Multi-Hop Question Answering over Knowledge Graphs

Janadhi Uyanhewage, Viraj Welgama, Ruvan Weerasinghe

***Abstract*—Multi-Hop Question Answering over Knowledge Graphs (MHQA-KG) plays a pivotal role in various applications, including but not limited to Question Answering, Recommendation Systems, and Semantic Search. Nevertheless, current models for MHQA have limitations in their ability to grasp all the information included in the question, resulting a reduction in accuracy when producing answers. In order to mitigate this limitation, this paper proposes a novel Multi-Hop Question Answering over Knowledge Graphs approach. It mainly utilizes question and path embedding to answer multi-hop questions, significantly improving accuracy. This approach effectively captures auxiliary information that may present in the question. The experimental findings provide evidence that the suggested methodology outperforms the current state-of-the-art models, achieving highly accurate outcomes with improvements.**

***Index Terms*—Question Answering, Knowledge Graphs, Multi-Hop, Path Embedding, Sentence Embedding**

I. Introduction

A Knowledge Graph (KG) is a powerful tool used to present and organize human knowledge by representing entities as nodes and establishing relationships between them using edges. Unlike traditional representations, KGs view entities as actual objects existing in the real world, rather than mere words. This means that entities are viewed as ”THINGS” rather than ”STRINGS” [1]. They are utilized by search engines like Google, Bing, and Yahoo, social media networks such as LinkedIn and Facebook, and question answering services like Siri and Alexa. KGs are structured based on the SVO triple format derived from Knowledge Bases (KBs). [2].

Question Answering (QA) aims to provide suitable answers to questions asked in natural language. QA over KGs has gained attention as an approach where a knowledge graph is used as the information source to answer questions [3]. This technique involves analyzing the question, identifying significant entities and relationships, and constructing a response using the KG. Compared to conventional methods such as keyword matching, QA-KG has been shown to offer more accurate answers [4].

In the domain of QA over KG, the concept of ‘Hop’ refers to the distance between two nodes (entities) in the KG [5]. It represents the number of connections (edges) needed to traverse from the topic entity in the question to the entity that provides the answer. For example, in a KG which contains information about countries and their capital cities, a question like ”What is the capital of Spain?” only requires a single hop from the entity representing Spain to the entity representing its capital, Madrid. Such questions are called Simple Questions (SQ). The field of Question Answering over Knowledge Graphs (KG) has gained increasing importance due to the growth of large-scale knowledge graphs. These systems leverage the vast amount of information in KGs to provide answers to natural language questions, enabling users to access relevant knowledge without expertise in data structures.

This research proposes a comprehensive four-step approach for addressing natural language questions over KGs. Our approach includes decomposing the question into Subject-Verb-Object (SVO) triples, identifying optimal subgraphs within the knowledge graph, generating sentence and path embeddings for question and all the paths in the subgraph respectively, and selecting the best answer. It is designed to handle both simple and complex questions, specially those requiring multiple hops within the KG. Complex Questions (CQ) in QA-KG involve traversing along multiple relationships within the graph to arrive at an answer. For instance, consider a knowledge graph that maps out the connections between companies, employees, and their respective roles. The question ”Who is the CEO of Google?” on this specific KG would require two hops: traversing from ”Google” to ”employee” and then from ”employee” to ”CEO.” This process is known as Multiple Hops (MH). MHQA over KG refers to addressing complex questions that involve reasoning over multiple pieces of information in the knowledge graph.

Recent studies in MHQA-KG focus on identifying the topic entity to which the question refers and exploring all possible connections from the topic entity to other entities that may yield the answer. This is achieved by decomposing the question into SVO triples. However, this decomposition method may lead to the loss of important details, introducing ambiguity during subsequent processing steps.

To address this limitation, our study proposes a novel approach for locating answers to MH questions over KGs. We first identify the topic entity of interest and its related

Correspondence: Janadhi Uyanhewage (E-mail: uyanhewajanadhi@gmail.com)
Received: 08-01-2024 Revised: 10-07-2024 Accepted: 17-07-2024

Janadhi Uyanhewage, Viraj Welgama and Ruvan Weerasinghe are from University of Colombo School of Computing, Sri Lanka

(uyanhewajanadhi@gmail.com, wvw@ucsc.cmb.ac.lk, arw@ucsc.cmb.ac.lk)

DOI: https://doi.org/10.4038/icter.v17i2.7281

relationships by examining the question. Then, we extract a subgraph from the KG based on these components identified using the question. Our approach utilizes embedding generation for both the question and all paths in the subgraph to address challenges associated with subgraph extraction and to reduce information loss.

II. Literature Review

A. Question Decomposition

Question decomposition techniques have been explored in various studies to enhance the performance of question-answering systems. [6] introduced SRO triples to represent sentence structure, improving information retrieval efficiency and search relevance. The START system, which transforms user queries into a structured representation using NLP techniques, enabling precise and accurate retrieval of information [7]. Some studies on question generation for educational purposes, simplifying sentences into concise questions using a rule-based approach [8]. Automated question generation and question answering tasks using syntactic parsing and classification based on elementary sentence structures were also researched in studies such as [9]. Some studeis proposed decomposition methods for enhancing deep semantic question answering systems [10], while some others introduced an unsupervised approach using question decomposition to address complex questions [11].

There were systems that developed a concise question-answering system for Portuguese, utilizing open information extraction techniques [12] while another set of question-answering systems were based on subject-verb-object triplets extracted from unstructured text [13]. There were also systems that developed a Question Generator tool that generates various types of questions based on user preferences [14]. Additionally, research has been conducted on question answering in Subject-Object-Verb (SOV) languages, focusing on relation extraction and verb-argument frame creation. A new methodology called Hierarchical Semantic Parsing (HSP) has been introduced for complex question semantic parsing, involving question decomposition, information extraction, and integration stages.

There was an approach to answer sequences of interconnected queries within a conversational framework [15], while there approaches that decomposed complex fact-based questions into manageable subquestions [16]. Papers such as [17] addressed the challenges of handling constraints and multiple relationships in question answering by decomposing questions into query graphs with comprehensive semantics.

Systems such as [18] tackled the problem of Multi-hop Reading Comprehension (RC) by breaking down complex questions into simpler sub-questions using DECOMPRC. Their technique predicted relevant text parts to generate sub-questions, achieving performance comparable to human-generated ones.

B. Subgraph Extraction

Various approaches have been proposed to improve Knowledge Base Question Answering (KBQA) systems. One approach focuses on utilizing subgraph structures to enhance inference. As an instance, certain systems were designed to integrate pre-trained language models with a Multi-Hop Graph Relation Network (MHGRN) module, thereby enabling multi-hop reasoning on subgraphs extracted from external knowledge graphs [19]. Some systems also proposed an offline mechanism to extract subgraphs based on frequently asked entities, improving the speed and coverage of online QA systems. [20] introduces TS-Extractor, which combines topology and semantic data to extract relevant subgraphs centered around user-selected focus nodes. There were studies that propose SubGraph Temporal Reasoning (SubGTR) for answering temporal-based questions over temporal knowledge graphs, incorporating the extraction of implicit knowledge, retrieval of pertinent facts, and application of subgraph logical reasoning [21]. [22] present a method based on Personal Page Rank to efficiently extract relevant facts and improve real-time question answering. Furthermore, a trainable subgraph retriever (SR) is explored to optimize KBQA performance, utilizing the methodology of weakly supervised or unsupervised pre-training and end-to-end fine-tuning alongside a reasoner (ongoing research). These approaches enhance KBQA by considering subgraph structures, leveraging pre-trained models, incorporating semantic and temporal reasoning, and optimizing retrieval processes.

C. Generation of Path and Sentence Embeddings

A sentence embedding pertains to a numerical representation of a sentence designed to captures its meaning. It encodes the meaning in a fixed-size vector format. Path and sentence embeddings are two types that focus on different language aspects, resulting in different generation algorithms. Distance metrics can be used to evaluate similarity between embeddings.

[23] introduces sentence encoding models that generate accurate semantic similarity scores, leveraging transformer-based architectures and attention mechanisms. Building on this, a QA system was proposed for open domain question-answering, utilizing word embeddings and syntax-based techniques [24]. This approach improves upon traditional methods and demonstrates efficacy across various datasets. Expanding on the use of deep learning, the project aims to streamline the question answering process by employing bidirectional LSTM models to generate embeddings for questions and answers, eliminating the need for traditional linguistic tools and feature extraction.

In order to mitigate the shortcomings inherent in Knowledge Base Question Answering (KB-QA), this study suggests the utilization of pre-trained BERT language model embeddings alongside a multi-head attention encoder as a potential solution. This enhances the embeddings and filters candidate answers based on the relationships between the question and knowledge base context. In a similar vein [25] introduce novel techniques for machine reading comprehension, incorporating ETC, global-local attention mechanisms, and follow-up question-answer pairs to achieve outstanding performance.

There was a different approach by infusing question answering tasks into the pre-training of language models [26]. Their QA-Infused Pre-training (QAP) framework, which includes question embeddings, demonstrates superior performance in question answering benchmarks. [27] contribute to the field by developing a self-learning humanoid robotic system that effectively answers questions using word embeddings.

In the context of context paragraphs, [28] introduce the Bi-Directional Attention Flow (BIDAF) network, which models context paragraphs and it employs bidirectional attention flow to create a context representation that is query-aware. [29] propose a framework for medical question answering using supervised sentence embedding learning, incorporating self-attention and scoring techniques to capture semantic and syntactic features.

Exploring the design of sentence embeddings, and leveraging bilingual data to enhance sentence semantics, [30] propose a generative model. [31] contribute Sent2Vec, an unsupervised model that combines word vectors and n-gram embeddings to generate high-quality sentence embeddings.

Investigating the impact of supervision signals on sentence embeddings, this research compares the SBERT and DefSent models to optimize natural language processing tasks. Finally, [32] introduce KEQA, a powerful tool for answering straightforward questions by improving MHQA-KGs through KB embeddings, accurately identifying topic entities and predicates in KGs.

D. Selecting the answer from multiple potential options

To facilitate answer selection in natural language processing, an effective approach is to utilize distance metrics and cosine similarity for sentence embedding similarity. By transforming the question and subgraph paths into vector representations using embedding techniques, the closeness or similarity between the embeddings can be determined using metrics such as L1 distance or cosine similarity. These metrics effectively measure the distance or similarity between vectors in the embedding space, indicating semantic similarity when the similarity between sentence embeddings is small [33].

Building upon this concept, this research addresses the challenge of answer selection in natural language processing by introducing an innovative approach that utilizes learned semantic embeddings for words and sentences. By employing sentence embeddings, the research aims to develop a latent representation capturing the analogical relationship between two sentences, enabling a nuanced assessment of likeness between a question and its response. Leveraging the analogical proportion within sets of four sentences, where the differences between questions and answers exhibit a comparable pattern, this approach offers a novel method to improve answer selection performance by facilitating a more comprehensive understanding of the relationships between text pairs [33].

In the context of medical question answering, an innovative framework is presented aimed at improving the precision and efficiency of the task. The researchers leverage supervised sentence embedding learning and incorporate the self-attention mechanism introduced by [29] to capture meaningful word blocks and informative entities. By assigning attention weights to each word and employing a contextual layer, the framework generates powerful sentence embeddings that capture the intricacies of medical terminologies.ithin these embeddings. To leverage the information contaned in these embeddingd, the framework employs a pair of scoring techniques, namely Semantic Matching Scoring (SMS) and Semantic Association Scoring (SAS). This comprehensive approach combining supervised sentence embedding learning and self-attention structures shows promise in effectively addressing medical question answering challenges.

E. High Impact Papers

EmbedKGQA, a groundbreaking method proposed in [34], addresses the challenge of capturing complex connections in KGs for multi-hop KGQA. It leverages KG embeddings, excelling in sparsely connected KGs and outperforming state-of-the-art baselines on benchmark datasets. EmbedKGQA's unique approach of using KG embeddings for multi-hop KGQA relaxes the neighborhood constraint and performs well without external text, making it a promising solution for addressing KG sparsity.

The START Information Server, developed by [7], is a sophisticated multimedia information retrieval system that efficiently retrieves multimedia information segments through English queries. It uses the START Natural Language system and natural language annotations to generate a knowledge base and retrieve information. With its valuable capabilities, the START Information Server serves as an effective tool for constructing and querying knowledge bases in English.

In [13] present an innovative question answering system that utilizes semantic graphs. This system provides comprehensive explanations through graphical displays of documents, subject-verb-object triplets, and summaries. It extracts triplets using statistical and OpenNLP parsers, combines question answering and document retrieval functions, and offers multiple exploration methods.

In [35] introduce a novel approach to QA systems that leverages semantic graphs for providing explanations to natural language questions. The system extracts subject-verb-object triplets from documents and generates visual representations, facts, and summaries to offer explanations. It utilizes a hybrid AP technique and achieves superior performance in the BioASQ challenge [36].

In the evolution of QA systems, two primary strategies have emerged: text-based and knowledge-based. Text-based QA involves question processing, document retrieval, and answer extraction, while knowledge-based QA relies on knowledge bases. Deep learning models for QA can be categorized as representation-based, interaction-based, and hybrid models, each with different approaches for matching questions and answers.

Utilizing a complete knowledge graph can be computationally challenging. In [37] propose a technique for extracting a domain-specific subgraph to reduce graph size and improve computational efficiency in recommendation systems. The methodology achieves high accuracy and outperforms existing techniques in book and movie domain recommendations.

This work addresses the challenge of effectively querying knowledge bases using NLQs and proposes a novel approach that utilizes binary templates for question comprehension. It decomposes input questions into binary queries, improving the accuracy and effectiveness of the querying process. Empirical assessments demonstrate the efficiency of this approach in comprehending natural language inquiries and retrieving structured knowledge databases.

Overall, the proposed methods and approaches presented in these works contribute to advancements in KGQA, information retrieval, semantic graph-based QA, and querying knowledge bases using NLQs.

III. Research Methodology

A. Background for Methodology

1) Knowledge Graphs: Given a set of entities, denoted as ϵ, and a set of relations between them R, a Knowledge Graph G can be defined as a collection of SVO triples, denoted as K', where $K \in \epsilon \times R \times \epsilon$. A triple is denoted as (h, r, t), where $h, t \in \epsilon$ representing subject, object, and $r \in R$ connecting them respectively, and the (h, r, t) represents all available facts in the knowledge graph. In the task of knowledge graph question answering, the goal is to extract an entity $e_t \in \epsilon$ from a question q in natural language that contains a head entity $e_h \in \epsilon$, such that the extracted entity correctly answers the question.

2) Embedding Generation: For each question q a sentence embedding S_q is generated by transforming the q into a vector of a fixed-size. Similarly path embeddings $P_{hi},\ i = 0..n$ is generated for each K in the subgraph, where h is the base entity, and n is the total number of paths in the subgraph.

B. High-level Methodology

The methodology begins by decomposing the question q into SVO triples to identify e_h and R involved. This is followed by extracting a subgraph from the KG using these entities and relations as key identifiers. Next, a question embedding (Sq) is generated for the question, and path embeddings $P_{hi},\ i = 0..n$ are obtained for the paths in the subgraph. Finally, the L1 distance metric is applied to check the closeness or similarity between two embeddings. Then the best path that is selected based on the closeness or similarity between the two embeddings. The four stages are as follows,

1) Question Decomposition Module - Decomposes the question in to Subject, Verb, Object triples, which is crucial for extracting the necessary elements for subgraph extraction. By the Question Decomposition method, both eh and the R in the question are identified. This module caters both simple and complex questions. Simple questions involve a single SVO triple, while complex questions may have multiple SVO triples.
2) Subgraph Extraction Module - Extracting the subgraph encompasses the relevant topic entity.This module extracts the relevant subgraph from the KG based on q. It decomposes the question into SVO triple/s and selects an entity-relation pair from the set of SVO triples. Then extracts all K paths from the KG corresponding to the subject/object-verb pair, resulting in a subgraph that encapsulates all relevant information pertaining to q. This process acts as an effective KG pruning method, greatly enhancing system performance.Generate a corresponding Sq for the given q, enables comparison with path embeddings within the subgraph. This transformation of the q into a fixed-size vector allows to capture question semantics preserving all information in the q to reduce information loss.
3) Question Embedding Generation Module - Generating the embedding for the question.
4) Path Embedding Generation Module - This module generates embeddings P_{hi} for all K paths in the subgraph. Path embeddings are created by converting the paths into fixed-length vectors. These embeddings are then used by the employed by the Answer Selection Module to identify the most optimal correspondence for the provided q, enhancing the accuracy of the subgraph-based question answering system.

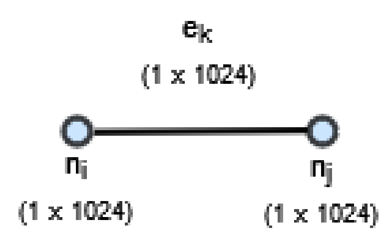

Fig. 1. Relation (edge) e_k connecting the entities (nodes) n_i and n_j

(n_i, e_k, n_j) x w = S_{ai}
where w is the trainable weight matrix
Single-Answer-path = $arg_{min}(\beta_{hi})\ i \in [0, n]$
Multiple-Answer-path = $sigmoid(1 - \beta_h)$

5) Answer Selection Module - This module utilizes embeddings generated by the Question Embedding Generation Module and the Path Embedding Generation Module [38]. It selects the path embedding that shows

the highest degree of closeness to the embedding of the question. Sentence embeddings are compared using the L1 distance metric to calculate the distance between them. The module selects the best answer by evaluating the absolute differences the between the embedding of the question (Sq), and each of the path embeddings (Phi), choosing the path with the closest embedding to the question. However, for multi-hop questions, the selected path may not directly provide the answer entity but instead leads to it. Therefore, the path is chosen based on its ability to enable the answering of the multi-hop question. When the module receives the question (Sq), it obtains the Subject and Verb combination, while path embeddings include the Subject-Verb-Object (SVO) triple. The module selects the path with the closest embedding to that of the question based on the calculated L1 distance.

$$\beta_{hi} = SIM(S_{hi}, S_q) \text{ Single-Answer-path} = arg_{min}(\beta_{hi})\ i \in [0, n]$$
$$\text{Multiple-Answer-path} = sigmoid(1 - \beta_h)$$

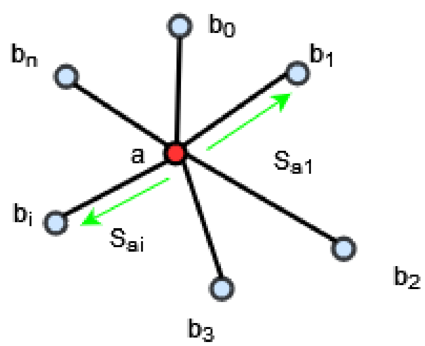

Fig. 2. Extracted subgraph around the topic entity a identified after the question decomposition phase

C. Prediction

Applying a sigmoid function to the L1 layer generates predictive probabilities between 0 and 1, crucial for the objective function and backpropagation. The model's predictions are represented by a 43,234 x 1 vector of values between 0 and 1 after sigmoid transformation. Optimization and dynamic answer extraction employ the adaptive thresholding method, using a criterion of 0.76 to evaluate probability of a correct answer. When the value exceeds threshold, it is considered a viable answer, and the corresponding index is identified from an array with a value of 1, enabling accurate entity identification.

The adaptive thresholding method optimizes answer extraction by evaluating probabilities against a specific criterion. Values exceeding the threshold are potential answers, and their corresponding indices with a value of 1 are identified. This technique improves accuracy by enabling precise predictions and identification of linked entities

D. Model Objective Function

The selection of the binary cross entropy serves as the model's objective function due to its compatibility with the multi-hot encoding technique. The effectiveness of the model performance is evaluated through the overall binary cross entropy metric.

This research adopts a multi-label classification approach for formulating the objective function to handle scenarios where a question can have multiple responses. The final layer of the model employs the sigmoid function in combination with another vector and the multi-hot encoding scheme to compute predictions and target set binary cross-entropy.

IV. Experiment

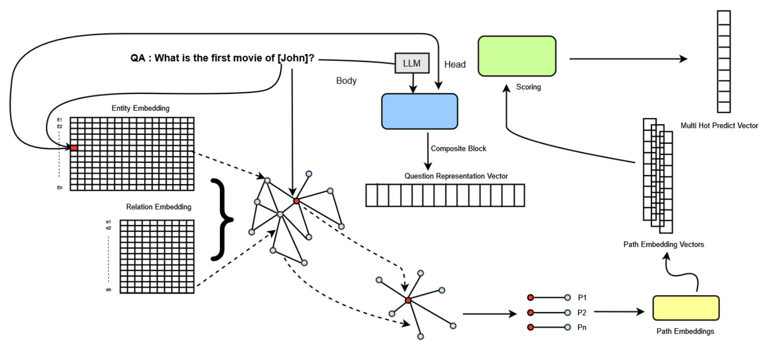

Fig. 3. An overview of the proposed model's training methodology.

A. Data Reading

1) Knowledge graph data reading: The Knowledge Graph is structured as entity-relation-entity triplets, with each line representing a connection between two entities. Entities and relations are separated by spaces, and multi-word relations are joined with underscores. Unlike other MHQA datasets, the Knowledge Graph is not composed of textual documents but is machine-readable, eliminating the need for additional natural language processing. Appendix A.1 provides the source code. To evaluate the algorithm's performance, the Knowledge Graph is commonly partitioned into train, test, and validation sets for hold-out evaluation. The MetaQA dataset contains information on 43,234 entities and 9 relations per direction (18 relations in total). Incorporated within the framework are pre-existing 400-dimensional features pertaining to each entity and relationship. The ComplexSQ model, a widely recognized knowledge graph extraction approach, was used to train the MetaQA dataset, which includes both a Knowledge Graph dataset and a QA dataset. The Knowledge Graph dataset comprises 10,000 links, with 1,000 in the training and validation sets, and 1,000 in the test set. The validation and test sets are subsets of entities used during training, ensuring dataset robustness for further research and experimentation [39].

2) QA Data reading: The QA dataset reading section includes textual question data categorized into three types: MetaQA 1Hop, MetaQA 2Hop, and MetaQA 3Hop. Each question follows a consistent structure of Head-Body-Tail, where head entity represents the starting entity, the body shows the interconnection between various entities, and the tail indicates the answer. For example, a sample question from the dataset is: "Who directed The Godfather?". Entities in the dataset are delimited by square brackets during the process of reading QA data, and a regular expression (regex) is used to extract and isolate these entities.

3) Pre-trained embedding reading: The MetaQA dataset, with 43,234 entities and 9 unidirectional relations, has been extensively simulated across bidirectional knowledge graph datasets. Pre-trained embeddings for entities and relations are provided, each with 400 features. Compressed npz files, entity.npz (43234×400) and relation.npz (18×400), contain the embeddings. Appendix A.8 showcases the source code for this functionality.

B. Graph Building

1) E-R-E path extraction: The main goal is to extract bidirectional links between head entities from the training knowledge graph dataset and create an incidence matrix to represent entity relationships. The incidence matrix consists of cells indicating relations between entities, with values representing the type of relation. To optimize sub-graph extraction, a path dictionary is used to map head entities to their outgoing links.

To generate entity-relation-entity (ERE) pairs for the graph, the train, test, and validation KG data in text files are scanned and sorted into lists based on the starting or head node. This organization results in an ERE path, which efficiently structures the data in the incidence matrix. Each row of the incidence matrix represents a distinct set of data.

2) Graph building: In graph theory, graph representation often involves a matrix, specifically the incidence matrix. This matrix is defined using ERE triplets extracted in previous steps. Techniques for constructing the incidence matrix include assigning numerical values (0 to 17) to represent connections, and using binary encoding (1 for presence, 0 for absence). In this research, we choose the first option and use the assigned numerical values in the incidence matrix.

3) Node and edge feature assignment: A torch embedding technique constructs an embedding space by converting keys and features associated with each node into a dictionary format. Using the torch embedding technique, an embedding is generated for each entity (node) and relationship (edge) within the Knowledge Graph by utilizing the dictionary.

C. DataLoader and Datasets

The Dataloader function serves a specialized purpose in the management of Question-Answer data, distinct from its use in handling Knowledge Graph data. This distinction arises from the direct feeding of KG data into the model and its training in batches or per epochs exclusively for QA data.

1) Head, Tail, and Question body selection: The data from the Knowledge Graph are fed into the model as distinct sets, namely the head, body, and tail. It is noteworthy that the tail is transformed into a one-hot representation that is of size 43,234. The decision behind using this encoding method is rooted in the potential for multiple valid answers to certain questions. The ultimate result generated by the model is conveyed in the form of a one-hot encoded representation.

2) Batching: The term "batching" refers to the process of feeding a set of headers, bodies, and tails, as opposed to feeding an individual question-answer format. In graph modeling, batching is typically not a widely adopted technique due to its intricate implementation. However, our methodology employs batching to effectively manage the extensive number of entities available and to accommodate time constraints.

3) Head entity extraction: This section primary focuses on the identification of the entity within the knowledge graph (head entity). This identification process relies on the extraction of the topic entity from the question. To ensure accurate localization of the appropriate head entity within the knowledge graph, it is imperative to make use of the topic entity identified within the provided question. This process allows for efficient and precise mapping of relevant entities, enabling a more comprehensive understanding of the relationships and connections within the graph.

4) Sentence embedding generation: Sentence embeddings can be generated using a sentence embedding model, which entails generating embeddings for each word within a given sentence and subsequently combining them to construct a representation at the sentence level. To achieve a standardized representation, both embeddings undergo processing via a composition unit, which employs non-linear operations and batch normalization techniques to compose and fuse the embeddings into a meaningful sentence embedding. The resultant single sentence embedding is designed to capture the inter-relationships between entities present in the sentence.

5) Subgraph extraction: In this stage, a subgraph is extracted from the complete KG, centered around the topic entity. The subgraph's size can vary based on analytical requirements, ranging from a single hop to multiple hops. Traditionally, the subgraph is obtained using representative relationship vectors. However, previous research employed an alternative approach. Initially, a set of entity embeddings with 43,234 elements was established, followed by a correlation analysis to identify highly correlated entities as potential outputs. Instead of directly generating entities, a mask was created, designating potential entities as "1" and the rest as "0." This mask was then multiplied with 43,234, extracting relevant floating-point features only for entities directly connected to the relevant entity, while assigning a value of "0" to all other entities.

6) Relation extraction: In the preliminary stage, we employ a non-linear function to facilitate the analysis process and explore the correlation between the subgraph entity's feature set and the relation's representative vector. Entities demonstrating significant correlation are identified as potential answers. Our novel approach utilizes relative distance to extract answer entities more effectively and with mathematical stability. By evaluating the relative distance between the sentence embedding and potential paths, we aim to identify the most reliable path embedding. This is achieved by computing the L1 distance over residual vectors and selecting the embedding key with the minimum distance as the prospective answer.

V. Results and Discussion

A. Metrics

Experiments were performed on the widely used WebQuestionSP and MetaQA datasets. MetaQA, with over 400,000 questions in the movie domain, provided three complexity levels: 1-hop, 2-hop, and 3-hop questions [40]. Dataset was partitioned into training, testing, and validation subsets. Evaluations were conducted in both ”full” and ”half” settings, using the complete knowledge graph or omitting 50

Understanding and utilizing auxiliary information is crucial for question answering models, as it provides essential cues for accurate answers. An experiment was designed to evaluate the significance of auxiliary information in the question-answering task. A comparison model was trained to generate sentence embeddings solely for entity and relation composition, while the proposed model incorporated auxiliary information in the sentence embedding generation process.

Another experiment focused on addressing the path discontinuity issue in graph-based question answering systems. A sub-graph extraction mechanism was incorporated to construct a sub-graph based on the entity node and edge present in the question, effectively addressing non-existent relations and the path discontinuity problem. The modified model outperformed state-of-the-art models, highlighting the effectiveness of the sub-graph extraction mechanism.

Experimental findings showed the potential of the newly introduced model to enhance the effectiveness of graph-based question answering systems. By incorporating a sub-graph extraction mechanism, the model effectively accounted for non-existent relations and addressed the path discontinuity problem. This methodology can be implemented in similar models to improve performance and provide more accurate results.

B. Evaluation

For the evaluation of the novel approach, Hits at N (H@N) metric was employed. For each question posed, we provide the feasible responses in increasing order of their similarity scores. Subsequently, we determine the percentage of correct answers ranked within the top N positions. This approach aids in comprehending the efficiency of the model in identifying the correct responses with precision.

$$Hits@N = \sum_{i=1}^{|Q|} 1 \text{ if } rank_{(s,p,o)_i} \leq \text{N}$$

where Q is a set of triples, and (s,p,o) is a triple $\in$ Q.

C. Results

This section summarizes the experimental results of our novel approach for answering multi-hop questions over knowledge graphs. Due to limitations included in the limitations section, the evaluation was conducted solely on the MetaQA dataset, which contains questions with one to three hops. The proposed model outperformed state-of-the-art models across all three sub-datasets of MetaQA, achieving high scores of 83.8%, 98.0%, and 62.0% for 1-hop, 2-hop, and 3-hop questions, respectively, in the full dataset. Even in a demanding scenario with 50% of triplets intentionally removed, the model achieved scores of 91.4%, 90.8%, and 63.9% for 1-hop, 2-hop, and 3-hop questions in the half dataset [40]. Particularly in the challenging 2-hop and 3-hop datasets, the proposed model demonstrated comparable or superior performance compared to previous state-of-the-art models. The evaluation results are summarized in Table 5.2 (MetaQA-full and WebQSP comparison) and Table 5.3 (proposed model vs. MetaQA-half). Although the current model has limitations, addressing them can potentially lead to even higher accuracy levels than state-of-the-art models.

TABLE I
Evaluation results (H@1) of the proposed model and SOTA models

Methods	WebQSP	MetaQA 1-Hop	MetaQA 2-Hop	MetaQA 3-Hop
KV-Mem [41]	46.7	96.2	82.7	48.9
VRN [21]	-	97.5	89.9	62.5
GraftNet [42]	66.4	97	94.8	77.7
PullNet [43]	68.1	97	99.9	91.4
EmbedKGQA [34]	66.6	97.5	98.8	94.8
DCRN [44]	67.8	97.5	99.9	99.3
Our Model	-	83.8	98.0	62.0

TABLE II
Evaluation results (H@1) of the proposed model and MetaQA - half models

Methods	MetaQA 1-Hop	MetaQA 2-Hop	MetaQA 3-Hop
KV-Mem [41]	41.8	37.6	48.9
GraftNet [42]	97	94.8	77.7
PullNet [43]	97	99.9	91.4
EmbedKGQA [34]	66.6	97.5	98.8
DCRN [44]	97.5	99.9	99.3
Our Model	91.4	90.8	63.9

VI. Future Work and Conclusion

A. Future Work

The research study successfully resolved the primary challenge through model implementation, meeting initial objectives based on assessment metrics. However, limitations were encountered due to time constraints and insufficient computational resources. Acquiring suitable resources can address these issues, while further improvement and generalization of the model using the WebQuestionSP dataset are possible. These findings demonstrate the model's potential for future research and development.

The discussed challenges and limitations emphasize the importance of efficient and accessible resources for training machine learning models. Particularly in regions with limited access to high-end computing resources, cloud-based platforms like Amazon Web Services (AWS) and Microsoft Azure can offer scalable and cost-effective solutions for machine learning training. It's crucial to consider regulatory and infrastructure limitations associated with these platforms.

B. Conclusion

In conclusion, our novel approach introduces a five-step methodology that utilizes knowledge graphs for accurate natural language question answering. Evaluation on multiple knowledge graphs confirms the efficiency of our proposed methodology, surpassing SOTA models.

The role of ”hops” is crucial in multi-hop question answering over knowledge graphs, involving reasoning over multiple pieces of information and traversing relationships to make inferences. Recent research emphasizes identifying the referenced entity and exploring possible connections to find answers. Our methodology identifies optimal subgraphs, extracts candidate answers, and selects the most appropriate response, addressing a wide range of questions including multi-hop ones. With advanced techniques, our approach surpasses existing methods in accuracy.

References

[1] P. Hitzler, M. Krötzsch, and S. Rudolph, "Foundations of semantic web technologies," *Foundations of Semantic Web Technologies*, 08 2009.

[2] L. Ehrlinger and W. Wöß, "Towards a definition of knowledge graphs," 09 2016.

[3] C. Paris, "Towards more graceful interaction: a survey of question-answering programs," 04 2023.

[4] D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager, N. Schlaefer, and C. Welty, "Building watson: An overview of the deepqa project," *AI Magazine*, vol. 31, pp. 59–79, 09 2010.

[5] V. Mavi, A. Jangra, and A. Jatowt, "A survey on multi-hop question answering and generation," 04 2022.

[6] K. Boris, "Annotating the world wide web using natural language," 06 2001.

[7] B. Katz, "From sentence processing to information access on the world wide web," 01 1997.

[8] M. Heilman and N. Smith, "Good question! statistical ranking for question generation." pp. 609–617, 01 2010.

[9] C. Y. Ali, Husam and S. A. Hasan, "Automatic question generation from sentences," pp. 213–218, Jul. 2010. [Online]. Available: https://aclanthology.org/2010.jeptalnrecital-court.36

[10] S. Hartrumpf, "Semantic decomposition for question answering," pp. 313–317, 01 2008.

[11] E. Perez, P. Lewis, W.-t. Yih, K. Cho, and D. Kiela, "Unsupervised question decomposition for question answering," 02 2020.

[12] R. Rodrigues, "Rapport: A fact-based question answering system for portuguese," 2017.

[13] L. Dali, D. Rusu, B. Fortuna, D. Mladenić, and M. Grobelnik, "Question answering based on semantic graphs," 04 2023.

[14] D. Vu and J. Blake, "Design and development of a question generator for learners of english," *SHS Web of Conferences*, vol. 102, p. 01011, 01 2021.

[15] M. Iyyer, W.-t. Yih, and M.-W. Chang, "Answering complicated question intents expressed in decomposed question sequences," 11 2016.

[16] A. Kalyanpur, S. Patwardhan, B. Boguraev, A. Lally, and J. Chu-Carroll, "Fact-based question decomposition in deepqa," *IBM Journal of Research and Development*, vol. 56, pp. 13:1–13:11, 05 2012.

[17] S. Shin and K.-H. Lee, "Processing knowledge graph-based complex questions through question decomposition and recomposition," *Information Sciences*, vol. 523, 02 2020.

[18] S. Min, V. Zhong, L. Zettlemoyer, and H. Hajishirzi, "Multi-hop reading comprehension through question decomposition and rescoring," pp. 6097–6109, 01 2019.

[19] Y. Feng, X. Chen, B. Lin, P. Wang, J. Yan, and X. Ren, "Scalable multi-hop relational reasoning for knowledge-aware question answering," 05 2020.

[20] B. Fu, Y. Qiu, C. Tang, Y. Li, H. Yu, and J. Sun, "A survey on complex question answering over knowledge base: Recent advances and challenges," 07 2020.

[21] H. Zhang, J. Cai, J. Xu, and J. Wang, "Complex question decomposition for semantic parsing," pp. 4477–4486, 01 2019.

[22] S. Aghaei, K. Angele, and A. Fensel, "Building knowledge subgraphs in question answering over knowledge graphs," pp. 237–251, 01 2022.

[23] D. Khashabi, "Reasoning-driven question-answering for natural language understanding," 08 2019.

[24] M. Medved, A. Horak, and V. Kovář, "Bilingual logical analysis of natural language sentences," 2016.

[25] H. Sun, W. W. Cohen, and R. Salakhutdinov, "Iterative hierarchical attention for answering complex questions over long documents," 2021.

[26] R. Jia, M. Lewis, and L. Zettlemoyer, "Question answering infused pre-training of general-purpose contextualized representations," 2022.

[27] W. Budiharto, V. Andreas, and A. A. S. Gunawan, "Deep learning-based question answering system for intelligent humanoid robot." vol. 7, 2020. [Online]. Available: http://dblp.uni-trier.de/db/journals/jbd/jbd7.htmlBudihartoAG20

[28] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, "Bidirectional attention flow for machine comprehension," 2018.

[29] Z. Lin, M. Feng, C. N. dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding." *CoRR*, vol. abs/1703.03130, 2017. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1703.htmlLinFSYXZB17

[30] J. Wieting, G. Neubig, and T. Berg-Kirkpatrick, "A bilingual generative transformer for semantic sentence embedding." *CoRR*, vol. abs/1911.03895, 2019. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1911.htmlabs-1911-03895

[31] M. N. Moghadasi and Y. Zhuang, "Sent2vec: A new sentence embedding representation with sentimental semantic," pp. 4672–4680, 2020.

[32] X. Huang, J. Zhang, D. Li, and P. Li, "Knowledge graph embedding based question answering," *WSDM '19: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pp. 105–113, 01 2019.

[33] A. Diallo, M. Zopf, and J. Fürnkranz, "Learning analogy-preserving sentence embeddings for answer selection." vol. abs/1910.05315, 2019. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1910.htmlabs-1910-05315

[34] A. Saxena, A. Tripathi, and P. Talukdar, "Improving multi-hop question answering over knowledge graphs using knowledge base embeddings," pp. 4498–4507, 01 2020.

[35] D. Dimitriadis and G. Tsoumakas, "Word embeddings and external resources for answer processing in biomedical factoid question answering," *Journal of Biomedical Informatics*, vol. 92, p. 103118, 02 2019.

[36] K. Cios, W. Pedrycz, R. Swiniarski, and L. Kurgan, "Data mining: A knowledge discovery approach," *Data Mining: A Knowledge Discovery Approach*, 01 2007.

[37] S. Lalithsena, S. Perera, P. Kapanipathi, and A. Sheth, "Domain-specific hierarchical subgraph extraction: A recommendation use case," pp. 666–675, 12 2017.

[38] Z. Li, H. Wang, and W. Zhang, "Translational relation embeddings for multi-hop knowledge base question answering," *Journal of Web Semantics*, vol. 74, p. 100723, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570826822000154

[39] S. Buosi, M. Timilsina, A. Janik, L. Costabello, M. Torrente, M. Provencio, D. Fey, and V. Novácek, "Machine learning estimated probability of relapse in early-stage non-small-cell lung cancer patients with aneuploidy imputation scores and knowledge graph embeddings," *Expert Systems with Applications*, vol. 235, p. 121127, 08 2023.

[40] Q. Guo, X. Wang, Z. Zhu, P. Liu, and L. Xu, "A knowledge inference model for question answering on an incomplete knowledge graph," *Applied Intelligence*, vol. 53, no. 7, p. 7634–7646, jul 2022. [Online]. Available: https://doi.org/10.1007/s10489-022-03927-0

[41] A. Miller, A. Fisch, J. Dodge, A.-H. Karimi, A. Bordes, and J. Weston, “Key-value memory networks for directly reading documents,” pp. 1400–1409, 01 2016.
[42] H. Sun, B. Dhingra, M. Zaheer, K. Rivard, R. Salakhutdinov, and W. Cohen, “Open domain question answering using early fusion of knowledge bases and text,” pp. 4231–4242, 01 2018.
[43] H. Sun, T. Bedrax Weiss, and W. Cohen, “Pullnet: Open domain question answering with iterative retrieval on knowledge bases and text,” 04 2019.
[44] J. Cai, Z. Zhang, F. Wu, and J. Wang, “Deep cognitive reasoning network for multi-hop question answering over knowledge graphs,” pp. 219–229, 01 2021.