IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 15, 2022, accepted April 28, 2022, date of publication May 9, 2022, date of current version May 16, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3173355

Tree-KGQA: An Unsupervised Approach for Question Answering Over Knowledge Graphs

MD RASHAD AL HASAN RONY[1,3], DEBANJAN CHAUDHURI[1], RICARDO USBECK[2], AND JENS LEHMANN[1,3]

[1]Smart Data Analytics Research Group, University of Bonn, 53115 Bonn, Germany
[2]Semantic Systems Group, University of Hamburg, 22527 Hamburg, Germany
[3]Fraunhofer IAIS Dresden, 01069 Dresden, Germany

Corresponding author: Md Rashad Al Hasan Rony (rashad.rony@iais.fraunhofer.de)

This work was supported in part by the SPEAKER under Grant BMWi FKZ 01MK20011A, in part by the JOSEPH (Fraunhofer Zukunftsstiftung), in part by the OpenGPT-X under Grant BMWK FKZ 68GX21007A, in part by the excellence clusters ML2R under Grant BmBF FKZ 01 15 18038 A/B/C, in part by the ScaDS.AI under Grant IS18026A-F, in part by the TAILOR under Grant EU GA 952215, and in part by the Federal Ministry for Economic Affairs and Energy of Germany in the Project CoyPu under Project 01MK21007G.

ABSTRACT Most Knowledge Graph-based Question Answering (KGQA) systems rely on training data to reach their optimal performance. However, acquiring training data for supervised systems is both time-consuming and resource-intensive. To address this, in this paper, we propose **Tree-KGQA**, an unsupervised KGQA system leveraging pre-trained language models and tree-based algorithms. Entity and relation linking are essential components of any KGQA system. We employ several pre-trained language models in the entity linking task to recognize the entities mentioned in the question and obtain the contextual representation for indexing. Furthermore, for relation linking we incorporate a pre-trained language model previously trained for language inference task. Finally, we introduce a novel algorithm for extracting the answer entities from a KG, where we construct a forest of interpretations and introduce tree-walking and tree disambiguation techniques. Our algorithm uses the linked relation and predicts the tree branches that eventually lead to the potential answer entities. The proposed method achieves 4.5% and 7.1% gains in F1 score in entity linking tasks on LC-QuAD 2.0 and LC-QuAD 2.0 (KBpearl) datasets, respectively, and a 5.4% increase in the relation linking task on LC-QuAD 2.0 (KBpearl). The comprehensive evaluations demonstrate that our unsupervised KGQA approach outperforms other supervised state-of-the-art methods on the WebQSP-WD test set (1.4% increase in F1 score) - without training on the target dataset.

INDEX TERMS Knowledge based systems, information retrieval, question answering, entity linking, relation linking, indexing, pre-trained language models.

I. INTRODUCTION

A knowledge graph can be viewed as an abstraction of the real world that describes real-world entities and their relationships. Knowledge graphs are widely used as a source of structured data for KG-based question answering, dialogue systems, retrieval systems. Since the advent of large-scale knowledge graphs (KG) such as DBpedia [1], Freebase [2], and Wikidata [3], KG-based systems have evolved significantly. Given a natural language question, the task of a KG-based question answering (KGQA) system is to retrieve the correct answer from the knowledge graph. Entity and relation linking are the primary sub-tasks of KGQA. These sub-tasks include determining the *surface form* (mentions in the question) of the entity and relation in the question and subsequently mapping them to the respective entity and relation in the knowledge graph. The linked entity and relation are then utilized to obtain the answer entity in the final step [4].

KGQA on both simple and complex questions is a well-researched topic [5]–[7]. For training, supervised systems depend heavily on knowledge graph-based question answering datasets. Reaching peak performance often requires a significant amount of training data [8], [9]. Since both data collection and training processes are time consuming and cost-intensive, this is a bottleneck in developing dataset-independent KGQA systems. Furthermore,

The associate editor coordinating the review of this manuscript and approving it for publication was Gianmaria Silvello.

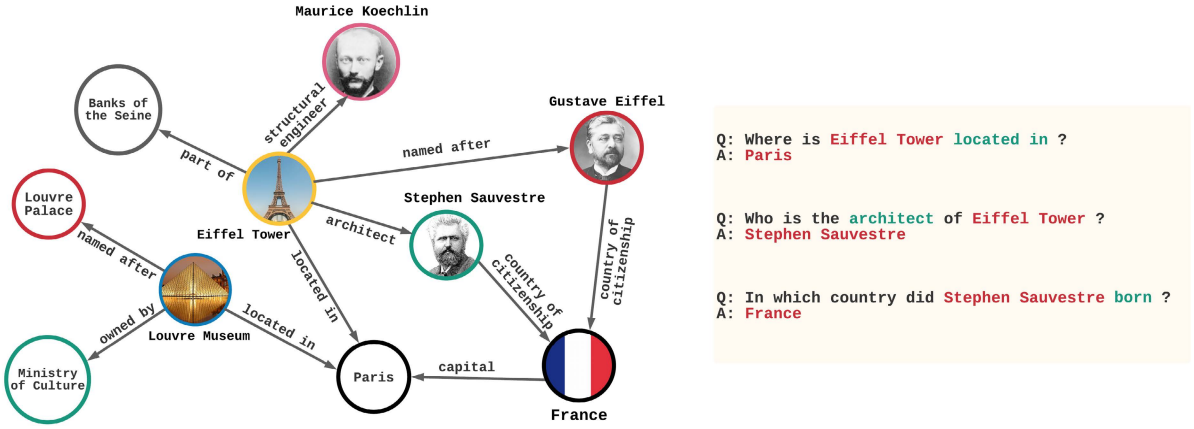

FIGURE 1. An illustration of question answering over a knowledge graph. Figure a) depicts a sub-graph of the Wikidata KG, where Figure b) demonstrates sample question-answer pairs based on the example sub-graph. In the sample question-answer pairs, the surface form of the entities and relations are in red and green, respectively.

supervised systems are often vulnerable to brittleness [10]. Since they aim to capture the underlying dynamics in the training data, they frequently fail to generalize well when tested on previously unseen data. The KGQA task is depicted in Figure 1, where the circular nodes indicate entities and the connecting directed lines represent the relationship between two KG entities.

To alleviate the time and effort necessary to develop a question answering (**QA**) system, researchers recently explored unsupervised and few-shot question answering techniques [11], [12]. Effective unsupervised **KGQA** is still a challenging research problem. Unsupervised KGQA is particularly hard because, **firstly**, large-scale knowledge graphs such as Wikidata [3] contain more than 80 million entities and a few thousand relations. Linking the entity and relation mentioned in the question to the corresponding large-scale KG entity and relation is thus a challenging task. **Secondly**, it is a standard practice to execute a query (e.g., using SPARQL) over the KG to extract answer entities [4], [13]. Query construction for this purpose adds an additional layer of difficulty.

Addressing the issues mentioned above, we propose a simple yet effective unsupervised KGQA method leveraging pre-trained language models. The primary motivation of this research is to develop a dataset-independent KGQA system, which can answer natural questions from various datasets without additional training or fine-tuning. We adopt powerful off-the-shelf language models pre-trained on named entity recognition (NER) and natural language inference tasks for the KGQA sub-tasks [14], [15]. Specifically, we split the KGQA task into three sub-tasks: entity linking, relation linking, and answer entity extraction. **Firstly**, we employ a BERT-based [14] pre-trained NER model to detect the surface form of the entity. Additionally, we pre-process and index the contextualized representation of the entities into a dense space for effective and fast candidate entity generation during the inference. The index is utilized to generate a set of candidate entities, which are then disambiguated to obtain the final predicted entity (details in Section III-A). **Secondly**, by combining the 1-hop connected relations of the entities linked in the previous step, a set of candidate relations for relation linking is created. A pre-trained BART model [15] is then applied to the candidate relations to obtain the most probable relation in a zero-shot manner (details in Section III-B). **Finally**, we construct a set of k-level trees from the k-hop sub-graphs of the linked entities. Then, *tree-walking* and *tree-disambiguation* techniques are employed to extract answer entities from the constructed trees (details in Section III-C).

To assess the performance of our proposed approaches, we conduct experiments on four publicly available benchmarks: LC-QuAD 2.0 [16], LC-QuAD 2.0 (KBpearl) [17], QALD-7-Wiki [18], and WebQSP-WD [19]. The empirical study confirms that our proposed system achieves a significant improvement in entity and relation linking sub-tasks. In the entity linking task, we notice an absolute increase of 4.5% on the LC-QuAD 2.0, 7.1% on the LC-QuAD 2.0 (KBpearl), and 0.1% on the QALD-7-Wiki in F1 score. The improvement in relation linking is 5.4% on the LC-QuAD 2.0 (KBpearl) in F1 score. Despite the simplicity, our proposed Tree-KGQA achieves an absolute increase of 1.4% in the F1 score over the state-of-the-art methods without training on WebQSP-WD test set. To encourage further research on unsupervised KGQA, we have made our code open source.[1] We anticipate that our findings will lay the groundwork for further study on unsupervised KGQA. The contributions of this paper can be summarized as follows:

- We propose an unsupervised entity linking method that achieves state-of-the-art (SOTA) results on LC-QuAD 2.0, LC-QuAD 2.0 (KBpearl), and QALD-7-Wiki datasets.
- We introduce a zero-shot relation linking mechanism that achieves SOTA results on the LC-QuAD 2.0 (KBpearl).

[1] https://github.com/rashad101/Tree-KGQA

- We introduce a novel *tree-walking* and *tree-disambiguation* techniques for extracting answer entities. In particular, we propose a modular and unsupervised KGQA system that does not require any training and can be applied to any Wikidata-based KGQA dataset. Finally, we establish a new baseline for KGQA on the LC-QuAD 2.0 KBpearl dataset.

Rest of the part of this paper is organised as follows. In Section II, we review the previous research efforts on various methods for entity linking, relation linking, and answer extraction. In Section III, we describe our proposed unsupervised KGQA approach which includes, unsupervised entity linking, unsupervised relation linking, and tree-walking based answer extraction method. In Section IV, we describe the experiments and results. A comprehensive analysis of the proposed system and its components is provided in Section V. Finally, in Section VI, we summarize the key findings and identify future study areas.

II. RELATED WORK

Our research mainly focuses on leveraging pre-trained language models for question answering over knowledge graphs (KGQA). The KGQA task is often divided into three atomic sub-tasks namely, entity linking, relation linking and answer entity extraction.

A. ENTITY LINKING

Previous works on entity linking primarily focused on detecting entity mentions in the question and then linking these mentions to the correct entity in the knowledge using entity labels as well as other features such as entity type information [20], [21]. Several studies in a separate line of research focused on training entity mention detection and entity disambiguation together to perform entity linking [8], [9]. However, in order to train these systems, it is necessary to have datasets with annotated entity mention boundaries. Recently, natural language processing has reached a new height of success with the emergence of Transformer-based [22] pre-trained language models [14], [15]. In the context of question answering, pre-trained language models have been widely studied for the entity linking task [9], [23].

B. RELATION LINKING

Relation linking is another challenging task in KGQA since it requires complex language inference capabilities. Both supervised and distantly supervised approaches have been explored for the relation linking task [21], [24]. In a different research, systems use already linked entities from the preceding step to perform relation linking, utilizing the structural information of the knowledge graph [25]. Unseen relation linking has also been studied recently, where the model needs to predict relations which are not seen during the training step [26]. In a similar line of research [27], [28], models jointly use knowledge graph embedding for entity linking, where the linked relation information is used additionally to perform disambiguation among the candidate entities. In a disparate research, a zero-shot methodology has also been used to investigate relation linking [29].

C. ANSWER ENTITY EXTRACTION

The two most prevalent methodologies for the answer entity extraction sub-task are semantic parsing-based and retrieval-based methods. Semantic parsing-based methods transform the natural question into a logical form which is then utilized to fetch the answer entities from the target KG [30], [31]. On the contrary, retrieval-based methods use the entity and relation extracted from the natural question to obtain the answer entities from the KG [32], [33]. In a different line of research, a graph neural network-based method for KGQA has been proposed by Sorokin and Gurevych [19], while other approaches fetch candidate SPARQL queries using the entities and predicted relations and re-rank them using neural network-based methods [4], [34]. More recently, a message-passing based system for the KGQA task has been developed, where a confidence score is propagated throughout the knowledge graph, computed by input question parsing and matching [5].

Several studies proposed pre-trained language model-based zero-shot **QA** systems [35], [36]. In contrast to the previous works, our proposed system focuses on solving the **KGQA** problem in an unsupervised way, utilizing pre-trained language models without fine-tuning for entity and relation linking, and tree-based techniques for answer entity extraction.

III. APPROACH: TREE-KGQA

In this section, first, we define the knowledge graph and knowledge tree. Following that, we discuss each component of our proposed Tree-KGQA system in depth.

Definition 1 (Knowledge Graph): A *knowledge graph* $\mathcal{G}$, is a labelled and directed multi-graph consisting of a set of entities $\mathcal{E}$ as nodes and a set of relations $\mathcal{R}$ as edges between them. A k-hop sub-graph $\mathcal{G}_i^k$ associated to a node $E_i \in \mathcal{E}$, denotes the set of all the connected nodes and edges within the radius-k distance from node E_i.

Definition 2 (Knowledge Tree): A *knowledge tree* with k-levels $\mathcal{T}_i^k$, associated to an entity E_i, is a labelled and directed tree; consisting of nodes Ω and branches Ψ, where $\{\Omega, \Psi\} \in \mathcal{G}_i^k$. A Forest $\mathcal{F}$, is denoted as the set of knowledge trees; $\mathcal{F} = \{\mathcal{T}_1^k, \mathcal{T}_2^k, .., \mathcal{T}_p^k\}$ where p is the number of trees in the forest.

Given a natural language question $\mathcal{Q}$, our proposed system aims to predict a set of answer entities $\mathcal{E}^a \subseteq \mathcal{E}$ that answers the question. Table 1 provides an overview of the notations of the concepts covered in this research.

A. ENTITY LINKING

The entity linking task entails a) mention detection – spotting the *surface form* of the entity that appears in the question and b) mapping the detected mention to the corresponding

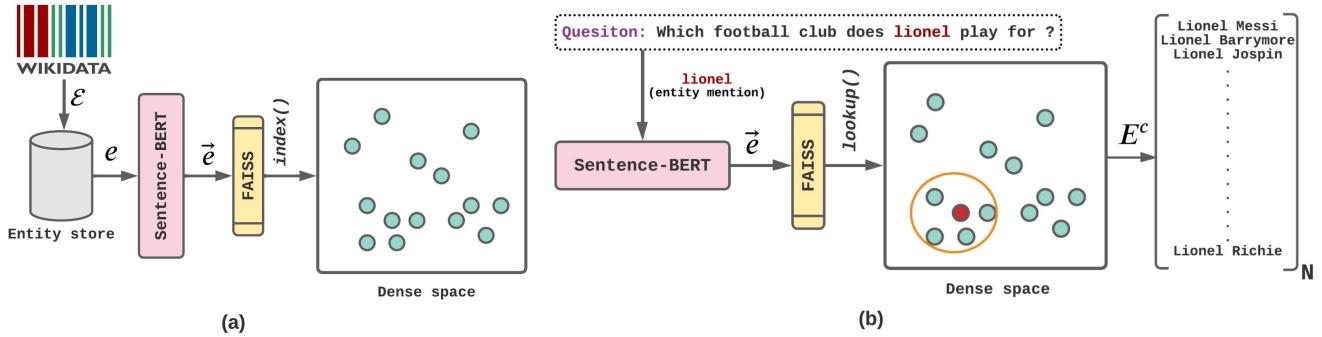

FIGURE 2. Figure (a) illustrates how the entity labels are encoded with Sentence-BERT and then indexed into a dense space using FAISS. The Indexing algorithm *IndexFlatIP* of FAISS, clusters similar entities together into the dense space. Figure b) demonstrates the candidate entity generation procedure given a detected entity mention. Sentence-BERT is used to obtain the vector representation of the entity mention *lionel*. The encoded vector is then passed to the FAISS module that performs a lookup into the dense space and generates *N* candidate entities that are similar to the provided entity span, *lionel*. The red circle represents the given entity mention in the dense space, where the other circles inside the larger orange circle indicate similar entities around it.

TABLE 1. Notation of the concepts used in tree-KGQA.

Notation	Concept
e	Label of the entity E
$\vec{e}$	Embedding representation of the entity label e
m_i	i-th entity mention in the question
E_i^m	Linked entity for the entity mention m_i
E_i^c	A set of candidate entities with labels similar m_i
$\mathcal{E}^L$	A set of linked entities corresponding to the entity mentions in $\mathcal{Q}$
$\mathcal{R}^L$	Linked relation for a given question
h_i	A set of relations connected to 1-hop of entity E_i

knowledge graph entity. The steps involved in entity linking are described below.

1) MENTION DETECTION

To detect the entity mentions in the question, we employ a BERT-large [14] model pre-trained for the named entity recognition task.

$$W_m = f(\mathcal{Q}) \tag{1}$$

The function $f(\cdot)$ in Equation 1, is a pre-trained BERT-large model that takes a question $\mathcal{Q}$ as input and predicts a set of named entity word tokens, W_m as the output. For instance, consider the question, `Which football club does lionel play for?`. The system detects `lionel` as the entity mention in this step using Equation 1. In the following steps, the detected entity mention is mapped or in other words linked to the corresponding knowledge graph entity.

2) ENTITY MAPPING

We first index all the entity labels from a target KG into a dense space as a pre-processing step of entity mapping. During inference, the system generates candidate entities from the dense space for each detected entity mention from the previous step. To obtain the final linked entity from the set of candidate entities, an additional entity disambiguation step is performed in the cases where the same entity label appears more than once. The entity mapping technique is explained in detail below.

a: ENTITY INDEXING

In this step, **firstly**, we extract all the entities from the target KG, in our case Wikidata, and store it in an *Entity store* (see Figure 2a). The *Entity store* contains all the Wikidata entity labels (e.g., *Lionel Messi*) and their Wikidata ID (e.g., *Q615*). **Secondly**, we encode all the knowledge graph entity labels using Sentence-BERT [37]. Sentence-BERT captures the overall meaning of the entity label better since entity labels frequently contain multiple words in them. We obtain a vector of dimension 1×768 for each entity label from Sentence-BERT. **Finally**, the encoded vector representations of the KG entities are indexed into a dense space using FAISS [38]. During the inference, the system utilises a hierarchical indexing algorithm *IndexHNSWFlat* from FAISS, which enables the system to generate candidate entities (see Figure 2b) in an optimized way [8], [9]. Given an entity span, the hierarchical indexing algorithm generates N candidate entities from the dense space based on k-nearest neighbors (KNN) approximate search.

For each detected entity span $m_i \in W_m$, the system performs entity linking separately. The system generates a set of $N = 10$ candidate entities $E_i^c = \{E_1, E_2, \ldots, E_N\}$ for each entity mention $m_i \in W_m$, using FAISS (Figure 2b). Each generated candidate entity has an indexing score (from the FAISS approximate search) indicating how similar they are to the *entity mention* in the dense space. The candidate entity with the highest indexing score is then considered as the linked entity. Henceforth, a disambiguation step between the generated entity candidates is not required if all the candidate entity labels appeared once in the set.

b: ENTITY DISAMBIGUATION

The system performs entity disambiguation if an entity label appears multiple times in the candidate entity set. In that

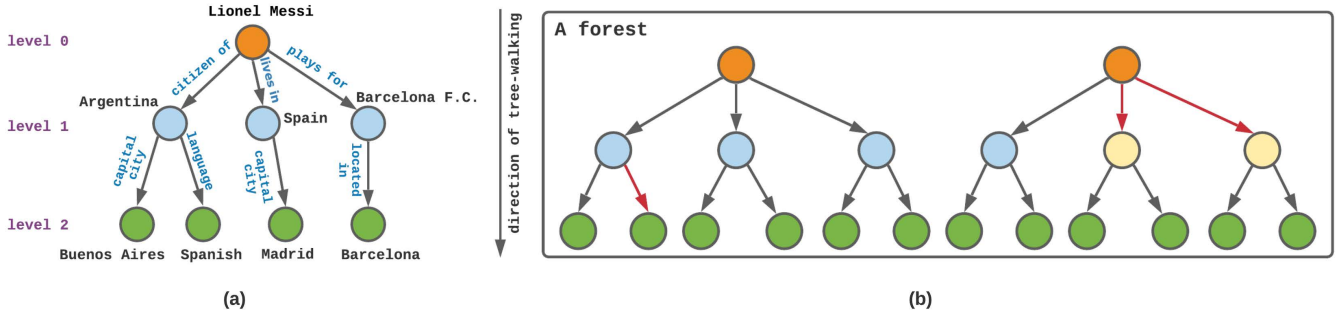

FIGURE 3. Figure a) depicts a *k*-level tree (with *k*=2). Since a tree has many nodes and branches (edges), we present a toy example. Figure b) shows a forest consists of a set of trees constructed from the sub-graph of the linked entities. For the demonstration purpose, we show a forest consists of two trees. The red branches show the position of predicted relation in different trees. The green nodes represent the leaf nodes at level-*k*, where the blue nodes refer to the intermediary nodes between the root and leaf nodes. Furthermore, the yellow nodes represent the predicted answer entity nodes connected by the red branches.

case, it firstly predicts a temporary relation R_t using Algorithm 5. Although we develop Algorithm 5 to perform relation linking (details in Section III-B), in this section we utilize Algorithm 5 to obtain $\mathcal{R}_t$. The question $\mathcal{Q}$, and a set of all the 1-hop connected relations of the candidate entities are used as input to the Algorithm 5. As the output, Algorithm 5 predicts a relation which we denote as $\mathcal{R}_t$ in this section. The system selects an entity with the highest similarity score from E_i^c as linked entity E_i^m, which is connected to the predicted relation $\mathcal{R}_t$ at a distance of 1-hop in the KG. For instance, for the question `Which company's CEO is Tim Cook?`, the predicted entity mention is `Tim Cook`. The entity label `Tim Cook` appears multiple times in the set of generated candidate entities; hence, entity disambiguation is required. By utilizing Algorithm 5, *CEO* is obtained as $\mathcal{R}_t$. In the generate candidate entity set, *Tim Cook (Q265852)* has the relation *CEO* in its 1-hop connected relations. Where the other candidate entities with the same entity label (e.g., *Tim Cook (Q7803347)* an Australian rules footballer, *Tim Cook (Q1404825)* an American ice hockey player) do not have the relation *CEO* in their 1-hop connections. Consequently, *Tim Cook (Q265852)*, an American business executive, gets predicted as the final linked entity. In the cases where there exist multiple candidate entities with the same label, and $\mathcal{R}_t$ in their 1-hop, the entity with the highest indexing score that contains $\mathcal{R}_t$ in its 1-hop is selected as the linked entity.

Finally, after repeating the whole entity mapping process for each entity mention, the system produces the final set of linked entities, $\mathcal{E}^L$ as follows:

$$\mathcal{E}^L = \bigcup_{m_i \in W_m} E_i^m \tag{2}$$

For the running example question, the entity mention `lionel` gets linked to the Wikidata entity, *Lionel Messi (Q615)*.

B. ZERO-SHOT RELATION LINKING

We model the relation linking problem as a classification task, where the system aims to link the given natural language question to one of the KG relations based on label information. In our proposed approach, we firstly generate a set of candidate relations $\mathcal{R}^c$ from all the 1-hop connected relations of the already linked entities $\mathcal{E}^L$ as follows:

$$\mathcal{R}^c = \bigcup_{E_i \in \mathcal{E}^L} h_i \tag{3}$$

where h_i denotes the set of 1-hop connected relations of the entity E_i. For the running example question and linked entity *Lionel Messi*, the set of candidate relations $\mathcal{R}^c$ is {*citizen of*, *lives in*, *plays for*}(see Figure 3a). Furthermore, we mask all the detected entity mentions in the question with a generic token `<ENT>`, to obtain a *masked question* representation denoted by $\hat{Q}$, `Which football club does <ENT> play for?`. We mask the entity mentions in the question to reduce noises in the relation classification task. In Algorithm 5, the function *maskEnt*(.) masks the entities in the question. The system then performs zero-shot relation label classification, leveraging a pre-trained language model called BART [15], which was pre-trained for the natural language inference (NLI) task. In Equation 4, function $\mathcal{Z}(\cdot)$ is a BART-large model [15] that computes the probability of being the correct relation label given the modified question ($\hat{Q}$) and a set of candidate relation labels (labels of relations in $\mathcal{R}^c$).

$$p(r_i|\hat{\mathcal{Q}}, \mathcal{R}^c) \leftarrow \mathcal{Z}(\hat{\mathcal{Q}}, \mathcal{R}^c) \tag{4}$$

Here, $r_i \in \mathcal{R}^c$ is a candidate relation. Finally, we obtain the predicted relation $\mathcal{R}^L$ as follows:

$$\mathcal{R}^L = \underset{r_i \in \mathcal{R}^c}{\text{argmax}}\ p(r_i) \tag{5}$$

From Equation 5, the system obtains *plays for* as the predicted and linked relation $\mathcal{R}^L$. Algorithm 5 summarizes the relation linking task described in this section.

C. ANSWER ENTITY EXTRACTION

To extract the answer entities from the knowledge graph, firstly, we build a forest utilizing the sub-graph information associated to the linked entities (obtained from Section III-A). Then, we perform tree-walking over all the trees within the constructed forest, using the relation predicted in Section III-B. Finally, we obtain the answer entities

Algorithm 1: Relation Linking

Input: A question $\mathcal{Q}$, a set of candidate relations $\mathcal{R}^{cand}$

Output: A relation $\mathcal{R}^p$

$\mathcal{R}^p \leftarrow \emptyset$
$\hat{\mathcal{Q}} \leftarrow maskEnt(\mathcal{Q})$
$p(r_i|\hat{\mathcal{Q}}, \mathcal{R}^{cand}) \leftarrow \mathcal{Z}(\hat{\mathcal{Q}}, \mathcal{R}^{cand})$
$\mathcal{R}^p \leftarrow \text{argmax}\, p(r_i)$, where $r_i \in \mathcal{R}^{cand}$
return $\mathcal{R}^p$

from the tree, based on the tree-disambiguation technique following Algorithm 20.

1) BUILDING A FOREST

In order to build a forest, first we construct a set of knowledge-trees. For each linked entity $E_i \in \mathcal{E}^L$, we generate a k-level tree $\mathcal{T}_i^k$ constructed from the k-hop sub-graph associated to E_i as follows:

$$\mathcal{T}_i^k \leftarrow buildTree(E_i, \mathcal{G}_i^k) \tag{6}$$

The linked entity is designated as the tree's root node (in orange color) at level 0 (Figure 3a). In this case, *Lionel Messi* is the root node of a tree. The other nodes and edges in the k-hop sub-graph of the linked entity are connected to the tree's root node at the same stage as they are in the sub-graph $\mathcal{G}_i^k$. The function *buildTree*$(\cdot)$ in Algorithm 20, performs the tree-construction operation. A set of generated k-level trees are denoted as a forest $\mathcal{F}$ (as specified by the definition 2). In cases where no entities are linked, as predicted answer entities the system returns an empty set. For the running example question, the system constructs a forest with one tree for the linked entity *Lionel Messi (Q615)*.

Each branch of the tree represents a relation between the parent and the child entity node. For instance in Figure 3a, a branch "*capital city*" connects a parent entity node, "*Spain*" and a child entity node, "*Madrid*" (*Spain* $\xrightarrow{capital\ city}$ *Madrid*). Each node in a tree preserves a state variable $\mathcal{V}$, which holds a set of values $\{\mathcal{S}_r, \mathcal{K}, \text{and } \mathcal{R}_{max}\}$. Where $\mathcal{K}$ denotes the tree level, $\mathcal{R}_{max}$ the relation for which the node obtained the maximum score, and S_r the maximum similarity score for the relation $\mathcal{R}_{max}$. During the answer entity extraction process, the values of the state variable aid in the tree-disambiguation process. At this stage, all state variables are initialized with null value.

2) TREE-WALKING

In this step, the predicted relation $\mathcal{R}^L$ performs tree-walking across all the trees in the forest, starting from the root node till the nodes at level-k of each tree. During the walk, for each tree $\mathcal{T}_i^k \in \mathcal{F}$ the system computes embedding-based cosine similarity between the predicted relation $\mathcal{R}^L$ and all the 1-hop connected branches h_i of each node $E_i \in \mathcal{T}_i^k$. At each step of the walk, the system updates the node state

Algorithm 2: Answer Entity Extraction

Input: A forest $\mathcal{F}$, predicted relation $\mathcal{R}^{pred}$ and hops k

Output: A set of entities $\mathcal{E}^a$

$\mathcal{E}^a, \mathcal{S}_r^{max}, \mathcal{R}_{max} \leftarrow \emptyset$
for $\mathcal{T}_i^k \in \mathcal{F}$ **do**
  for $E_i \in \mathcal{T}_i^k$ **do**
    for $r_i \in h_i$ **do**
      $S_c \leftarrow cosine(emb(\mathcal{R}^{pred}), emb(r_i))$
      if $S_c > \mathcal{S}_r^{max}$ **then**
        $\mathcal{S}_r^{max} \leftarrow S_c$; $\mathcal{R}_{max} \leftarrow r_i$
      if $S_c > E_i[\mathcal{S}_r]$ **then**
        $E_i[\mathcal{V}] \leftarrow updateState(S_c, r_i)$
$h_{low} \leftarrow k$
for $\mathcal{T}_i^k \in \mathcal{F}$ **do**
  for $E_i \in \mathcal{T}_i^k$ **do**
    if $E_i[\mathcal{S}_r] = \mathcal{S}_r^{max}$ **then**
      if $E_i[\mathcal{K}] < h_{low}$ **then**
        $h_{low} \leftarrow E_i[\mathcal{K}]$
        $\mathcal{E}^a \leftarrow connE(E_i[\mathcal{R}_{max}])$
      else if $E_i[\mathcal{K}] = h_{low}$ **then**
        $E^a \leftarrow connE(E_i[\mathcal{R}_{max}])$
        $\mathcal{E}^a \leftarrow \mathcal{E}^a \cup E^a$
return $\mathcal{E}^a$

(value of $\mathcal{S}_r$ and $\mathcal{R}_{max}$) with the similarity scores of the connected 1-hop relations. The values of a node state only get updated when a higher value than the existing $\mathcal{S}_r$ of that node is obtained for any connected relation (or branch). The function *updateState*$(\cdot)$ in Algorithm 20, updates the node state values with the values passed in as parameters. We employ QuatE [39], a knowledge graph embedding model trained on Wikidata, to compute the similarities between two relations in order to consider KG structural information during the process. In Algorithm 20, the function *emb*$(\cdot)$ takes a relation as input and returns the knowledge graph embedding of the relation from QuatE. Finally, the system selects all entities connected to the node with the highest $\mathcal{S}_r$ value, by $\mathcal{R}_{max}$, as answer entities $\mathcal{E}^a$.

3) TREE-DISAMBIGUATION

We introduce a tree disambiguation technique for extracting the answer entities from the forest. In this technique, the system chooses the tree in which the node with the highest score ($\mathcal{S}_r$) resides. If multiple trees have a node with the same maximum score in their node state, the tree with the highest scoring node at the lowest level (lower value of k) is chosen (Figure 3). Moreover, in rare cases (less than 1% in the WebQSP-WD dataset), when several trees have nodes

TABLE 2. Dataset statistics.

	LC-QuAD 2.0	LC-QuAD 2.0 (KBpearl)	WebQSP-WD	QALD-7-Wiki
Split (train/test)	24,180 / 6,064	24,180 / 1,942	2,880 / 1,033	100 / 50
Number of entities per question	1.47	1.48	1.47	1.08
% of question with no entity	0.02%	0.41%	0.0%	8.0%
Number of words per question	10.61	14.10	6.72	7.62

TABLE 3. Performance of the entity linking component on LC-QuAD 2.0.

Systems	Precision	Recall	F1
OpenTapioca [51]	0.237	0.411	0.301
Falcon 2.0 [49]	0.395	0.268	0.320
VCG [48]	0.403	0.498	0.445
PNEL [41]	0.688	0.516	0.589
Tree-KGQA	**0.720**	**0.566**	**0.634**

with the highest scores at the same level (k), the system selects all the trees with such cases and extracts all the answer entities connected to the $\mathcal{R}_{max}$. In Algorithm 20, line no. 10-19 demonstrate the tree-disambiguation process. Finally, *Barcelona F.C.* is chosen as the answer entity from the tree since the predicted relation *plays for* connects *Barcelona F.C.* to the linked entity *Lionel Messi*. The function *connE(·)* in Algorithm 20 selects all the answer entities connected to the entity E_i by the relation $\mathcal{R}_{max}$.

IV. EXPERIMENTS AND RESULTS

A. DATA

We chose Wikidata [3] (based on May 2019 English Wikipedia release) as the knowledge graph to gauge our proposed method since Wikidata is frequently used as a knowledge base for KGQA datasets. We evaluate our proposed method on four publicly available knowledge graph based question answering datasets:

- *LC-QuAD 2.0* [16]: A large-scale dataset on Wikidata Knowledge Graph which was generated semi-automatically and consists of complex questions and their paraphrases.
- *LC-QuAD 2.0 (KBpearl)* [17]: A subset of the LC-Quad 2.0 dataset, selected by [17]. The KBpearl split of the LC-QuAD 2.0 data comprises of 1,942 test questions.
- *QALD-7-Wiki* [18]: A manually constructed small, complex question answering dataset, developed for Task 4 (''English question answering over Wikidata'') of the QALD-7 challenge [18].
- *WebQSP-WD* [19]: A Wikidata-based question answering dataset constructed from the original Freebase-based WebQSP dataset [40].

It is noteworthy that the system can be extended to different knowledge graphs with low effort (discussed in Section V-D). Table 2 lists the statistics of the datasets used in this research.

B. EXPERIMENTAL SETUP

We run our experiments on a system with 28 CPU cores, 12GB of GPU memory, and 256GB of RAM. A pre-trained BERT-large [14] model with 340M parameters and BART-large model [15] with 406M parameters are used in this paper. We use macro-F1 score to evaluate the components of our system similar to other baseline models [17], [41].

C. BASELINES

We select a wide range of baseline models related to KGQA sub-tasks. The baseline models used in this paper are summarised below:

DBpedia Spotlight: An open-source tool and a popular baseline for the entity linking task in TAC-KBP [42], [43].

TagMe: An entity linking tool that index Wikipedia pages and performs annotation on a given text [44].

QKBfly: An information extraction (IE) tool based on ClausIE [45], which predicts a triple from the KG, on-the-fly [13].

EARL: Jointly performs entity and relation linking from the knowledge graph, by solving a *Traveling Salesman Problem* on the candidate nodes [46].

ReMatch: A part-of-speech and dependency parsing based relation linking tool for question answering [47].

Falcon: A tool that jointly performs entity and relation linking leveraging the concept of morphology and knowledge graph information [21].

VCG: A jointly optimized model for entity mention detection and disambiguation using contextual information [48].

KBPearl-NN: A neural network based end-to-end system that performs joint entity and relation linking [17].

PNEL: A pointer network based entity linking system [41].

Falcon 2.0: A morphology based entity and relation linking system [49].

STAGG: A semantic parsing approach for question answering over knowledge graph [50]. A re-implementation of STAGG from Sorokin and Gurevych [19] to facilitate the KGQA task, is used as a baseline in this work.

GGNN: Uses a complex semantic parser for performing question answering over knowledge bases [19].

The baseline scores in this paper are all reported from [17], [19], [41].

TABLE 4. Performance of the entity linking component on the LC-QuAD 2.0 (KBpearl).

Systems	Precision	Recall	F1
EARL [46]	0.403	0.498	0.445
QKBfly [13]	0.518	0.479	0.498
TagMe [44]	0.352	**0.864**	0.500
Falcon [21]	0.533	0.598	0.564
KBPearl-NN [17]	0.561	0.647	0.601
Spotlight [43]	0.585	0.657	0.619
PNEL [41]	**0.803**	0.517	0.629
Tree-KGQA	0.737	0.666	**0.700**

TABLE 5. Performance of the entity linking component on the QALD-7-Wiki.

Systems	Precision	Recall	F1
TagMe	0.349	0.661	0.457
EARL	0.516	0.460	0.486
QKBfly	0.592	0.510	0.548
Spotlight	0.619	0.634	0.626
Falcon	0.708	0.651	0.678
KBPearl-NN	0.647	0.715	0.679
Tree-KGQA	**0.714**	**0.648**	**0.680**

TABLE 6. Performance of the relation linking component on the LC-QuAD 2.0 (KBpearl).

System	Precision	Recall	F1
EARL [46]	0.259	0.251	0.255
ReMatch [47]	0.201	0.214	0.207
Falcon [21]	0.302	0.325	0.313
KBPearl-NN [17]	0.358	**0.479**	0.410
Tree-KGQA	**0.554**	0.400	**0.464**

D. RESULTS

1) ENTITY LINKING

Table 3 shows the entity linking performance of the baseline models and our approach on LC-QuAD 2.0. All the results reported in this section are on the [0, 1] scale and test split of the datasets. From the results in Table 3, it is evident that our system achieves higher precision, recall and F1 scores as compared to the other baseline models.

We notice a substantial improvement (increment of 7.1%) on LC-QuAD 2.0 KBpearl in entity linking, see Table 4. We observed the majority of baseline systems have either low accuracy or recall scores. This is mostly due to the fact that the dataset is complex and often comprises many things. Our proposed entity linking mechanism achieved a balanced precision and recall score, resulting in a superior F1 score. The entity linking result on the small yet challenging dataset (QALD-7-Wiki) is reported in Table 5. Improved results across several datasets verify the effectiveness of our unsupervised entity linking approach.

2) RELATION LINKING

The relation linking performance of the baseline models and our proposed approach on LC-QuAD 2.0 (KBpearl) is reported in Table 6. The baseline scores are reported as in Lin *et al.* [17]. Our proposed zero-shot relation label classification approach achieves an increased score of 5.4% over the previous state-of-the-art models.

TABLE 7. Performance of KGQA on WebQSP-WD test set. Models marked with (*) are the re-implementation from Sorokin and Gurevych [19] to meet the KGQA task.

System	Precision	Recall	F1
STAGG* [51], [53]	0.191	0.227	0.183
Single Edge*	0.224	0.271	0.215
Pooled Edges*	0.209	0.255	0.203
GNN*	0.242	0.289	0.233
GGNN [19]	0.269	**0.318**	0.259
Tree-KGQA	**0.327**	0.233	**0.273**

TABLE 8. Component-wise results of tree-KGQA.

Approach	Precision	Recall	F1
Entity Linking (EL)	0.854	0.810	0.831
Relation Linking (RL)	0.396	0.288	0.334
$KGQA_{ER}$	0.739	0.709	0.724
$KGQA_{k=1}$ (with EL and RL)	0.319	0.219	0.260
$KGQA_{k=2}$ (with EL and RL)	0.327	0.233	0.273

TABLE 9. Our introduced new baseline for the KGQA task on LC-QuAD 2.0 (KBpearl).

System	Precision	Recall	F1
Tree-KGQA	0.526	0.520	0.523

TABLE 10. Ablation study.

Task	Approach	F1	Δ
EL	EL (TF-IDF)	0.599	-
	EL ($FAISS_{KNN}$ + Fasttext)	0.661	+ 6.2%
	EL ($FAISS_{KNN}$ + Sentence-BERT)	0.682	+ 2.1%
	EL ($FAISS_{KNN}$ + disambiguation)	0.700	+ 1.8 %
RL	RL (Cosine similarity)	0.373	-
	RL (BART)	0.464	+ 9.1%
KGQA	KGQA (without tree-disambiguation)	0.244	-
	KGQA (with tree-disambiguation + Fasttext)	0.265	+ 2.1%
	KGQA (with tree-disambiguation + KGE)	0.273	+ 0.8%

3) KGQA

We report the KGQA score on WebQSP-WD dataset in Table 7. Our introduced Tree-KGQA system achieves an improved result (1.4% rise in F1 score) compared to the previous KGQA baselines. The KGQA scores reported in this paper are computed with $k = 2$. Furthermore, we provide a new baseline for the KGQA task on the LC-QuAD 2.0 KBpearl test set in Table 9. Moreover, we report the component-wise results of our proposed techniques on WebQSP-WD dataset in Table 8. The entries with the approach $KGQA_{ER}$ reflect the KGQA score given the ground truth values of EL and RL. We observe an improved KGQA

TABLE 11. Case study.

Task	Question	Ground Truth	Falcon 2.0	PNEL	Our approach
EL	What is in work of actor of Looney Tunes Super Stars' Pepe Le Pew: Zee Best of Zee Best ?	Looney Tunes Super Stars' Pepe Le Pew: Zee Best of Zee Best (Q6675710)	Looney Tunes Super Stars' Pepe Le Pew: (Q6675705), Best (Q4896530)	Looney Tunes Super Stars' Pepe Le Pew: (Q6675705)	Looney Tunes Super Stars' Pepe Le Pew: Zee Best of Zee Best (Q6675710)
	What is the country for head of state of mahmoud abbas?	country (Q6256), Mahmoud Abbas (Q127998)	Mahmoud Abbas (Q10515624)	Mahmoud Abbas (Q10515624)	Mahmoud Abbas (Q127998)

Task	Question	Ground Truth	Falcon 2.0	Our approach
RL	What is the socialist state for contains administrative territorial entity of Beijing?	contains administrative territorial entity (P150), instance of (P31)	contains administrative territorial entity (P131)	contains administrative territorial entity (P150)
	What kind of disease does montel williams have?	medical condition (P1050)	-	medical condition (P1050)

Task	Question	Ground Truth	GGNN	Our approach
KGQA	Where is jamarcus russell from?	Mobile (Q79875)	Mobile (Q79875)	Mobile (Q79875)
	Who did tim tebow play college football for?	Florida Gators football (Q5461394)	Florida Gators football (Q5461394)	Florida Gators football (Q5461394), Denver Broncos (Q223507), New York Jets (Q219602), Philadelphia Eagles (Q219714)

score with $k = 2$ than $k = 1$. It is noteworthy that increasing the value of k increases the search space. Although our system performs remarkably on the EL and answer entity extraction tasks, it has a relatively poor KGQA score due to the low RL score. Nevertheless, relation linking (RL) is a challenging task that is still far from being solved.

V. ANALYSIS

A. ABLATION STUDY

We conduct an ablation study to investigate the effectiveness of major components of our proposed system. Table 10 demonstrates the improvement that each of the components brings to the overall performance of the system. A TF-IDF based entity linking approach exhibits a low F1 score of 0.599, where our proposed indexing mechanism based approach achieves significant gain in the performance (+6.2% using Fasttext and +2.1% using Sentence-BERT embedding). A relation-based entity disambiguation method further improved the result by 1.8%. Our proposed BART-based relation linking approach demonstrates a remarkable improvement (+9.1%) over the cosine similarity based relation linking method.

Furthermore, we assess the performance of the answer extraction component without our proposed tree disambiguation technique. We extract the entities directly connected to the linked entities by the predicted relation as answer entities which achieves a low KGQA F1 score of 0.243. Then, we employ the *tree-walking* and *tree-disambiguation* technique which improves the F1 score by 2.1%. Moreover, we utilized knowledge graph-based embedding during the answer entity extraction procedure to compute the similarity between the predicted relation and the branches of every node in a tree. This method allows the system to surpass Fasttext embedding based similarity calculation by 0.8%.

B. CASE STUDY

Table 11 shows two cases from the entity linking, relation linking and KGQA tasks. The entity and relation linking cases are from LC-QuAD 2.0, where the KGQA cases are from WebQSP-WD.

1) ENTITY LINKING (EL)

Our proposed approach correctly detected and linked the entity in the first case, where Falcon 2.0 and PNEL failed to link the correct entity. This is a challenging case since it contains a long entity span. The underlined texts indicate the entity span in the question. In the second case, all the systems failed to detect *country* as the entity. Although *mahomoud abbas* is correctly detected as entity mention by Falcon 2.0 and PNEL, they linked the entity mention to the wrong KG entity *Mahmoud Abbas (Q10515624)*, who is a footballer. On the contrary, with the help of entity disambiguation where relation information is used, our method correctly linked the mention *mahomoud abbas* to the correct KG entity *Mahmoud Abbas (127998)*, who is the head of a state.

2) RELATION LINKING (RL)

The first case comprises *administrative territorial entity (P150)* and *instance of (P31)* as the ground truth relation. Since *instance of (P31)* does not appear explicitly in the question, it is difficult for the systems to predict it as a relation. In the second case, our proposed Algorithm 5 correctly predicted the relation *medical condition (P1050)*. We adopt a BART-large model [15] in Algorithm 5, pre-trained on natural language inference task, which gives better inference

capabilities in identifying the correct relation from a set of candidate relations.

3) KGQA

Our proposed unsupervised KGQA approach correctly extracted the answer entity in the first case. In the second case, *Florida Gators football (Q5461394)* is given as the ground truth which can be inferred by the relation *member of sports team (P54)* connected to the entity *Tim Tebow (Q517467)*. However, our system extracted all the entities as the answer entities that are connected to *Tim Tebow (Q517467)* by the relation *member of sports team (P54)*.

C. ERROR ANALYSIS AND LIMITATIONS

We conducted an error analysis to understand the cases where our system is not performing as expected. We observed that our proposed entity linker is unable to detect entities that are not named entities such as *president* (*Q30461*) and *governor* (*Q132050*), since it is using NER for detecting the entity mention(s). Here, *Q30461* and *Q132050* are Wikidata ID of the respective entities.

The most challenging aspect of KGQA is relation identification. Relations with similar labels exist in the Wikidata KG, which are difficult for systems to differentiate. For instance, the relations *head of government* (*P6*) and *head of state* (*P35*). This issue becomes more visible when we found that, F1 score on top-3 predicted relation is 49.39 and in top-10 it is 57.66. The relation accuracy results reported in Table 6 are based on the top-1 predicted results from the proposed zero-shot relation linker. Our system fails to predict relations requiring more complex reasoning capabilities, such as hierarchical relationships. For instance, for the question "`Give me cinematic technique that contains the word tilt in their name`", the correct relation that can be used to answer the question is *Instance of* (*P31*), which our system failed to capture. Furthermore, our proposed zero-shot relation linker can only predict one relation. Although this is a limitation of the system, questions generally contain one relation in the context of question answering.

Although our proposed answer extraction method is fairly straightforward, we observe that the KGQA model mainly suffers in the cases where no entities are predicted and the cases where a wrong relation is predicted. Similar to the relation linking, our system also fails to extract the correct answer entities for cases where comparative or logical reasoning is required to answer the questions (E.g., *Is Lake Baikal bigger than the Great Bear Lake?*).

D. DISCUSSION

The improved entity linking performance of our proposed model across all the benchmark datasets provides a solid foundation for the KGQA task. Despite the fact that our proposed relation linking approach outperforming previous methods in complex QA, it could benefit further from better logical inference capabilities. Furthermore, we designed our system in a modular way so that it can be easily extended and used across different KGQA sub-tasks. Within the scope of this paper, we explored Wikidata based datasets. However, from the description of our approaches, we can intuitively say that our system can be adapted for other knowledge graph based datasets. For that, first, the pre-processing step where entity indexing is performed needs to be executed. Then, we need to obtain the relation embedding from a knowledge graph embedding model to perform tree-walking (Section III-C).

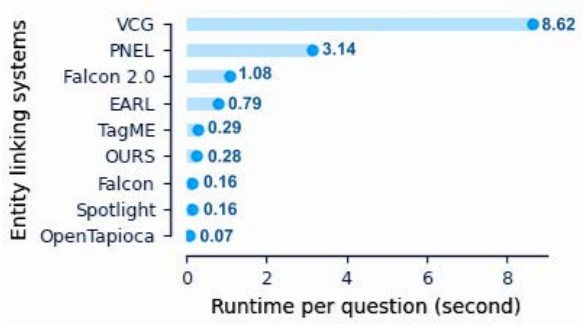

FIGURE 4. Inference time efficiency of the entity linking systems.

Our proposed KGQA system is runtime efficient. Several factors contributed to the fast runtime of our system. In entity linking, the FAISS indexing technique provides fast candidate generation (takes ∼0.04 seconds to generate 10 candidates per question). The performance of the entity linking baselines is shown in Figure 4 (baseline runtimes are reported from Banerjee *et al.* [41]). Furthermore, the relation linking component requires ∼0.09 seconds per question. Moreover, our proposed tree-based answer extraction process takes ∼0.39 seconds per question. Overall, the system takes ∼0.76 seconds per question to perform the entire KGQA task.

VI. CONCLUSION AND FUTURE WORK

We presented Tree-KGQA, an unsupervised technique to perform KGQA without any explicit training. Despite the simplicity, our proposed pre-trained language model-based, unsupervised method outperforms existing supervised systems by a fair margin in all the sub-tasks involved in KGQA. To substantiate our claim, we evaluate our proposed system across several benchmark datasets. Tree-KGQA achieves 4.5%, 7.1%, and 0.1% improvement in the entity linking task on LC-QuAD 2.0, LC-QuAD 2.0 (KBpearl), and QALD-7-Wiki datasets, respectively. Furthermore, it achieves a 5.4% gain in the relation linking task on LC-QuAD 2.0 (KBpearl) and 1.4% improvement in the KGQA task on the WebQSP-WD test set. Although our system proves to be useful for the majority of the types of questions found in the datasets studied, further work is required to tackle more challenging questions requiring counting, comparisons, and logical reasoning capabilities. In our future work, we plan to perform an extensive evaluation on datasets that are based on other knowledge graphs such as DBpedia [1] and

Freebase [2]. Additionally, we want to explore the possibility of advanced clustering methods such as [53], [54] for the entity clustering task.

REFERENCES

[1] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer, "DBpedia—A large-scale, multilingual knowledge base extracted from Wikipedia," *Semantic Web*, vol. 6, no. 2, pp. 167–195, 2015.

[2] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: A collaboratively created graph database for structuring human knowledge," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2008, pp. 1247–1250.

[3] D. Vrandečić, "Wikidata: A new platform for collaborative data collection," in *Proc. 21st Int. Conf. Companion World Wide Web*, 2012, pp. 1063–1064.

[4] G. Maheshwari, P. Trivedi, D. Lukovnikov, N. Chakraborty, A. Fischer, and J. Lehmann, "Learning to rank query graphs for complex question answering over knowledge graphs," in *Proc. Int. Semantic Web Conf.* Auckland, New Zealand: Springer, 2019, pp. 487–504.

[5] S. Vakulenko, J. D. F. Garcia, A. Polleres, M. de Rijke, and M. Cochez, "Message passing for complex question answering over knowledge graphs," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 1431–1440.

[6] W. Zhao, T. Chung, A. Goyal, and A. Metallinou, "Simple question answering with subgraph ranking and joint-scoring," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.* Minneapolis, MN, USA: Association for Computational Linguistics, Jun. 2019, pp. 324–334.

[7] S. Mohammed, P. Shi, and J. Lin, "Strong baselines for simple question answering over knowledge graphs with and without neural networks," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 2. New Orleans, LA, USA: Association for Computational Linguistics, 2018, pp. 291–296.

[8] B. Z. Li, S. Min, S. Iyer, Y. Mehdad, and W.-T. Yih, "Efficient one-pass end-to-end entity linking for questions," in *Proc. EMNLP*, 2020, pp. 6433–6441.

[9] L. Wu, F. Petroni, M. Josifoski, S. Riedel, and L. Zettlemoyer, "Scalable zero-shot entity linking with dense entity retrieval," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds. Stroudsburg, PA, USA: Association for Computational Linguistics, Nov. 2020, pp. 6397–6407.

[10] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann, "Shortcut learning in deep neural networks," *Nature Mach. Intell.*, vol. 2, no. 11, pp. 665–673, Nov. 2020.

[11] A. Fabbri, P. Ng, Z. Wang, R. Nallapati, and B. Xiang, "Template-based question generation from retrieved sentences for improved unsupervised question answering," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2020, pp. 4508–4513.

[12] O. Ram, Y. Kirstain, J. Berant, A. Globerson, and O. Levy, "Few-shot question answering by pretraining span selection," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, vol. 1. Stroudsburg, PA, USA: Association for Computational Linguistics, 2021, pp. 3066–3079.

[13] D. B. Nguyen, A. Abujabal, N. K. Tran, M. Theobald, and G. Weikum, "Query-driven on-the-fly knowledge base construction," *Proc. VLDB Endowment*, vol. 11, no. 1, pp. 66–79, Sep. 2017.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[15] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics, (ACL)*. Stroudsburg, PA, USA: Association for Computational Linguistics, Jul. 2020, pp. 7871–7880.

[16] M. Dubey, D. Banerjee, A. Abdelkawi, and J. Lehmann, "LC-QuAD 2.0: A large dataset for complex question answering over Wikidata and DBpedia," in *Proc. 18th Int. Semantic Web Conf. (ISWC)*. Auckland, New Zealand: Springer, 2019, pp. 69–78.

[17] X. Lin, H. Li, H. Xin, Z. Li, and L. Chen, "KBPearl: A knowledge base population system supported by joint entity and relation linking," *Proc. VLDB Endowment*, vol. 13, no. 7, pp. 1035–1049, Mar. 2020.

[18] R. Usbeck, A.-C. N. Ngomo, B. Haarmann, A. Krithara, M. Röder, and G. Napolitano, "7th open challenge on question answering over linked data (QALD-7)," in *Semantic Web Evaluation Challenge*. Portorož, Slovenia: Springer, 2017, pp. 59–69.

[19] D. Sorokin and I. Gurevych, "Modeling semantics with gated graph neural networks for knowledge base question answering," in *Proc. 27th Int. Conf. Comput. Linguistics*. Santa Fe, NM, USA: Association for Computational Linguistics, Aug. 2018, pp. 3306–3317.

[20] L. Logeswaran, M.-W. Chang, K. Lee, K. Toutanova, J. Devlin, and H. Lee, "Zero-shot entity linking by reading entity descriptions," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 3449–3460.

[21] A. Sakor, I. O. Mulang, K. Singh, S. Shekarpour, M. E. Vidal, J. Lehmann, and S. Auer, "Old is gold: Linguistic driven approach for entity and relation linking of short text," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, 2019, pp. 2336–2346.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[23] W. Yu, L. Wu, Y. Deng, R. Mahindru, Q. Zeng, S. Guven, and M. Jiang, "A technical question answering system with transfer learning," in *Proc. Conf. Empirical Methods Natural Lang. Process., Syst. Demonstrations*. Stroudsburg, PA, USA: Association for Computational Linguistics, Oct. 2020, pp. 92–99.

[24] M. Miwa and M. Bansal, "End-to-end relation extraction using LSTMs on sequences and tree structures," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, vol. 1. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1105–1116.

[25] Y. Zhang, P. Qi, and C. D. Manning, "Graph convolution over pruned dependency trees improves relation extraction," in *Proc. Conf. Empirical Methods Natural Lang. Process.* Brussels, Belgium: Association for Computational Linguistics, Oct./Nov. 2018, pp. 2205–2215.

[26] P. Wu, S. Huang, R. Weng, Z. Zheng, J. Zhang, X. Yan, and J. Chen, "Learning representation mapping for relation detection in knowledge base question answering," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 6130–6139.

[27] R. Nedelchev, D. Chaudhuri, J. Lehmann, and A. Fischer, "End-to-end entity linking and disambiguation leveraging word and knowledge graph embeddings," 2020, *arXiv:2002.11143*.

[28] X. Huang, J. Zhang, D. Li, and P. Li, "Knowledge graph embedding based question answering," in *Proc. 12th ACM Int. Conf. Web Search Data Mining*, Jan. 2019, pp. 105–113.

[29] O. Levy, M. Seo, E. Choi, and L. Zettlemoyer, "Zero-shot relation extraction via reading comprehension," in *Proc. 21st Conf. Comput. Natural Lang. Learn. (CoNLL)*, 2017, pp. 333–342.

[30] Y. Lan and J. Jiang, "Query graph generation for answering multi-hop complex questions from knowledge bases," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, Jul. 2020, pp. 969–974.

[31] C. Liang, J. Berant, Q. Le, K. D. Forbus, and N. Lao, "Neural symbolic machines: Learning semantic parsers on freebase with weak supervision," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*. Vancouver, BC, Canada: Association for Computational Linguistics, Jul. 2017, pp. 23–33.

[32] A. Miller, A. Fisch, J. Dodge, A.-H. Karimi, A. Bordes, and J. Weston, "Key-value memory networks for directly reading documents," in *Proc. Conf. Empirical Methods Natural Lang. Process.* Austin, TX, USA: Association for Computational Linguistics, Nov. 2016, pp. 1400–1409.

[33] H. Sun, T. Bedrax-Weiss, and W. Cohen, "PullNet: Open domain question answering with iterative retrieval on knowledge bases and text," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*. Hong Kong: Association for Computational Linguistics, Nov. 2019, pp. 2380–2390.

[34] H. Zafar, G. Napolitano, and J. Lehmann, "Formal query generation for question answering over knowledge bases," in *Proc. Eur. Semantic Web Conf.* Heraklion, Greece: Springer, 2018, pp. 714–728.

[35] P.-N. Kung, T.-H. Yang, Y.-C. Chen, S.-S. Yin, and Y.-N. Chen, "Zero-shot rationalization by multi-task transfer learning from question answering," in *Proc. Findings Assoc. Comput. Linguistics (EMNLP)*, 2020, pp. 2187–2197.

[36] M. Yan, H. Zhang, D. Jin, and J. T. Zhou, "Multi-source meta transfer for low resource multiple-choice question answering," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 7331–7341.
[37] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 3982–3992.
[38] J. Johnson, M. Douze, and H. Jegou, "Billion-scale similarity search with GPUs," *IEEE Trans. Big Data*, vol. 7, no. 3, pp. 535–547, Jul. 2021.
[39] S. Zhang, Y. Tay, L. Yao, and Q. Liu, "Quaternion knowledge graph embeddings," in *Advances in Neural Information Processing Systems*, vol. 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2019.
[40] J. Berant, A. Chou, R. Frostig, and P. Liang, "Semantic parsing on freebase from question-answer pairs," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2013, pp. 1533–1544.
[41] D. Banerjee, D. Chaudhuri, M. Dubey, and J. Lehmann, "PNEL: Pointer network based end-to-end entity linking over knowledge graphs," in *Proc. Int. Semantic Web Conf.* Athens, Greece: Springer, 2020, pp. 21–38.
[42] P. N. Mendes, J. Daiber, M. Jakob, and C. Bizer, "Evaluating DBpedia spotlight for the TAC-KBP entity linking task," in *Proc. TAC-KBP Workshop*, vol. 116, 2011, pp. 118–120.
[43] P. N. Mendes, M. Jakob, A. García-Silva, and C. Bizer, "DBpedia spotlight: Shedding light on the web of documents," in *Proc. 7th Int. Conf. Semantic Syst.*, 2011, pp. 1–8.
[44] P. Ferragina and U. Scaiella, "TAGME: On-the-fly annotation of short text fragments (by Wikipedia entities)," in *Proc. 19th ACM Int. Conf. Inf. Knowl. Manage.*, 2010, pp. 1625–1628.
[45] L. Del Corro and R. Gemulla, "ClausIE: Clause-based open information extraction," in *Proc. 22nd Int. Conf. World Wide Web*, 2013, pp. 355–366.
[46] M. Dubey, D. Banerjee, D. Chaudhuri, and J. Lehmann, "EARL: Joint entity and relation linking for question answering over knowledge graphs," in *Proc. Int. Semantic Web Conf.* CA, USA: Springer, 2018, pp. 108–126.
[47] I. O. Mulang, K. Singh, and F. Orlandi, "Matching natural language relations to knowledge graph properties for question answering," in *Proc. 13th Int. Conf. Semantic Syst.*, Sep. 2017, pp. 89–96.
[48] D. Sorokin and I. Gurevych, "Mixing context granularities for improved entity linking on question answering data across entity categories," in *Proc. 7th Joint Conf. Lexical Comput. Semantics*. New Orleans, LA, USA: Association for Computational Linguistics, Jun. 2018, pp. 65–75.
[49] A. Sakor, K. Singh, A. Patel, and M.-E. Vidal, "Falcon 2.0: An entity and relation linking tool over Wikidata," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 3141–3148.
[50] W.-T. Yih, M.-W. Chang, X. He, and J. Gao, "Semantic parsing via staged query graph generation: Question answering with knowledge base," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process*, vol. 1. Beijing, China: Association for Computational Linguistics, Jul. 2015, pp. 1321–1331.
[51] A. Delpeuch, "OpenTapioca: Lightweight entity linking for Wikidata," 2019, *arXiv:1904.09131*.
[52] J. Bao, N. Duan, Z. Yan, M. Zhou, and T. Zhao, "Constraint-based question answering with knowledge graph," in *Proc. 26th Int. Conf. Comput. Linguistics, Tech. Papers (COLING)*. Osaka, Japan: The COLING Organizing Committee, Dec. 2016, pp. 2503–2514.
[53] L. Hu, X. Pan, Z. Tan, and X. Luo, "A fast fuzzy clustering algorithm for complex networks via a generalized momentum method," *IEEE Trans. Fuzzy Syst.*, early access, Oct. 4, 2021, doi: 10.1109/TFUZZ.2021.3117442.
[54] L. Hu, K. C. C. Chan, X. Yuan, and S. Xiong, "A variational Bayesian framework for cluster analysis in a complex network," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 11, pp. 2115–2128, Nov. 2020.

MD RASHAD AL HASAN RONY received the bachelor's degree from BRAC University, Bangladesh, and the master's degree from the University of Bonn, Germany, where he is currently pursuing the Ph.D. degree with the Smart Data Analysis Group. He is currently a Research Scientist at Fraunhofer IAIS, Dresden. His primary research interests include knowledge graph based dialogue systems, question answering systems, evaluation of generative systems, machine reading comprehension, and language models.

DEBANJAN CHAUDHURI is currently pursuing the Ph.D. degree with the University of Bonn. His Ph.D. thesis titled "Enriching Text-Based Human-Machine Interactions with Additional World Knowledge." He is an AI Expert with Uniper. He has several years of expertise solving business challenges with machine learning, especially deep learning. He is actively working on document understanding and processing using natural language processing and computer vision. His research interests include chatbots, question answering, optical character recognition, and image generation using generative adversarial networks.

RICARDO USBECK received the Ph.D. degree from the University of Leipzig, in 2017. He has been a Junior Professor (W1TTW2) of semantic systems with the University of Hamburg, since May 2021. His main research interests include circulate around knowledge-driven, semantic technologies, and methods to enable computers to understand and help humans.

JENS LEHMANN received the Ph.D. degree *(summa cum laude)* from the University of Leipzig and the joint master's degree in computer science from the Technical University of Dresden and the University of Bristol. He is currently the Head of the Smart Data Analysis Research Group, a Full Professor with the University of Bonn, and a Lead Scientist with Fraunhofer IAIS. He has authored more than 100 publications, which were cited more than 18 000 times and have won 12 international awards. His research interests include semantic web technologies, question answering, machine learning, and knowledge graph analysis. He contributed to various open-source projects such as DL-Learner, SANSA, LinkedGeoData, and DBpedia.