first, identifying the meeting with the most divided votes; next, determining the main agenda of that meeting; then, listing the members who voted against it; and finally, calculating the margin by which it was approved or rejected. Each of these steps, or "hops", demands not only the retrieval of additional information but also a nuanced understanding of the context and the relationships between various entities. This complexity underscores the challenge of MHQA, where the goal is to navigate through layers of information to arrive at the answer. Our work is aimed at tackling these multifaceted questions.

LLMs have emerged as a powerful set of tools for various NLP tasks (Zhang et al., 2023; Wad-

---

*Corresponding author : pranoy.panda@fujitsu.com

Input

Question: Who was the main vocal of "Set the Fire to the Third Bar"?

Supporting Documents

Title-1: Set the Fire to the Third Bar
Passage-1: "**Set the Fire to the Third Bar**" is a song from **Snow Patrol's** fourth album, "**Eyes Open**", featuring **Martha Wainwright** on vocals.

Title-2: Snow Patrol
Passage-2: **Snow Patrol** are a Northern Irish-Scottish rock band formed in 1993, consisting of **Gary Lightbody** (vocals, guitar), Nathan Connolly (guitar, backing vocals), ...

Title-3: Rita Redshoes
Passage-3: ... his second solo album. In 2008 she released her debut solo album "Golden Era". In 2010 the band **Snow Patrol** invited her to perform single "**Set the fire to the third bar**" in Rock in Rio Lisbon.

Title-4: Laura Jean
Passage-4: ... support from community and alternative radio stations such as Triple J. She received high exposure when **Snow Patrol**, fans of her, asked her to sing **Martha Wainwright's** part in the song **Set the Fire to the Third Bar** during their two Australian tours in 2007. ...

GPT-4 → **Predicted Answer:** Martha Wainwright ✗

Struct-QA + GPT-4 (Li and Du, 2023) → **Predicted Answer:** Martha Wainwright ✗

Struct-QA's CoT-Based Output
To determine the main vocal of "Set the Fire to the Third Bar", let's go through the docs step-by-step:
1. The document on "**Set the Fire to the Third Bar**" confirms **Martha Wainwright** as the featured
2. Cross-referencing details on "**Set the Fire to the Third Bar**" the "**Snow Patrol**" document lacks information on the main vocal. "Laura Jean" mentions Laura Jean covering **Martha Wainwright's** part during tours ...
3. The main vocal for "**Set the Fire to the Third Bar**" is **Martha Wainwright**, featured in the original song along with **Gary Lightbody**.

HOLMES + GPT-4 (Our Method) → **Predicted Answer:** Gary Lightbody ✓

Our Method's Output
The question asks for the main vocal of the song "**Set the Fire to the Third Bar**"

From the facts provided, "**Set the Fire to the Third Bar**" by **Snow Patrol** features **Gary Lightbody** as the lead singer. However, **Martha Wainwright** is mentioned as providing vocals, suggesting her involvement. Considering the lead singer's role, **Gary Lightbody** is the main vocal.

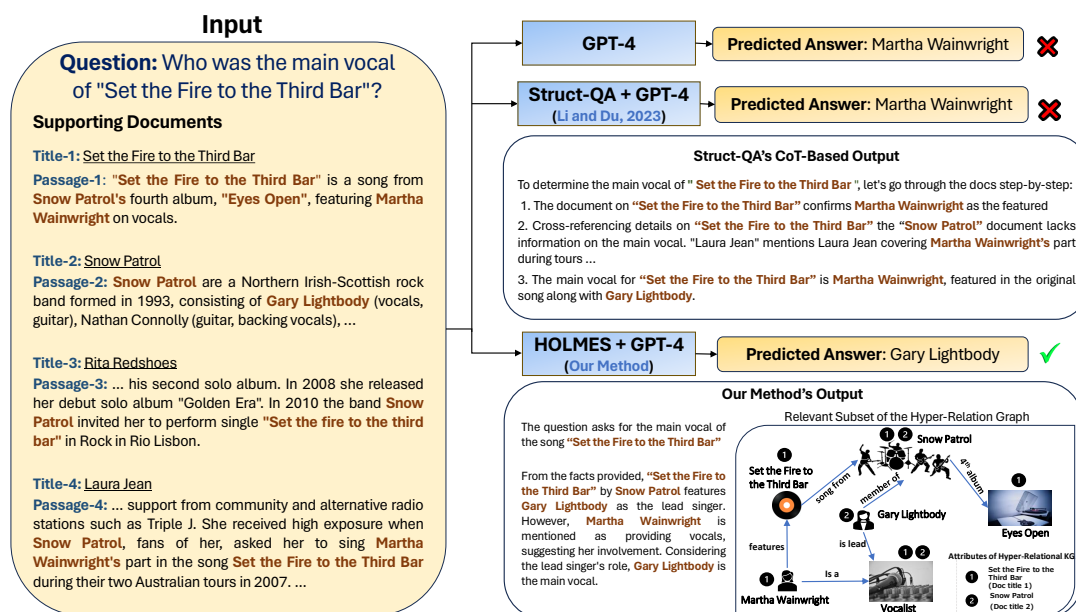Relevant Subset of the Hyper-Relation Graph

Figure 2: **Multi-Hop QA Case Study**: This figure illustrates a bridge-type multi-hop question from HotpotQA dataset for comparing our method with the baselines. It highlights our method's accurate identification of '*Snow Patrol*' as the crucial bridge entity and subsequently finding the lead vocalist, a feat not achieved by baselines.

hwa et al., 2023; Robinson et al., 2022; Xu et al., 2023). However, despite their prowess, LLMs oftentimes are found lacking, when confronted with complex, multi-hop questions (see Figure 1 for an example). We hypothesize that the degradation in performance is due to the complexity involved in *filtering and contextual aggregation of information present in the unstructured text*.

To address this challenge with unstructured text, recent methods, such as StructQA (Li and Du, 2023), extract structured knowledge in the form of KG triples from raw text and couple it with Chain-of-Thought (CoT) reasoning (Wei et al., 2022). However, the extracted KG triples are not dependent on the query and lack the context under which these facts are valid. To understand the ambiguity due to the lack of context, consider this KG triple: {*subject: "Apple", relation: "prices rose", object: "10%"*}. Without additional context, it is difficult to determine whether the entity "*Apple*" refers to the fruit or the company. Moreover, StructQA provides both the extracted KG triples and the raw text as input to the LLM, leading to significantly longer prompts (see Table 3) and information redundancy. Our method, HOLMES[1], addresses these challenges by creating a query-focused context-aware KG and using it as the sole input for the LLM, *i.e.*, without inputting the raw text. Specifically, we

(i) synthesize a hyper-relational KG from unstructured text that captures both facts, and the context under which these facts are valid, and (ii) prune the hyper-relational KG using a knowledge schema that encodes the type of information necessary to answer the query. These steps collectively furnish the LLM with a curated set of relevant facts. As an example, we provide a case study on the HotpotQA dataset in Figure 2.

To rigorously evaluate our approach, we use two challenging multi-hop QA datasets, HotpotQA (Yang et al., 2018) and MuSiQue (Trivedi et al., 2022b), and experiment with three SoTA reader LLMs for QA: GPT-3.5, GPT-4, and Gemini-Pro. We compare with the baselines across 6 metrics: EM, F1, Precision, Recall, Human-Eval, BERTScore, and achieve significant gains.

Our contributions are as follows:

- A new multi-hop QA approach that transforms unstructured text into a hyper-relational KG using a query-derived schema, serving as an input to the LLM.
- A significant improvement over the SoTA multi-hop QA method (Li and Du, 2023): 18.7% and 20% in EM scores on the Hotpot dataset, and 26% and 14.3% on the MuSiQue dataset for GPT-3.5 and GPT-4, respectively (see Table 1).
- Using our query-focused hyper-relational KG we use up to 67% fewer tokens to represent

---

[1] Named after Sherlock Holmes for its ability to deduce query relevant information from unstructured text

query relevant information than the current SoTA method, and up to 60% fewer tokens w.r.t. original supporting documents (see Table 3).

2 Related Work

Multi-hop Question Answering (MHQA) Recent achievements in addressing straightforward questions (Lan et al., 2019) coupled with availability of high-quality MHQA datasets (Yang et al., 2018; Trivedi et al., 2022b) have prompted a shift towards tackling multi-hop questions.

Existing works in MHQA have adopted various approaches, including: (i) Construction of dynamic graphs (Qiu et al., 2019), hierarchical (Fang et al., 2020) and Abstract meaning representation (AMR) based graphs (Deng et al., 2022) as well as leveraging KGs (Li and Du, 2023) (ii) Employing end-to-end differentiable learning methods, utilizing Multi-task Transformers for retrieval, reranking and predicting in an iterative manner (Qi et al., 2021) or using Recurrent Neural Networks (RNN) to sequentially retrieve documents from a graph of entities (Asai et al., 2020) and (iii) dynamically converting multi-hop questions into single-hop questions by generating subsequent questions based on the answers to previous ones (Perez et al., 2020), or by updating single-hop questions in the embedding space (Sun et al., 2021). In contrast to these methodologies, our work introduces a novel approach by constructing a hyper-relational KG from the documents, which is then utilized exclusively to answer questions.

Relational Graphs for MHQA While retrieving and relying on purely unstructured text is a go to approach for single-hop question answering (Watanabe et al., 2017). It may not be ideal for MHQA, as it is designed to make multi-step, comprehensive reasoning difficult. This issue is often addressed by constructing structured sources of information from raw text and in most of the cases using KGs (Li and Du, 2023). By using KGs, the relational information among question concepts and answers can be easily captured (Dong et al., 2023). However, one of the main drawback of KGs is the lack of context *i.e.*, they focus only on triples, overlooking qualifiers important for inference. To avoid overlooking qualifiers, our method involves building hyper-relational KGs to serve as an input to LLMs for MHQA, as explained in detail in Sect 4.2.1. To the best of our knowledge, our work is the first to use hyper-relational KGs for MHQA.

Training Free MHQA Previously, KGs have been applied to MHQA in two ways: training-based (Sun et al., 2018, 2019; Yavuz et al., 2022; Ramesh et al., 2023) and training-free (Li and Du, 2023), also known as prompting. In these methods, KGs used were either human-curated (Speer et al., 2017; Bollacker et al., 2008) or training-based (Izacard and Grave, 2021; Bosselut et al., 2019). In some cases, LLMs were employed for triple extraction from documents to form graphs (Li and Du, 2023; Carta et al., 2023). However, no prior works automatically create a schema for the graph and use the graph for MHQA, a unique approach used by us.

3 Preliminaries

Hyper-Relational Knowledge Graph - A hyper-relational KG $\mathcal{H}$ is an enriched form of a traditional KG. It allows for the representation of multiple relationships between entities.

Let $\mathcal{A}$ denote a collection of *attribute sets* whose elements represent different types of attribute sets. For example, $A_{ts} \in \mathcal{A}$ is a set of *timestamps*, while $A_{dt} \in \mathcal{A}$ is a set of *document titles*, that serve as additional attributes. We then define our hyper relational KG $\mathcal{H}$ as: $\mathcal{H} = \{(e_s, r, e_o, a) \mid e_s, e_o \in E, r \in R, a \in A_{dt}\}$, where E is the set of named entities, R is the set of relations and A_{dt} is the set of document titles that are additional attributes. This structure not only links entities e_s and e_o via relation r but also integrates the corresponding document title attributes A_{dt}, offering a more nuanced description. We refer (e_s, r, e_o, a) as a hyper triple.

Graph Schema - A graph schema $\mathcal{G}_S$, defined as $\mathcal{G}_S = \{(\tilde{e}_s, r, \tilde{e}_o) \mid \tilde{e}_s, \tilde{e}_o \in E_T, r \in R\}$, outlines the structure of a KG through entity types E_T and relations R. Each triple in the schema specifies the permissible entity types $\tilde{e}_s$ (subject) and $\tilde{e}_o$ (object) for each relation r, serving as a blueprint for organizing knowledge.

Named Entity and Relation Extraction - Named entity extraction identifies textual spans referring to specific entities, while relation extraction classifies semantic relationships between entities in text, enabling structured representation of unstructured data. In this work, we harness LLMs' capabilities for these tasks. Pre-trained on extensive text, these models accurately classify named entities and extract relations with minimal examples (Li and Du, 2023; Wang et al., 2023; Wadhwa et al., 2023), showcasing their proficiency in contextual understanding. See Appendices A.4.1, A.4.2 for detailed prompts and few-shot examples used in our method
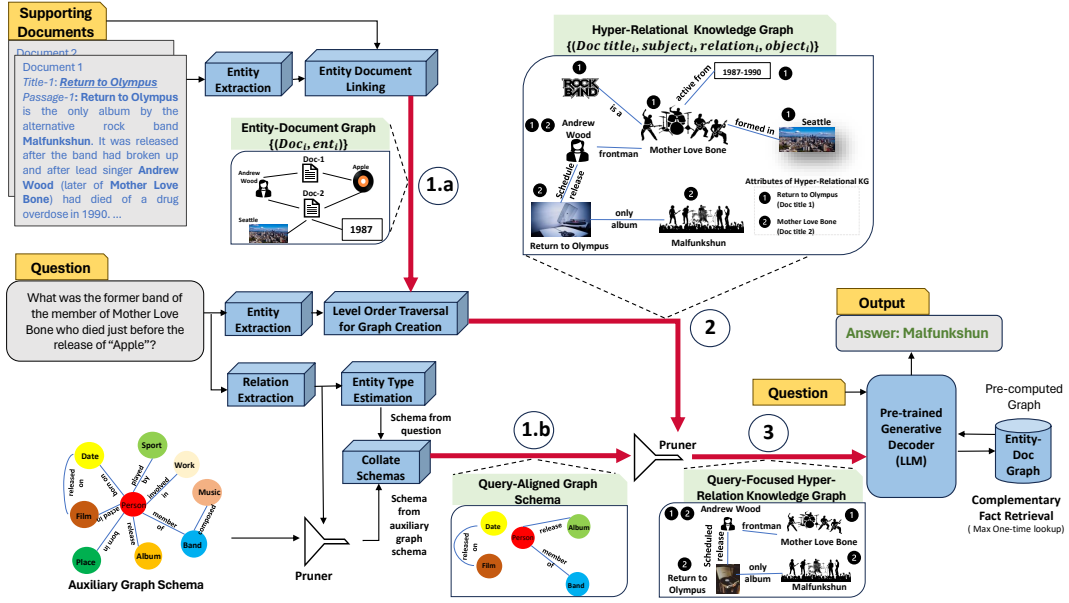
Figure 3: **Method Overview**: Our method has three key steps - (i) Query-Dependent Structured Knowledge Discovery (Section 4.2.1), (ii) Knowledge Schema Construction for Information Refinement (Section 4.2.2), and (iii) Reader LLM Prompt Construction (Section 4.2.3). Step (i) involves creation of an entity document graph (1.a in the Figure), and performing a level-order traversal on it to get a Hyper-relational KG (2 in the Figure). Next, in step (ii), we create a query-aligned knowledge schema from the question and an auxiliary graph schema (1.b in the Figure), and use it to prune the Hyper-Relational KG (3 in the Figure) - which forms the input for the LLM.

for named entity and relation extraction.

4 Methodology

The key idea of our method is to identify the subset of documents that contain the answer to the multi-hop query, and subsequently extract context-aware structured information from them (where context comes from the documents). We further perform a refinement step to retain query-relevant information. Below, we begin with a brief description of the problem statement in Section 4.1, followed by an explanation of our method in Section 4.2. The overview of our approach can be seen in Figure 3.

4.1 Problem Statement

In this work, we address the challenge of MHQA in a zero-shot setting. This implies that we assume no prior domain-specific labeled data, rendering the problem both challenging and practically useful (Davison, 2020). Moreover, we operate in a training-free setting, leveraging the reasoning capabilities of LLMs for this task.

Formally, the task is to extract the answer a_q from a given natural language question q and a collection of supporting documents $\mathcal{S}_q$. The set of supporting documents is defined as $\mathcal{S}_q := \{D_q^1, D_q^2, \ldots, D_q^m\}$, where each document $D_q^j := (t_q^j, p_q^j)$ consists of a title t_q^j and the associated passage p_q^j. Given

that q is a multi-hop question, deducing its answer requires the aggregation of information from at least two documents $D_q^i, D_q^j \in \mathcal{S}_q$ (where $i \neq j$).

4.2 Proposed Method (HOLMES)

We begin by traversing the supporting documents to identify the subset of documents relevant to the query, and extract structured, context-aware information (Section 4.2.1). This information is then refined using a query-based knowledge schema (Section 4.2.2). Next, we format the distilled graph for the reader LLM prompt, including a complementary fact retrieval step within the prompt to ensure higher coverage of query-relevant information. (Section 4.2.3).

4.2.1 Query-Dependent Structured Knowledge Discovery

Unstructured text, contains a complex web of facts and relationships forming a latent semantic graph. This graph represents interconnected factual information not explicitly structured but implied by relationships and named entities in the text. Our approach navigates this structure using the entity-document graph—a bipartite graph linking documents and named entities. It helps uncover parts of the latent graph relevant to the given query.

Entity-Document Graph Construction The entity-document graph has two node

types—documents and entities—with a single edge connecting them. We begin by extracting named entities from supporting documents. Then, we establish edges between document and entity nodes, forming a bipartite graph that captures the connections between entities and the documents they appear in (See Figure 3 - (1.a) for an example).

Level Order Traversal and Structured Information Extraction We begin by extracting named entities from the query and use them for a level-order/breadth-first traversal of the entity-document graph. Similar to Li and Du (2023); Wadhwa et al. (2023), we use LLMs to extract KG triples from document nodes, framing the task as a sequence-to-sequence problem. By providing the LLM with a detailed prompt (a clear description of the task and output format) and few-shot examples, we guide the model to generate triples directly from raw text (see Appendix A.4.4 for prompts).

After extracting triples, we enhance them into hyper-relational KG quadruples (or hyper triples) by appending the title of the source document to each triple. This title serves as an additional attribute, offering the LLM context to understand when the triple is valid. During the traversal, we filter hyper triples with the current entity node as either the subject or object, adding the counterpart entity to the traversal queue. This iterative process continues for a predefined number of hops k, where the value of k defines the query complexity, *i.e.*, the number of facts required to answer the question. Example of a hyper-relational KG in Figure 3-(2). When we perform the above specified k-hop traversal starting from the named entities in the query, we are effectively probing the latent semantic graph in the raw text by incrementally selecting subsets of documents related to the query.

4.2.2 Knowledge Schema Construction for Information Refinement

The hyper-relational graph construction is rooted in the inference query. However, it contains some hyper triples which although are related to the named entities in the query, capture relationships that are not useful for answering it. For instance, a query might focus on a person's occupation, but the graph also captures their hobbies. To eliminate such extraneous information, we construct a query-aligned knowledge schema (Figure 3 - (1.b) shows an example) and then carry out a refinement step (refined output example in Figure 3 - (3)).

Query-Aligned Graph Schema Creation The graph schema acts as a structural template, guiding the organization of information crucial for answering the given question. It is essential for capturing the types of entities and relations likely to form the backbone of the answer. For example, in response to the question *"Who is the 2nd daughter of the 1st president of X?"* the schema should capture relations such as {<Person>, daughter of, <Person>} and {<Person>, president of, <Country>}, representing the direct information sought by question. Our graph schema is populated using two sources:

(i) We first derive schema elements from the inference query by identifying relations in it, and then using LLMs to estimate the subject and object entity types for each of these relations (prompt in Appendix A.4.3). This forms a schema with a direct alignment with the question's intent.

(ii) We enrich the schema with additional domain-specific relations to aid multi-hop reasoning, using an auxiliary graph schema derived from in-domain questions (Appendix A.5). This includes a wider array of entity-relation pairs, such as {<Person>, child of, <Person>} and {<Person>, head of, <Country>}, to account for potential inferential steps when direct query matches are absent in supporting documents. We select schema triples with relations similar to the question's (based on cosine similarity) to retrieve a relevant subset. While constructing this auxiliary schema incurs a one-time compute cost, its amortized cost is minimal over multiple queries, especially given it helps to reduce the input token length for the reader LLM by upto 60% w.r.t. original source documents.

In cases where the inference query diverges from the auxiliary graph schema, we rely on the schema derived from the query. This ensures system flexibility, adapting to specific query requirements, while still benefiting from the broader knowledge in the auxiliary schema when applicable.

Pruning Hyper-relational Knowledge Graph The final step in our methodology involves refining the constructed hyper-relational KG '$\mathcal{H}$' by pruning it using the query-aligned graph schema $\mathcal{G}_S$. This pruning process is essential to distill the graph to its most relevant components, thereby enhancing the efficiency and effectiveness of the reader LLM in generating answers.

The pruning process begins by computing embeddings for the relations in both the hyper-relational KG and the graph schema. Let $\mathbf{v}_r$ denote the embedding of a relation r in the hyper-relational KG,

and let $\mathbf{u}_{r'}$ denote the embedding of a relation r' in the graph schema. We compute cosine similarity between each pair of relation embeddings:

$$\text{sim}(\mathbf{v}_r, \mathbf{u}_{r'}) = \frac{\mathbf{v}_r \cdot \mathbf{u}_{r'}}{\|\mathbf{v}_r\| \|\mathbf{u}_{r'}\|}, \qquad (1)$$

where $\cdot$ denotes the dot product and $\| \cdot \|$ denotes the Euclidean norm. For each relation r_h in a hyper triple $(e_s, r_h, e_o, a) \in \mathcal{H}$, we compute its highest similarity score w.r.t. any relation r_s in the schema:

$$\text{score}(r_h) = \max_{r_s \in \mathcal{G}_S} \text{sim}(\mathbf{v}_{r_h}, \mathbf{u}_{r_s}). \qquad (2)$$

We compute the scores for all those relations whose entity types in the hyper triple match with that in the schema. We then select p hyper triples with the highest scores. This gives us the pruned graph $\mathcal{H}' = \text{sort}_{\text{score}}(\mathcal{H}, p)$, where $\text{sort}_{\text{score}}$ denotes the sorting operation based on the computed similarity scores, and p - no. of hyper triples to retain.

4.2.3 Reader LLM Prompt Construction

Next, we create the prompt for the reader LLM. Each hyper triple in the pruned hyper-relational KG, $\mathcal{H}'$, is verbalized into an English sentence as the LLMs are adept at understanding the same (Jiang et al., 2023). The verbalization process transforms the structured triple into a natural language text. To be specific, we concatenate each item in the hyper triple into a long sentence marked by specific separation and boundary symbols. The resulting sentences are then arranged in the descending order of their similarity scores w.r.t. the schema to form the input prompt. We arrange the facts based on the relevance to the query (measured via similarity scores) as the information retrieval by LLMs (w.r.t. their input prompt) is done best when the gold information is placed closest to query (Cuconasu et al., 2024).

As structured information extraction is an unsolved problem, some pertinent details may be missed in the input graph. To mitigate this, we include a verification step in the prompt, described below.

Complementary Fact Retrieval If the LLM identifies that facts about a particular set of named entities is missing from the input graph, then we instruct it to list those named entities. We then fetch the corresponding documents from the entity-document graph and integrate them with the initial set of relevant facts. This process not only enriches the input for the LLM but also ensures that any missing query-relevant information is retrieved, enhancing the accuracy of the system's responses. We

assume that a single-step verification is sufficient. Refer Appendix A.4.5 for the reader LLM prompt.

5 Experimental Setup

5.1 Evaluation Details

Datasets: We use two benchmark multi-hop question-answering datasets, namely HotpotQA (Yang et al., 2018) and MuSiQue (Trivedi et al., 2022b) for evaluating our method. Table 9 displays the total number of samples for both datasets in the training and development sets.

For our evaluation process, we utilize questions, context, and gold answers from the development sets of both HotpotQA and MuSiQue. To tune hyperparameters, we randomly select 50 questions from the development set and use them (following (Trivedi et al., 2022a; Li and Du, 2023)). We create our test set by randomly sampling an additional 1000 questions from the development set of HotpotQA and 1200 questions from development set of MuSiQue. Its worth noting that our test set size is twice as big as StructQA (Li and Du, 2023). In both HotpotQA and MuSiQue datasets, for each question there are ten and twenty supporting documents available, respectively. Each document in both datasets contains a title and a passage of text.

Baselines: We operate in a training-free setting, utilizing LLMs as the *reader* model (takes the question and corresponding supporting documents as input and gives an answer). We experiment with two popular LLMs - GPT-3.5, GPT-4 (Achiam et al., 2023) in the main paper, and report results with Gemini (Lee et al., 2023) in Appendix A.1. We compare our method against three baselines
(i) *StructQA* (Li and Du, 2023): CoT (Wei et al., 2022) based SoTA method for MHQA with LLMs. We use the same prompts and methodology as outlined in their study.
(ii) *Base (with supp docs)*: To test the base multi-hop reasoning of the reader LLM, in this baseline, we feed the reader LLM with the question and supporting documents directly.
(iii) *Base (w/o supp docs)*: To test the parametric knowledge of the reader LLM, in this baseline, we feed it with just the question and elicit a response. All prompts used in our experiments are documented in the Appendix A.4 for reproducibility.

Evaluation Metrics: We use Exact-Match (EM), Precision, Recall, and F1-Score as automatic metrics (Li and Du, 2023; Mavi et al., 2022) to measure correctness of the predicted answers.

Datasets	HotpotQA				MuSiQue			
Methods	EM (↑)	F1 (↑)	P (↑)	R (↑)	EM (↑)	F1 (↑)	P (↑)	R (↑)
Reader: gpt-4-1106-preview								
Base (w/o supp docs)	0.26	0.45	0.45	0.50	0.09	0.21	0.22	0.21
Base (with supp docs)	0.54	0.74	0.75	0.77	0.39	0.55	0.55	0.56
StructQA (Li and Du, 2023)	0.55	0.77	0.75	**0.80**	0.42	0.56	0.57	0.56
Our Method	**0.66**	**0.78**	**0.80**	0.79	**0.48**	**0.58**	**0.59**	**0.59**
Reader: gpt-3.5-turbo-1106								
Base (w/o supp docs)	0.23	0.37	0.38	0.40	0.06	0.15	0.17	0.15
Base (with supp docs)	0.47	0.65	0.66	0.68	0.24	0.36	0.36	**0.37**
StructQA (Li and Du, 2023)	0.48	0.64	0.62	0.67	0.23	0.37	0.37	**0.37**
Our Method	**0.57**	**0.69**	**0.69**	**0.70**	**0.29**	**0.38**	**0.39**	0.37

Table 1: **Multi-hop Reasoning Evaluation (Automatic Metrics)**: We report the Exact-Match (EM), F1, Precision (P) and Recall (R) scores of all methods in comparison on two MHQA datasets. We experiment with two SoTA reader LLMs for the QA task - GPT-4 and GPT-3.5. We report results on Gemini-pro in the Appendix A.1. The results indicate consistent and significant improvements across datasets, metrics and LLMs. *Base*: Only reader LLM; *supp docs*: supporting documents w.r.t. query

For semantic evaluation of the predicted answers, we also compute Human Evaluation Score (H-Eval) and BERTScore (Zhang et al., 2019) on a random sample of 100 questions from the development set of the HotpotQA dataset. The Human Evaluation Scores are obtained by averaging scores from three annotators. We use a smaller subset of question for this analysis due to the resource-intensive nature of human evaluation.

5.2 Implementation Details

Problem Setting We focus on the *distractor* setting (Yang et al., 2018; Trivedi et al., 2022b) for question answering. In this setting, the supporting documents for each question consist of a set of distractor documents, *i.e.*, documents not useful for predicting the answer. This setting poses a challenge, demanding robustness to noise in input.

Knowledge Triple Extraction Both StructQA and our method use LLMs for knowledge triple extraction. Thus, for a fair comparison, we always use the same triple extractor LLM for both. Results in Table 1, use `gpt-3.5-turbo-instruct` for triple extraction. We further study the impact of different triple extractor LLM in Table 6.

HOLMES hyperparameters: We use OpenAI embedding model (`text-embedding-ada-002`) to compute text embeddings. We set the value of k (number of levels in the level order traversal) to 4 across both the datasets in Table 1,6. Beyond 4 levels of traversal, performance remains the same as the complexity of the dataset vary from 2-4 hop (See Table 10). Similarly, we set the value of p (number of hyper triples to be retained after pruning) to 50 across all result tables (sensitivity analysis in the Appendix A.1). Both of these values

were chosen by experimenting on 50 samples from the development sets of the respective datasets.

Auxiliary Graph Schema Creation: We randomly sample 10,000 questions (without answers and supporting documents) from the training data of HotpotQA and MuSiQue, for creating the graph schema. We provide further details about the auxiliary schema creation in the Appendix A.5.

6 Results and Analysis

Here, we study the performance of our method by investigating several key dimensions of multi-hop question answering (especially in the era of LLMs):

(i) Multi-hop reasoning capability
    (a) w.r.t automatic metrics
    (b) w.r.t human & semantic metrics
(ii) Performance w.r.t. different question types
    (a) Reasoning type-wise performance
    (b) Hop-wise performance
(iii) Query Information efficiency (input token count for reader LLM)
(iv) Measure of confident predictions

We also study the impact of the LLM used for knowledge triple extraction, and conduct further studies in the App A.1 and a case study in App A.6.

Multi-hop Reasoning: (a) **Evaluation using Automatic Metrics** - In Table 1 we report the performance of our method w.r.t. Exact Match (EM) and F1 scores for both datasets. We find that, across reader LLMs, our method consistently outperforms all baseline methods. This underscores the importance of our data organization and pruning process, retaining only the relevant information in input prompts for the reader LLMs. Notably, our method even outperforms the SoTA StructQA, which employs the CoT mechanism (Wei et al., 2022).

Dataset	HotpotQA	
Methods	H-Eval ($\uparrow$)	B-Score ($\uparrow$)
Base	86.5	87.00
StructQA	87.5	85.20
Our Method	**89.0**	**89.05**

Table 2: **Multi-hop Reasoning Evaluation (Semantic Metrics):** Human evaluation (H-Eval) on 100 samples & BERTScore (B-Score) for 1000 samples from HotpotQA.

(a) Human and Semantic Evaluation As generative models can generate long worded answers, its important to semantically evaluate the predicted answers. Thus, we use BertScore and human evaluators to judge the correctness of the predicted answers. We use GPT-4 reader based responses for this study and report the results in Table 2.

We observe that our method achieves a 1-2 point improvement on both the metrics w.r.t. the baselines, confirming the efficacy of our method.

Query Information Efficiency In multi-hop question answering, where reader LLMs like GPT-4 are employed, the presence of irrelevant information poses a significant challenge because (i) it increases the computational load (and thus the cost) (ii) and complicates the LLM's task of connecting disparate pieces of information across documents. Thus, efficiently managing input data by filtering out unnecessary content becomes crucial to both performance and cost-effectiveness. To quantify this factor, we measure the reader input token count and use an efficiency metric for input tokens. The query information effiency (or reader input token efficiency) metric is a normalized score between zero and one, computed through min-max normalization of input token count (min & max are based on input token counts of all methods in comparison). Our results, in Figure 1 and Table 3, show a significant reduction in average input token count across both datasets compared to baselines.

Datasets	HotpotQA	MuSiQue
Methods	Token Count($\downarrow$)	Token Count($\downarrow$)
Supp docs	1333.81	2361.36
StructQA	3078.85	5908.87
Our Method	**1230.90**	**1398.15**

Table 3: **Reader Input Token Count**: Comparing the average input token size across datasets to measure the efficiency of representing query relevant information.

Performance w.r.t. different question types
To analyze HOLMES's performance across question types and complexities, we perform an analysis based on reasoning types (with HotpotQA dataset) and hops (with MuSiQue dataset). We choose re-

spective datasets for each analysis based on the query type information available in the datasets.

(a) Reasoning type-wise performance
For any multi-hop question, the data generating process dictates the type of reasoning required to answer the question. HotpotQA dataset contains questions which fall in two categories w.r.t. reasoning - (i) bridge question (ii) comparison questions. We advise the reader to refer to the dataset paper (Yang et al., 2018) for more details. We report results across question categories in Table 4.

Type	# Samples	Base	StructQA	HOLMES
Bridge	787	0.55	0.56	**0.64**
Comparison	213	0.55	0.51	**0.71**

Table 4: **Performance Across Reasoning Types:** Evaluation of different methods. *Base* - Reader LLM with supp docs.

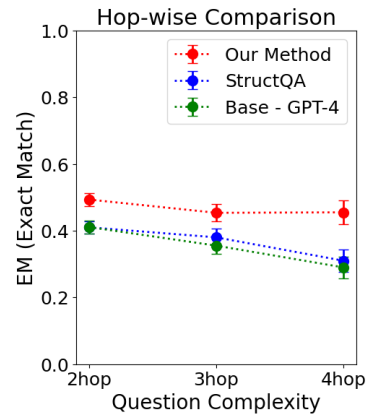**(b) Hop-wise Performance Comparison**

Figure 4: **Hop-wise Performance:** Comparison on MuSiQue dataset with varying question complexity. Bars denote standard error - $\sqrt{\frac{em(1-em)}{n}}$.

We evaluated the hop-wise performance of our method on MuSiQue dataset (Figure 4). This evaluation aimed to determine if our performance improvements (in Table 1) were primarily attributed to the system's ability to answer 2-hop questions, which are less complex than 3 and 4-hop questions. We find that our method maintains its performance across an increasing number of hops and outperforms the baselines too.

Measure of Confident Predictions LLMs demonstrate reasoning capabilities but often struggle to recognize their limitations in understanding queries or retrieving correct answers. We found that structured inputs and task-dependent reasoning steps, including the option for models to indicate uncertainty, alleviates this issue. Inspired by this, we introduce the *Self-Aware Exact Match* (Self-

Aware EM) score to assess LLMs on complex QA tasks. This metric evaluates the accuracy and confidence of model responses, focusing on instances where the model provides answers with high confidence (or self-awareness). It aims to highlight the model's precision and reliability, offering a nuanced view of its performance. It is defined as: Self Aware EM $= \frac{\sum_{q \in Q_A} \text{EM}(q)}{|Q_A|}$ where Q_A is the set of questions the system answers. We report the Self-Aware EM scores in Figure 1.

In our evaluation using the HotpotQA and MuSiQue datasets, HOLMES opted for "None" or "No answer" in 3% and 10% of the samples, respectively (with GPT-4 as reader LLM). In contrast, StructQA always provided an answer, often inaccurately (see Table 1). Upon reviewing instances where HOLMES did not provide an answer, we found it was due to incomplete information in the input graph or a misunderstanding of the question. This self-awareness—recognizing when it cannot reliably answer—is a key feature of dependable systems. Our findings, detailed in Figure 1, show that HOLMES achieves significantly higher self-aware EM scores compared to other baselines.

7 Sensitivity & Ablation Analysis

Configuration				Performance		
Hyper KG	Prune	Aux. Schema	Comp. Fact Retrieval	EM	F1	Reader Inp Tokens
X	-	-	-	0.56	0.69	1867
X	X	-	-	0.56	0.70	1227
X	X	X	-	0.58	0.72	1220
X	X	X	X	0.61	0.77	1230

Table 5: **Impact of different components:** Performance metrics (EM, F1) and Reader Input Token Count for different configurations of HOLMES. **X** refers to "*included*", and **-** refers to "*excluded*" in the table. For example, in the first row, only Hyper-Relational KG is included in the HOLMES algorithm, every other component is removed. These results are reported on a set of 100 samples from HotpotQA dev set.

Ablation Studies In Table 5 we share the results of our ablation study that systematically evaluates the contribution of each component within the HOLMES framework.

Sensitivity Analysis We evaluated the effect of using a different LLM (gpt-4-1106-preview) for triple extraction, comparing our performance with StructQA in Table 6 (reader: GPT-4). Our approach consistently outperforms StructQA, although improved triple extraction models benefit both.

Datasets	HotpotQA (100 samples)			
Methods	EM (↑)	F1 (↑)	P (↑)	R (↑)
Triple Extractor: gpt-4-1106-preview				
StructQA	0.56	0.76	0.79	0.77
Our Method	**0.68**	**0.79**	**0.82**	**0.80**
Triple Extractor: gpt-3.5-turbo-1106				
StructQA	0.47	0.71	0.72	0.76
Our Method	**0.61**	**0.77**	**0.78**	**0.78**

Table 6: **Impact of Triple Extractor** on MHQA performance

8 Conclusion

We introduce HOLMES, an approach leveraging a hyper-relational KG to enhance multi-hop question answering by minimizing noise and refining relevant facts. Constructing an entity-doc graph from the question's supporting documents, we employ a level-order traversal, prune with an auxiliary graph schema, and utilize distilled graph as input for LLM-based answering. Our method achieves SoTA performance with a 20% improvement on HotpotQA and 26% on MuSiQue dataset w.r.t EM while using upto 67% fewer tokens to represent query relevant information w.r.t. SoTA methods.

9 Acknowledgement

Limitations

The following details outline areas for future improvements:

Possible Incompleteness in Constructed Graphs Extracting entities and triples to construct graphs via LLMs demonstrates high accuracy, yet the task of capturing all relevant triples faces inherent challenges. Structured information extraction remains an unsolved problem across the field, leading occasionally to incomplete graphs in our method as well

Increased Computational Effort: Our method's use of LLMs to generate an auxiliary schema introduces additional computational steps. While this increases the computational effort compared to prior methods, it's offset by our efficiency w.r.t fewer input tokens during inference.

While acknowledging the limitations of our approach, we remain optimistic, recognizing our work as a step forward in enhancing the capabilities of LLMs for multi-hop question answering.

Ethical Concerns

There are no ethical concerns associated with this work.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. 2020. Learning to retrieve reasoning paths over wikipedia graph for question answering. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. Comet: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779.

Salvatore Carta, Alessandro Giuliani, Leonardo Piano, Alessandro Sebastian Podda, Livio Pompianu, and Sandro Gabriele Tiddia. 2023. Iterative zero-shot llm prompting for knowledge graph construction. *arXiv preprint arXiv:2307.01128*.

Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. 2024. The power of noise: Redefining retrieval for rag systems. *arXiv preprint arXiv:2401.14887*.

Joe Davison. 2020. Zero-shot learning in modern nlp.

Zhenyun Deng, Yonghua Zhu, Yang Chen, Michael Witbrock, and Patricia Riddle. 2022. Interpretable amr-based question decomposition for multi-hop question answering. *arXiv preprint arXiv: 2206.08486*.

Junnan Dong, Qinggang Zhang, Xiao Huang, Keyu Duan, Qiaoyu Tan, and Zhimeng Jiang. 2023. Hierarchy-aware multi-hop question answering over knowledge graphs. In *Proceedings of the ACM Web Conference 2023*, WWW '23, page 2519–2527, New York, NY, USA. Association for Computing Machinery.

Yuwei Fang, Siqi Sun, Zhe Gan, Rohit Pillai, Shuohang Wang, and Jingjing Liu. 2020. Hierarchical graph network for multi-hop question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8823–8838, Online. Association for Computational Linguistics.

Gautier Izacard and Édouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880.

Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Structgpt: A general framework for large language model to reason over structured data. *arXiv preprint arXiv:2305.09645*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *International Conference on Learning Representations*.

Gyeong-Geon Lee, Ehsan Latif, Lehong Shi, and Xiaoming Zhai. 2023. Gemini pro defeated by gpt-4v: Evidence from education. *arXiv preprint arXiv:2401.08660*.

Ruosen Li and Xinya Du. 2023. Leveraging structured information for explainable multi-hop question answering and reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6779–6789.

Vaibhav Mavi, Anubhav Jangra, and Adam Jatowt. 2022. A survey on multi-hop question answering and generation. *arXiv preprint arXiv:2204.09140*.

Ethan Perez, Patrick Lewis, Wen-tau Yih, Kyunghyun Cho, and Douwe Kiela. 2020. Unsupervised question decomposition for question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8864–8880, Online. Association for Computational Linguistics.

Peng Qi, Haejun Lee, Tg Sido, and Christopher Manning. 2021. Answering open-domain questions of varying reasoning steps from text. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3599–3614, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Lin Qiu, Yunxuan Xiao, Yanru Qu, Hao Zhou, Lei Li, Weinan Zhang, and Yong Yu. 2019. Dynamically fused graph network for multi-hop reasoning. In

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 6140–6150, Florence, Italy. Association for Computational Linguistics.

Gowtham Ramesh, Makesh Narsimhan Sreedhar, and Junjie Hu. 2023. Single sequence prediction over reasoning graphs for multi-hop qa. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11466–11481.

Joshua Robinson, Christopher Michael Rytting, and David Wingate. 2022. Leveraging large language models for multiple choice question answering. *arXiv preprint arXiv:2210.12353*.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: an open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 4444–4451.

Haitian Sun, Tania Bedrax-Weiss, and William Cohen. 2019. 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)*, pages 2380–2390.

Haitian Sun, William W. Cohen, and Ruslan Salakhutdinov. 2021. Iterative hierarchical attention for answering complex questions over long documents. *arXiv preprint arXiv: 2106.00200*.

Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William Cohen. 2018. 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*, pages 4231–4242.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022a. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022b. Musique: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Somin Wadhwa, Silvio Amir, and Byron C Wallace. 2023. Revisiting relation extraction in the era of large language models. *arXiv preprint arXiv:2305.05003*.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023. Gpt-ner: Named entity recognition via large language models. *arXiv preprint arXiv:2304.10428*.

Yusuke Watanabe, Bhuwan Dhingra, and Ruslan Salakhutdinov. 2017. Question answering from unstructured text by retrieval and comprehension. *arXiv preprint arXiv: 1703.08885*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, and Enhong Chen. 2023. Large language models for generative information extraction: A survey. *arXiv preprint arXiv:2312.17617*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Semih Yavuz, Kazuma Hashimoto, Yingbo Zhou, Nitish Shirish Keskar, and Caiming Xiong. 2022. Modeling multi-hop question answering as single sequence prediction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 974–990.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Wenxuan Zhang, Yue Deng, Bing Liu, Sinno Jialin Pan, and Lidong Bing. 2023. Sentiment analysis in the era of large language models: A reality check. *arXiv preprint arXiv:2305.15005*.

A Appendix

In this section, we provide additional results and details that we could not include in the main paper due to space constraints. In particular, this appendix contains the following:

- Additional Results and Analysis
- Dataset Statistics
- Guidelines for Human Annotators
- Prompts used in HOLMES
- Auxilliary schema construction
- Case Study of HOLMES on HotpotQA dataset

A.1 Additional Results and Analysis

Extending our analysis in Section 6, here we report the following - (i) MHQA performance with Gemini-Pro reader LLM (ii) Sensitivity Analysis w.r.t. pruning process (iii) Sensitivity Analysis w.r.t. the number of depth of traversal while creating the Hyper-Relational KG.

Gemini-Pro MHQA Results

Datasets	HotpotQA		
Methods	EM (↑)	F1 (↑)	SA-EM (↑)
Reader: Gemini-Pro			
Base (w/ supp docs)	0.48	0.66	0.48
StructQA	0.49	0.66	0.52
Our Method	**0.58**	**0.67**	**0.66**

Table 7: **Multi-Hop QA performance** *SA-EM*: Self-Aware EM; *Base*: Only reader LLM; *supp docs*: supporting documents w.r.t. query

Similar to results with GPT-3.5 and GPT-4 reader LLMs (in Table 1), we observe consistent improvements across metrics. Thus demonstrating the efficacy of our method, across LLMs.
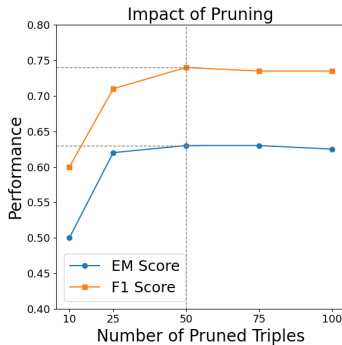
Impact of Pruning

Figure 5: **Impact of pruning** on MHQA performance in HotpotQA dataset

In Tables 1 and 6, we conducted our experiments by retaining 50 hyper triples after the pruning stage. Here we vary the number of hyper triples being retained after pruning and study the impact on the MHQA performance (EM and F1 scores).

Figure 5 shows that after 50 triples, the MHQA performance does not change indicating that the first 50 triples captures all the query relevant information. This number is dependent on the dataset, so we suggest conducting experiments to deduce this threshold.

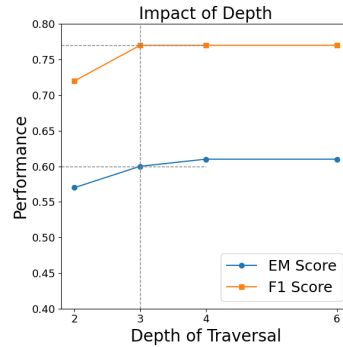### Impact of Depth in the Level Order Traversal

Figure 6: **Impact of Depth** on MHQA performance in HotpotQA dataset

In the Tables 1 and 6, we conducted our experiments by traversing the entity document graph upto a depth of four levels. As stated in Section 4.2.1, the depth of traversal depends on the maximum complexity of query we expect to see during inference. To verify this, we perform a sensitivity analysis on this variable. We experiment on 100 randomly sampled datapoints from HotpotQA development set(question complexity varies between 2-3 hops).

Figure 6 reflects that as the max query complexity of the dataset is three, beyond three levels of traversal the performance does not improve. Thus, verifying our claim. We choose a depth of four in the main paper as we wanted to cater to all the datasets.

Total Input and Output Token Lengths & Costs

Method	Inp Token Length (↓)	Out Token Length (↓)	Total Cost in $ (↓)	EM (↑)
StructQA	**9012**	3590	0.99	0.48
HOLMES	9388	**2524**	**0.85**	**0.57**

Table 8: **Performance and Cost Comparison:** Comparison of total input and output token lengths (across all LLM calls), total cost, and EM scores for different methods.

In table 8 we report the average total input and

output token lengths on HotpotQA dataset (w.r.t. 100 random samples from dev set). We also share the cost estimate w.r.t. the latest GPT-3.5 LLM (both as reader and triple extractor model)

The average input and output token lengths when simply using the reader LLM are 1334 and 10, respectively. Its worth noting that for performance improvements beyond LLMs on complex NLP datasets such as HotpotQA and MuSiQue, structured information extraction is immensely useful, as can be seen from the improvements across metrics in Table 1 of the paper.

A.2 Dataset Statistics

Dataset	Train	Dev
HotpotQA	90,447	7,405
MuSiQue	19,938	2,417

Table 9: **Train and Dev Statistics** - number of samples

Dataset	number of instances
MuSiQue 2-hop	667
MuSiQue 3-hop	366
MuSiQue 4-hop	200

Table 10: Hop-wise distribution of questions used for evaluation in Section 6 for MuSiQue

Table 9 provides details on the total number of training and development set samples in each dataset. From these, we have randomly sampled 10, 000 questions (unlabelled or unannotated) from the training set to create our auxiliary schema. For each dataset, we randomly sampled 1000 data points from the development set for evaluation purposes (as illustrated in section 5.1). Table 10 details the hop-wise question statistics of the MuSiQue dataset (HotpotQA dataset (Yang et al., 2018) does not provide hop-wise distribution of their dataset).

A.3 Guidelines for Human Annotators

In the evaluation phase of HOLMES (Table 2), human annotators played a crucial role in assessing the accuracy of the predicted answers. The process was structured as follows:

- Annotators were presented with a set of inputs for each question, which included:
  - The question
  - The gold answer
  - The predicted answer generated by one of the MHQA methods
- The primary task for the annotators was to determine the correctness of the predicted answer in comparison to the gold answer. This evaluation was binary, with annotators assigning:
  - A score of **0** if the predicted answer was deemed incorrect
  - A score of **1** if it was considered correct
- The evaluation process encompassed all methods: Base with supporting documents, StructQA, and HOLMES, each evaluated with a total of 100 questions. Therefore:
  - Each annotator was responsible for evaluating a comprehensive total of **300 questions**, covering all possible combinations of questions and methods.

The numbers in Table 2 represent the average scores from three human annotators.

A.4 Prompts used in HOLMES

In this section we detail the prompts used for solving different subtasks using LLMs.

A.4.1 Entity Extraction from supporting documents

For extracting named entities from the supporting documents, we use the following prompt. Few-shot examples used to improve the performance of the LLM is also included in the prompt.

```
Task: Extract ALL the named entities
from the given sentence (extract
all time intervals, names, dates,
organizations and locations).
  Examples:  Use  the  following
examples  to  understand  the  task
better.  Sentence: William Rast is

an American clothing line founded by
Justin Timberlake and Trace Ayala.
  Entities: William Rast, American,
Justin Timberlake, Trace Ayala

  Sentence:  The  Glennwanis  Hotel
is a historic hotel in Glennville,
Georgia, Tattnall County, Georgia,
built  on  the  site  of  the  Hughes
Hotel.
```

```
   Entities:    Glennwanis    Hotel,
Glennville,    Georgia,    Tattnall
County, Georgia, Hughes Hotel
```

A.4.2 Entity and Relation Extraction from the Question

Instead of separately extracting the entity and the relations from the inference query, we use a single LLM call. Below we detail the prompt along with the few-shot examples used.

```
Task:  extract all named entities
from  the  above  question.    Then
extract    all    the    important
information  (each   information
should  be  2-3  words)  needed  to
answer the question. Output format
- "entities:  [ent1, ent2, ...]
important relations: [info1, info2,
...]".  Please  note,  do not give
named  entities  in  the  'important
relations'
Use  the  following  examples  to
understand the task:

Question:   Who  is  the  author  of
the  book  that  inspired  the  movie
starring Tom Hanks as a symbologist?
Entities: [Tom Hanks]
Important  Relations:   [author  of,
inspired  the  movie,   stars,as  a
symbologist]

 Question: Did the company that Elon
Musk  co-founded  in  2002  eventually
merge  with  a  firm  that  had  been
contracted  by  NASA  to  resupply  the
International Space Station?
Entities:   ['Elon  Musk',  '2002',
'NASA',    'International    Space
Station']    Important  Relations:
['co-founded',     'merge     with',
'contracted by', 'resupply']
```

A.4.3 Subject and Object Entity type Estimation

As discussed in Section 4.2.2 and shown in Figure 3, to create the knowledge schema from the query, we need to estimate the subject and object entity types corresponding to a given relation. We use an LLM for the same and report the prompt below.

```
Task:  Generate  the  subject  and
object entity type for the following
relation,  w.r.t.    the  provided
context information.
relation: input
Context:  context ; DO NOT extract
partial info.
Output  format:   [<subject  entity
type>, <object entity type>]
```

A.4.4 Triples Extraction from supporting documents

Here we report the prompt along with the few-shot examples we used for knowledge triple extraction from raw text.

```
Task:  Comprehensively extract ALL
the  triples  (subject,  relation,
object) from below given paragraph.
Ensure that the subject and objects
in  the  triples  are  named  entities
(name   of   person,   organization,
dates  etc)  and  not  multiple  in
number.    You  will  be  HEAVILY
PENALIZED  if  you  violate  this
constraint.
  Examples:   Use   the   following
examples  to  understand  the  task
better.  Paragraph: William Rast is
an American clothing line founded by
Justin Timberlake and Trace Ayala.
It  is  most  known  for  their  premium
jeans. On October 17, 2006, Justin
Timberlake and Trace Ayala put on
their  first  fashion  show  to  launch
their  new  William  Rast  clothing
line. The label also produces other
clothing  items  such  as  jackets  and
tops. The company started first as
a denim line, later evolving into a
men's and women's clothing line.
 Triples:

 i. subject:       William    Rast,
    relation:      clothing    line,
    object: American
 ii. subject:       William    Rast,
    relation:  founded  by,  object:
    Justin Timberlake
iii. subject:       William    Rast,
    relation:  founded  by,  object:
```

Trace Ayala

iv. subject: William Rast, relation: known for, object: premium jeans

v. subject: William Rast, relation: launched on , object: October 17, 2006

vi. subject: Justin Timberlake, relation: first fashion show, object: October 17, 2006

vii. subject: Trace Ayala, relation: first fashion show, object: October 17, 2006

viii. subject: William Rast, relation: produces, object: jackets

ix. subject: William Rast, relation: produces, object: tops

x. subject: William Rast, relation: started as, object: denim line

xi. subject: William Rast, relation: evolved into, object: men's and women's clothing line

Paragraph: The Glennwanis Hotel is a historic hotel in Glennville, Georgia, Tattnall County, Georgia, built on the site of the Hughes Hotel. The hotel is located at 209-215 East Barnard Street. The old Hughes Hotel was built out of Georgia pine circa 1905 and burned in 1920. The Glennwanis was built in brick in 1926. The local Kiwanis club led the effort to get the replacement hotel built, and organized a Glennville Hotel Company with directors being local business leaders. The wife of a local doctor won a naming contest with the name "Glennwanis Hotel", a suggestion combining "Glennville" and "Kiwanis

Triples:

i. subject: Glennwanis Hotel, relation: is located in, object: 209-215 East Barnard Street, Glennville, Tattnall County, Georgia

ii. subject: Glennwanis Hotel, relation: was built on the site of, object: Hughes Hotel

iii. subject: Hughes Hotel, relation: was built out of, object: Georgia pine

iv. subject: Hughes Hotel, relation: was built circa, object: 1905

v. subject: Hughes Hotel, relation: burned in, object: 1920

vi. subject: Glennwanis Hotel, relation: was re-built in, object: 1926

vii. subject: Glennwanis Hotel, relation: was re-built using, object: brick

viii. subject: Kiwanis club, relation: led the effort to re-build, object: Glennwanis Hotel

ix. subject: Kiwanis club, relation: organized, object: Glennville Hotel Company

x. subject: Glennville Hotel Company, relation: directors, object: local business leaders

xi. subject: Glennwanis Hotel, relation: combines, object: "Glennville" and "Kiwanis"

A.4.5 LLM Reader

Below we report the reader LLM prompt. Please note that we instruct the LLM to reason in three steps - (i) relevant fact extraction (ii) reasoning based on the relevant facts (iii) final response (with an option to request details about named entities)

```
Question:{question}
Read the above question carefully,
understand the answer category. Now,
given you understand the question,
use the following facts to answer
the question. Please note that
the meaning of facts is HEAVILY
dependent on the context or the
document from which it was extracted.
prunned_and_verbalized_hyperkg
(Note: the above set of facts
```
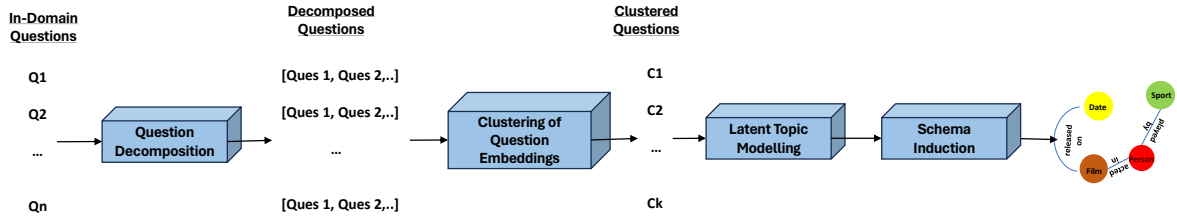
Figure 7: **Auxilliary Graph Schema Construction Overview**

```
can be noisy, so if ambiguous
information is present then focus
on the question and keywords in the
question - relations_in_question)
First, fetch the relevant set of
facts from the above taking into
consideration the context of the
fact (DO NOT generate your facts),
then by combining them answer the
question - question.
Instruction: Give the answer in the
following format -
Relevant facts: <facts (and context
of those facts) relevant to the
question>
Reasoning:       <Here,    give   your
thought    process    of   how    you
use     the      named     entities      -
named_entities_in_question,
relations - relations_in_question
of the question and accordingly
traverse the facts given above to
arrive at your answer>
Final  Response:      <If   all   the
necessary  facts  are  available  to
answer the question, express it in
3-4 words - Else, say None>
Further query: <if 'Final Response'
is None, then think step-by-step
using the 'Reasoning' and other
facts and state which named entity's
direct information is missing>
```

A.5 Auxilliary Schema Construction

As stated in Section 4.2.2, we use a collection of in-domain questions to construct a graph schema (See definition in Section 3), that captures the global level blueprint of information required to answer questions in the target domain. In this section, we detail the procedure used to create the same. We employ a four-step process, starting from the un-annotated in-domain multi-hop questions.

A.5.1 Question Decomposition

Given the complexity of multi-hop questions and the challenges in processing their embeddings, we first decompose these into simpler, single-hop questions. This decomposition is facilitated by the `llama-13b` LLM, which effectively breaks down complex questions into their constituent single-hop components. Decomposing $10,000$ questions using SoTA LLMs is very expensive, thus we opted for a smaller, open-weight LLM. We use few-shot prompting to ensure the model understands the task well. We use the following prompt for the same:

```
Task:       Decompose    a    multi-hop
question into a series of single-hop
questions to assist in finding the
answer in a step-by-step manner.
Instruction:
  i. Proceed with the decomposition
     according to the outlined chain
     of thought.
 ii. Only return the sub-questions,
     nothing else.
iii. Focus on the entities in the
     question
 iv. If decomposition is not clear,
     do not guess.

   Question:      Lily's     Driftwood
Bay   premiered    on    what   British
television channel that is operated
by a joint venture between Viacom
International Media Networks Europe
and Sky plc?
Chain of Thought Instructions:

 1. Identify   the   ultimate   goal
    of  the   question:   Find   the
```

```
   television  channel  on  which
   Lily's Driftwood Bay premiered.
2. Recognize  that  there  are  two
   pieces  of  information  needed:
   a)  Determine  the  British
   television channel operated by
   the specific joint venture. b)
   Find  out  if  Lily's  Driftwood
   Bay  premiered  on  this  channel.
3. Create single-hop sub-questions
   that will answer these pieces of
   information  separately.

Output:

1. Which    British    television
   channel  is  operated  by  a
   joint  venture  between  Viacom
   International  Media  Networks
   Europe and Sky plc?
2. Did   Lily's   Driftwood   Bay
   premiere   on   this   specific
   British television channel?
```

A.5.2 Cluster Question Embeddings

We then apply K-Means clustering to the embeddings of the decomposed single-hop questions to group similar questions. We use OpenAI embedding model (`text-embedding-ada-002`) to compute text embeddings. The optimal number of clusters (k) is determined by evaluating the clustering quality through a combined clustering evaluation score (CCES) which is a weighted sum of three normalized scores (normalized to lie between 0 and 1): silhouette score (S_{norm}), inverted Davies-Bouldin score (DB_{norm}), and Calinski-Harabasz score (CH_{norm}), with equal weightage assigned to each. These scores collectively assess the cohesion, separation, compactness, distinctness, and dispersion of the clusters. We use the elbow method for determining the optimal number of clusters. We report the elbow plot below.

Below, we report a couple of cluster centroids with a few data points (questions) around the centroid.

Cluster 1

- *Which football team did Kevin John Ufuoma Akpoguma play for?*
- *Who is the former Houston Texans head coach?*
- *Which NFL team currently employs him?*
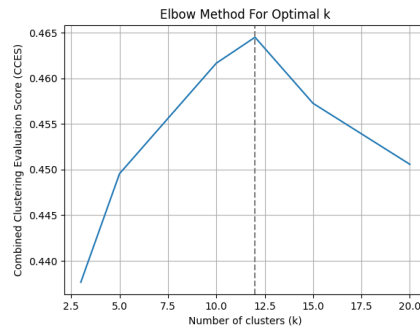
Cluster 2

Figure 8: **Elbow Plot** for Kmeans clustering on 10000 questions from HotpotQA dataset.

- *What is the name of the actress?*
- *Who are Rohan Bopanna and Cara Black?*
- *What is the name of the protagonist in the show "Pretty Little Liars"?*

As can be seen from the above clusters, they capture latent topics in the dataset such as sports (cluster 1) and movies (cluster 2).

A.5.3 Latent Question Category Modelling

The next step involves latent topic modeling to identify question categories within the single-hop question clusters. By extracting the cluster centroid question and the five questions closest to each centroid, we prompt an LLM (`gpt-4-1106-preview` in this case) to discern latent topics, focusing on entity types and relations rather than specific entities. The prompt we use deducing the latent question category is as follows:

```
Task: Extract latent topics with
focus  on  entity  types  (not  the
entities  themselves)  and  relations
in  below  given  questions.   These
topics  should  be  the  underlying
question  categories  for  these
questions.
Instructions:

  i. Keep in mind that I will later
     use  these  latent  topics  to
     create a KG schema such that the
     KG can answer questions such as
     the ones listed below.
 ii. Answer  in  1  sentence  using
     detailed and informative words
     only.
```

13279

```
iii. Consider        cross-question
     information too for determining
     the latent topics.
 iv. Directly give the latent topics
     without preamble text like "the
     latent topic is ...", and use
     bullets to separate topics.
```

A.5.4 Schema Induction

Finally, we prompt the LLM to generate a graph schema based on the broad question categories identified in the previous step. This schema is explicitly designed as an information organization blueprint relevant to the identified question categories, thereby facilitating a structured approach to answering questions within the domain. We use the following prompt for the same -

```
Task: Create a graph schema for
my KG using the broad question
categories that I want to be
answered by the KG.
The broad question categories are:
{question categories}
Instructions:

  i. Create a set of relations for
     a knowledge graph that clearly
     and unambiguously express the
     relationships between entities
     ensuring they are reusable,
     standardized, and semantically
     meaningful.
 ii. Generate a list of distinct,
     relevant, and comprehensive
     entities for a knowledge graph
     about wikipedia text, ensuring
     they are specific, meaningful,
     and cover all aspects of the
     domain.
iii. After extracting the entities
     and relation types, for
     the graph schema of the
     knowledge graph return the all
     the triplets "entity type -
     Relation Type - Entity Type" in
     the output as a list
```

```
 iv. Important: ensure that the
     triplets generated forms a
     connected graph
  v. Important: - nothing else other
     than the list of triplets
     should be returned. Format
     for the list: [("Entity type",
     "relation", "Entity type"),
     ("Entity type", "relation",
     "Entity type"),...]
```

A.6 Case Study of HOLMES on HotpotQA dataset

In this section, we showcase our constructed hyper-relational KG extracted from the supporting documents. We illustrate the corresponding reasoning process, highlighting relevant facts, and ultimately derive the answer for the question.

Now, consider the following example question from the HotpotQA dataset.

Question: *'What major truck road is located in Backford Cross?'*

Based on the question, our method discovers related supporting documents and then fetches structured information from them (Section 4.2.1). Below we list those discovered supporting documents.

Supporting Document: 1
Title: *Backford Cross*
Passage: *Backford Cross is a village on the Wirral Peninsula, Cheshire, England. It is a suburb of the town of Ellesmere Port and part of Cheshire West and Chester. Backford Cross is located around the A41/A5117 junction, south of Great Sutton and about 1.5 mi north of the village of Backford, near Chester. Backford Cross is largely made up of residential homes built from 1990 onwards and serves as a commuter village to Ellesmere Port and Chester, although inhabitants show no allegiance to either locality. The area is split between post-code districts, with parts of the village in Great Sutton, Ellesmere Port CH66 and other areas in Backford, Chester CH1.*

Supporting Document: 2
Title: *A5117 road*
Passage: *The A5117 is a road in Cheshire, England. It runs between Shotwick ( ) and Helsby ( ) and connects the A550 at Woodbank to the M56. As such it forms a northerly bypass to Chester and*

a shorter route between the North West and North Wales than the A55. The road is dualled west of the M56. There is roundabout with the A540 and at Dunkirk at the western terminus of the M56. East of the junction the road is single carriageway and crosses the A41 by way of a roundabout at Backford Cross. The A5117 intersects the M53 at Junction 10. This junction is just east of Cheshire Oaks Designer Outlet. The road then continues almost parallel to the M56, which it intersects at Junction 14, at which there is a Motorway service area. The road then continues south east to terminate where it joins the A56 at Helsby.

Supporting Document: 3
Title: *Strawberry Park, Cheshire*
Passage: *Strawberry Park and Strawberry Fields are suburbs in the town of Ellesmere Port, Cheshire West and Chester. They are located to the south of Hope Farm and to the west of Backford Cross.*

Supporting Document: 4
Title: *A41 road*
Passage: *The A41 is a major trunk road in England that links London and Birkenhead, although it has now in parts been superseded by motorways. It passes through or near various towns and cities including Watford, Kings Langley, Hemel Hempstead, Aylesbury, Solihull, Birmingham, West Bromwich, Wolverhampton, Newport, Whitchurch, Chester and Ellesmere Port.*

From these supporting documents, we create the Hyper-relational KG and further refine this graph to retain query relevant facts (Section 4.2.2). Below we report the same (blue colored text refers to the additional attributes of the hyper triple, red colored text refers to the subject entity, green colored text refers to the relation, and purple colored text refers to the object entity).

Distilled Hyper-Relational KG:

- *context:* 'Strawberry Park, Cheshire'
  subject: 'Strawberry Park'
  relation: 'is a suburb in'
  object: 'Ellesmere Port, Cheshire West'

- *context:* 'Strawberry Park, Cheshire'
  subject: 'Strawberry Fields'
  relation: 'is a suburb in'
  object: 'Ellesmere Port, Cheshire West'

- *context:* 'Strawberry Park, Cheshire'
  subject: 'Ellesmere Port'

- *relation:* 'located in'
  object: 'Cheshire West and Chester'

- *context:* 'Strawberry Park, Cheshire'
  subject: 'Strawberry Park'
  relation: 'located to the south of'
  object: 'Hope Farm'

- *context:* 'Backford Cross'
  subject: 'Backford Cross'
  relation: 'is located in'
  object: 'Wirral Peninsula, Cheshire, England'

- *context:* 'A5117 road'
  subject: 'A5117'
  relation: 'connects'
  object: 'A550 at Woodbank to M56'

- *context:* 'A5117 road'
  subject: 'A5117'
  relation: 'forms a bypass to'
  object: 'Chester'

- *context:* 'A5117 road'
  subject: 'A5117'
  relation: 'forms a shorter route between'
  object: 'North West and North Wales'

- *context:* 'A5117 road'
  subject: 'A5117'
  relation: 'is dualled west of'
  object: 'M56'

- *context:* 'A5117 road'
  subject: 'A5117'
  relation: 'has a roundabout with'
  object: 'A540'

- *context:* 'A5117 road'
  subject: 'A5117'
  relation: 'has a roundabout at'
  object: 'Dunkirk'

- *context:* 'A5117 road'
  subject: 'A5117'
  relation: 'intersects'
  object: 'M53 at Junction 10'

- *context:* 'A5117 road'
  subject: 'M53'
  relation: 'is intersected by'
  object: 'A5117 at Junction 10'

- *context:* 'A41 road'
  subject: 'A41'
  relation: 'is a major trunk road in'
  object: 'England'

- *context: 'A41 road'*
  subject: 'A41'
  relation: 'links'
  object: 'London and Birkenhead'

- *context: 'A41 road'*
  subject: 'A41'
  relation: 'superseded by'
  object: 'motorways'

- *context: 'A41 road'*
  subject: 'A41'
  relation: 'passes through'
  object: 'Watford'

- *context: 'A41 road'*
  subject: 'A41'
  relation: 'passes near'
  object: 'Kings Langley'

- *context: 'A41 road'*
  subject: 'A41'
  relation: 'passes near'
  object: 'Hemel Hempstead'

- *context: 'A41 road'*
  subject: 'A41'
  relation: 'passes near'
  object: 'Aylesbury'

- *context: 'A41 road'*
  subject: 'A41'
  relation: 'passes near'
  object: 'Solihull'

- *context: 'A41 road'*
  subject: 'A41'
  relation: 'passes through'
  object: 'Birmingham'

- *context: 'A41 road'*
  subject: 'A41'
  relation: 'passes near'
  object: 'West Bromwich'

- *context: 'A41 road'*
  subject: 'A41'
  relation: 'passes near'
  object: 'Wolverhampton'

- *context: 'A41 road'*
  subject: 'A41'
  relation: 'passes through'
  object: 'Newport'

This distilled graph (after verbalization and proper formatting (Section 4.2.3)) is provided as input, along with the question, to the reader LLM. For this example, we use the SoTA, `gpt-4-1106-preview` as the reader LLM. Below, we describe the LLMs response which comprises of three things:

- **Relevant facts** set of hyper triples which the LLM thought were important for the answering the question. These are verbalized hyper triples as the input to the LLM was also verbalized hyper triples.
- **Reasoning**: LLMs thought process of going from the distilled hyper-relational KG to the final answer
- **Final answer**: answer predicted by the LLM.

Our Method's Response

Relevant facts:

  - Backford Cross is located around A41/A5117 junction (This fact was extracted from document named "Backford Cross ")

  - A41 is a major trunk road in England (This fact was extracted from document named "A41 road ")

  - A5117 is a road in Cheshire, England (This fact was extracted from document named "A5117 road ")

Reasoning:

  The question asks for a major truck road located in Backford Cross. The fact that Backford Cross is located around the A41/A5117 junction indicates that these are the major roads in that area. The A41 is specifically described as a major trunk road, which fits the description of a "major truck road." The A5117 is also mentioned as a road in Cheshire, but it is not specified as a major trunk road. Therefore, based on the information provided, the A41 is the major truck road located in Backford Cross.

Final Answer:

  A41