Enhancing Question Answering with a Free-text Knowledge Graph

Huishi Qiu

2672234

Supervisor

Dr. Michael Cochez

Faculty of Science

VU University Amsterdam

The Netherlands

August 26, 2021

Huishi Qiu: *Enhancing Question Answering with a Free-text Knowledge Graph*, August 26, 2021

Supervisor:

Dr. Michael Cochez

Second Reader:

Dr. Stefan Schlobach

Abstract

Question Answering(QA) has been a long-term study subject in both Knowledge Representation and Natural Language Processing. Each field has introduced its own method, i.e. Knowledge Graph (KG) QA and Information Retrieval (IR) QA, respectively. However, KGQA suffers from a labor-intensive building process and a low recall rate while the IR method is limited by the noise in the text. Recent work proposed the free-text knowledge graph to store semi-structured data, and a Graph Neural Network(GNN) based QA system DELFT to reason over. Still, the free-text KG is fulfilled with irrelevant information to the question, which undermined the DELFT's general performance.

The goal of this research is to increase the DELFT's QA performance as well as improve the process and result's explainability. We propose a framework able to derive labels from the existing free-text KG. To this end, we perform two text classification tasks on question sentences and entity descriptions. Through matching a pruned graph is generated, of which irrelevant entities to question are excluded according to label. Besides externally pruning the graph, we introduce DELFT-LUKE, which internally integrates DELFT with an entity-oriented language model. The experiments show that our pruning method is capable of reducing the graph size by at least 40% and further improved the DELFT's performance by 4.4% on TriviaQA and 9.8% on QBLink. This indicates the benefits of our pruning method overtake its deficiency on QA result, also provide critical insight into the explainability of the QA process.

Table of Contents

1 Introduction

Factoid question answering(QA) has been a long-term study subject that acts as the downstream task for both Knowledge Representation and Natural Language Processing. Thanks to the television shows like *Who Wants to be a Millionaire* and *Jeopardy!*, as a subject QA is well known by the public and retained popularity for more than a decade. Leveraged by neural networks, QA enjoyed a revival following the success of DrQA[6]. With rapid improvements over models, very recently, some works turned toward the more challenging task of complex question answering[55,54,1], which needs to integrate multiple pieces of evidence from the corpus to answer a single question.

A key challenge in QA is the imbalance of the entity and especially relationship distribution in the knowledge bases, which is summarized as the long-tail effect. Since the knowledge base act as the basis for the entire system, tackling the long-tail effect is essential to the general performance of QA. For instance, 70% of relationships appear less than a thousand times in the popular New York Times corpus's[39]. In Knowledge Graph, the long-tail effect also happens with the usage of Wikidata property[37], and the distribution of SPARQL queries[4]. To state more graphically, the prevalence of all results roughly follows an exponential distribution as illustrated in fig. 1. Despite popular entities or relationships appear in the head, most others exist with a low frequency. However, due to the amount, adding up all low-frequency results contribute to a large share of the total outcome. As displayed in the figure, the deep-colored area in the head and the shallow-colored area in the tail are approximately the same sizes. Insufficient data for low-frequency relationships made the QA model's training struggle, poor performance on unfamiliar questions decreased the model's general capability.

To deal with the long-tail effect in QA, the main concern is to (1) acquire rich context also for low-frequency data, (2) meanwhile, retain structured form to ensure the data's accessibility. To address the issue, many works embraced semi-structured data[2,52,46] as the new format for knowledge base, which solved

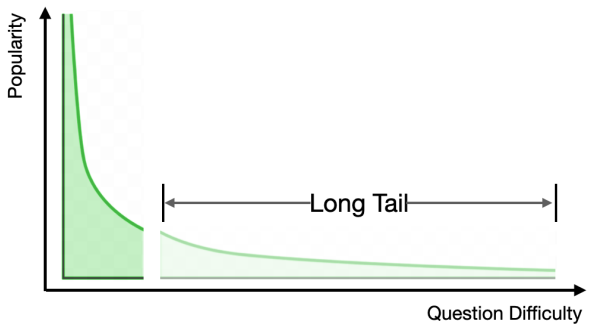


Fig. 1: An illustration of long-tail effect in QA

the problem of accessibility. To overcome the issue regarding low-frequency data, a novel approach named free-text knowledge graph[54] had been proposed as an intermediate data structure. Free-text KG preserved triple as the basic element while adopting sentences to replace the relationship to link between entity nodes, which solved the problem of relationships' low recall rate in QA. Afterward, the authors proposed a Graph Neural Network(GNN) named DELFT act as the reasoner to derive the answer from the evidence graph.

However, the mechanism of linking in free-text KG is based on the co-occurrence and sentence matching, which would inevitably introduce noisy text not related to the question's intent. The authors of DELFT rely on GNN, hopefully selecting useful information by a series of modifications on the question and evidence sentences, however, still missing the essential step to understanding the question's intention. According to a study[36], DrQA's over 30% of the incorrect answers could be attributed to unmatched type label between the entity and expected answer. In addition to retrieving semantically related entities, we should also take the question's intention into account, more specifically in factoid QA, figuring out the Expected Answer Type(EAT). Would it help to improve the QA performance by only remaining question-asked types of entities in the free-text KG?

On the contrary of too much evidence, another difficulty of QA comes with no informative gold evidence. Despite IRQA models become more and more sophisticated, information retrieval is different from QA to a great degree[43]. IR focuses on matching, while from the starting point QA is looking for unknown information. Due to this uncertainty, many questions may not have direct evidence that points to the answer. Thus in these scenarios, just like the human player in a quiz bowl game, the model has to guess the most likely answer to the best of its knowledge, i.e. the language embedding for DELFT. Recent works have demonstrated the impressive gains of neural models that have been pre-trained on a language modeling task[32,35,8]. In the previous hypothesis, we externally scope down the candidate set's size to increase the possibility of selecting the correct answer. Since entities play an important role as nodes in forming the free-text KG, would integrating an entity-focused language model into DELFT improve the QA benchmark internally?

To this end, we formulate two research questions, focusing on both external and internal modification's effect on QA system DELFT's performance over free-text KG:

- RQ1: *Does pruning the graph by excluding irrelevant nodes to question's intention, improve the DELFT's performance?*
- RQ2: *Can an entity oriented language representation model improve the DELFT's reasoning ability over free-text KG?*

For RQ1, we propose a pruning framework to leverage the existing information from the free-text knowledge graph. The core idea is to introduce a type label as a general property into the free-text KG for both questions and entities. By matching the label, a pruned free-text can be generated with more focus on the question's intention. In order to do that, the framework includes a question sentence classifier aim at extracting the question's Expected Answer Type, and an entity classifier aiming at summarizing the description into a label. By purposely training a fine-tuned language model with a simple KG, our model is able to achieve high accuracy on both text classification tasks. To evaluate the

outcome, we compare the DELFT's performance on both original and pruned free-text graphs.

RQ2 focusing on the effect on introducing knowledge from the language model, we integrate LUKE[48] with the QA system DELFT. LUKE extended the BERT's pretraining process naturally by also predicting masked entities along with the tokens, which achieved several promising results on entity-related tasks. To testify the RQ2, we perform experiments with multiple language embeddings to compare the output.

Overall, the contributions of this work are summarized as follows.

- We introduce a new and more complex labeled QA dataset from TriviaQA[20] and QBLink[13], following the taxonomy of FIGER[27]. The dataset is composed of a varied length of questions, which can be further easily extended following the method of zero-shot classification.
- We proposed a pruning framework for DELFT on free-text KG, consists of a question and an entity description classifier. Leveraged by the fine-tuned language model trained through BLP[7] with a simple KG, the model is able to derive labels with decent accuracy for both questions and entities.
- The results from experiments show that, our pruning framework can efficiently reduce the graph size by at least 40%. This pruning on graph benefited the GNN, outperformed the previous DELFT's state of the art performance with a wide margin of 4.4% and 9.8% on TriviaQA and QBLink.

The rest of the thesis is organized in the following way. In chapter 2, we discuss relevant topics with respect to free-text KG, text classification, and multiple neural networks. In Chapter 3, we introduce our proposed pipeline for graph pruning, together with dataset construction, methodologies of classification, and the DELFT model. In chapter 4, we perform experiments on two text classification tasks regarding question and entity description. Afterward, we study the effectiveness and efficiency of the pruning process. In Chapter 5, we testify our research questions by QA over the pruned graph and evaluate the newly introduced model DELFT-LUKE. Chapter 6 analyze the previous experiments

results and perform a case study. And finally, in Chapter7 conclusion and future work are discussed.

2 Related Work

2.1 Free-text Knowledge Graph

Knowledge Graph[17] emerged in recent years as a structured way to abstract and organize the facts and knowledge in the real world. The fundamental element in KG is a triple, which consists of entities with their descriptions, properties and relationships to other entities.

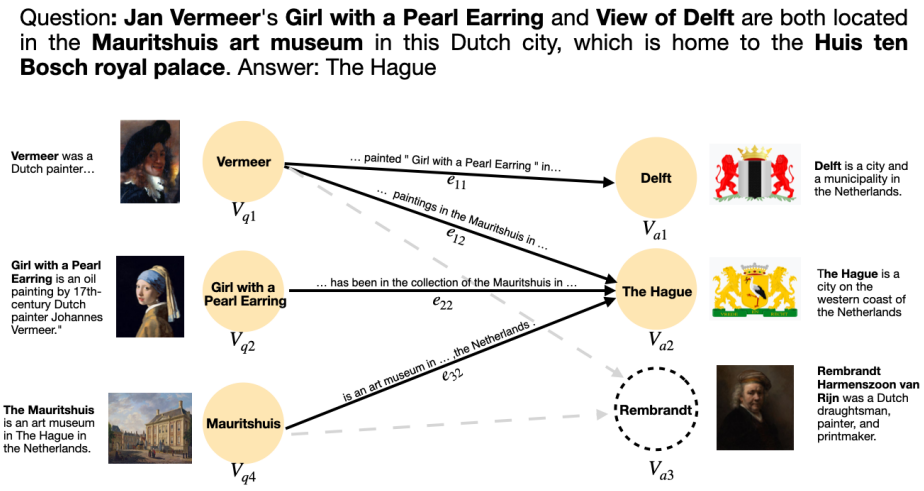


Fig. 2: An illustration of a question's free-text Knowledge Graph. Candidate entity Rembrandt on the right has been excluded as the question is explicitly asking a Dutch city.

As mentioned above, the relationships in a triple, e.g. property distribution in the Wikidata, also following the long-tail effect, which makes it hard to coverage. Compare to relationships, triple's another component entity has a relatively high recall rate, based on the fact that entities are more commonly acknowledged by the public while defining a relationship engages more human effort. This can be reflected by the number of relationship types are much higher than the number of entities. A statistical study[12] on knowledge bases showed that Freebase[3]

is composed of 1500 entity types and 40M instances while the relation amounts are 35000 types and 637M instances. Coming to a specific question, it would be relatively easy to retrieve relevant entities, while harder to link entities with questions following KG's predefined relationships, i.e. high recall rate on entities while low recall rate on the relationship. Thus free-text KG is proposed mainly focus on replacing relationships with sentences. This is inspired by the IRQA that searching text from corpus would have a much higher chance to retrieve relevant information compared to searching predefined relationships in KG. By adopting sentence as relationship, free-text KG has a relatively high recall rate on both entities and relationships.

Each question in free-text KG is defined as a tuple in following manner: $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{T}, \mathcal{D})$, consist by entity nodes $\mathcal{V}$, evidence sentences $\mathcal{E}$, semi-structured triples $\mathcal{T}$ and entity descriptions $\mathcal{D}$. Each triple in $\mathcal{T}$ is formed by (v_q, e, v_a) where the v_q represents an entity in the question, v_a as a candidate entity and e is the free-text sentence that build up the connection of between both. An example is given on how question answering is performed over the free-text KG in fig. 2. The overall structure of free-text KG is a bipartite graph. Given a question, the first step is to extract all entities from question text, which is implemented with Named Entity Recognition tool TagMe[1] with a certain threshold. Question entities are displayed on the left-hand side and represented as V_{qn}. Next, potential candidate answers V_{an} are collected following two methods: Either based on the co-occurrence in the sentence with question entities V_{qn} or using an information retrieval model to select the most relevant results. The total number of candidate entities is limited to a certain number to restrain the graph size. Candidates entities are displayed on the right side of the graph. Lastly, sentences that both entities occurred are selected and the free text on the edge is preserved as the edge feature. The edge may be composed of several sentences, which indicates the candidate is highly relevant to the question. Wikipedia dump serves as the corpus for the building process.

[1] https://sobigdata.d4science.org/web/tagme/tagme-help

During the building of free-text KG, the linking between question entities and candidate entities is either achieved by co-occurrence or information retrieval. This only indicates candidates are related to the question entity semantically to some extent, but not necessarily relevant to the entire question's intention. This leads to a large number of irrelevant nodes in the candidate set V_{an}, along with invalid sentences on the edges. In the example question, candidate entity Rembrandt is linked with two question entities: Both *Vermeer* and *Rembrandt* are famous Dutch Golden Age painters; The *Mauritshuis* held many of Rembrandt's paintings. However, *Rembrandt* is apparently not a reasonable candidate given the question is asking for a city explicitly. Thus our later methods focus on reducing irrelevant nodes along with the invalid relationships by considering the question's intent.

2.2 Knowledge Graph Question Answering

KGQA aims to answer a natural language question automatically using the facts from KG. Knowledge graphs such as Wikidata contain a tremendous amount of facts[2], make it difficult for regular users to access. To narrow down the gap, KGQA is proposed to transform natural language questions into queries to retrieve over KG, for instance, translating questions to SPARQL queries [9,16]. KGQA relies on two prerequisites that (1) the queried triple exist in the graph, and (2) query templates are predefined in the system. Despite simple questions, both requirements are hard to fulfill. Thanks to the clean and structured data type, the KGQA model could achieve high precision, though it may struggle when no supporting evidence or query templates are unavailable.

Except matching, another KGQA way is to enrich the language model with knowledge in the KG by fusing the Knowledge Graph Embedding(KGE). The idea is based on KGE contains knowledge in the form of embedding, which may enhance the language model's representation. ERNIE[53] first introduced the idea of incorporating knowledge and language embedding into the same vector

[2] https://wikidata-todo.toolforge.org/stats.php

space. E-BERT[34], adopted the similar idea to inject KGE Wikipedia2Vec[49] into the BERT, which achieved state-of-the-art performance on QA benchmark LAMA[33]. Comparing E-BERT with ERNIE, they both aligned the KGE with the BERT's word vector space. The difference is E-BERT does not need pre-training on BERT, which is more efficient. Besides fusing with the language model, KEQA[18] proposed a direct KGE based QA framework, by jointly setting the question's head entity and tail entity along with predicate representations into the KGE spaces. Afterward, the closest fact to the learned three vectors is returned as the answer. Quey2Box[38] adopted a similar idea to embeds KG entities and queries into a box and further extended the ability to answering complex questions.

2.3 Information Retrieval Question Answering

Besides open domain question answering, another type of QA is based on reading comprehension, which is mostly achieved by the Information Retrieval(IR) models. IR models focus on the so-called cloze style QA: Given a paragraph, the following asked questions can be answered by extracting from the foregoing context. For IR based model, the key issues are to understand the intention of the question and select the appropriate domain from the text span. Traditionally, IRQA used the syntax of the question based on linguistics to understand the questions. Using semantic parsing, the question can be answered by constructing the related question sets[41]. When the question becomes complicated, it may need multiple documents to support a single answer, thus approaches turned toward deep learning methods. DrQA[6] first adopted bidirectional LSTM to tackle semantic understanding, which yielded promising results. Moreover, in recent years the success of massively pre-trained representation models such as BERT[8] and RoBERTa[28] have demonstrated the improvements in QA from the perspective of the language model.

2.4 Text Classification

In this section, we introduce two neural network architects relevant to text classification, namely Recurrent Neural Network(RNN) and Convolution Neural Network(CNN). In the following section, we also discuss another architect GNN, which is used as reasoner to derive answer from multiple evidences.

Recurrent Neural Network In text classification, RNN-based models take a sequence of tokens as input. A string of cells in the model is designed to capture the pattern and dependencies from the input text, which can be saved as parameters. In the very basic structure of the so-called Vanilla RNN, the input is transformed by a single cell with weight and generated output for the next cell. However, the simple cell structure performs poorly, especially having the issue of gradient exploding or gradient vanishing during the backpropagation. Thus more sophisticated cell structure is proposed.

Among a number of variants, Long Short Term Memory(LSTM) is a milestone that is still popular nowadays. In LSTM, the basic cell structure is replaced by a mechanism with series of input, forget and output gates that regulate the data from input to output. Moreover, the introduction of two data flows, i.e. long memory and short memory, granted the model the ability to capture the long-term dependencies compared to the Vanilla-RNN.

In order to leverage the ability of LSTM, several modifications have been proposed. The most relevant version to the QA is the Bidirectional-LSTM[15]. Bi-LSTM is a natural extension from Bidirectional-RNN[40], of which state cells have been separated as forward cells and backward cells to help better capture the feature. In QA, the latter information is essential for understanding the former context. Bi-LSTM had been widely adopted in reading comprehension models.

Convolution Neural Network RNN is used to recognize patterns over time, whereas CNN is trained to recognize patterns across space[25]. Despite the common impression that CNN is applied on pixels to discover certain patterns in

Computer Vision, CNN can be also applied to text tasks, especially to the patterns in the text regardless of time. The scenario includes identifying expressive words in the sentiment analysis and keywords which may influence the sentence classification result.

The pioneering work Dynamic CNN(DCNN), is named after the mechanism using dynamic k-max pooling for selecting syntax information from the input sentence[21]. Later on, a simplified CNN structure[23] has been proposed. Compare to DCNN, the model contains only one convolution layer after the embedding layer. Both models have applied on text classification tasks achieved a state-of-the-art performance at the time.

2.5 Graph Neural Network for Reasoning

Compared to KGQA, IRQA provides a method with a broader but noisier context. Therefore, the new obstacle lies in how to remove noise from noisy text and obtain useful information. In neural networks, GNN has appeared in recent years and has demonstrated its reasoning ability, which is useful for both IRQA and KGQA. GCN [24] first extends the neural network to graph structure data other than images and text. The proposed model can encode graph structure and node features. On the basis of GCN, attention weight is introduced to focus on the important part of the graph, namely Graph Attention Network[45]. For IRQA, CogQA[10] proposed a method of using BERT-based language model to retrieve information and GNN for reasoning. This is motivated by the theory in cognitive psychology where humans use system1 for cognition while system2 for logical thinking. Two modules collaborated with each other to answer multi-hop questions. The system achieved state-of-the-art performance on HotpotQA[51]. A similar combination of BERT and GNN can be also found in Transformer-XH[55]. The highlight is using extra hop to connect different pieces of evidence. It uses the special token [CLS] as the attention hub of the sentence, studying both internal relationships in the sentence and external relationships with other sentences. Besides IRQA, the GNN model also can be performed in KGQA.

STARE[14] introduced how to encode additional qualifier information on edges by encoding and decoding with L-GCN. L-GCN focuses on complex large graphs, and how to propagate edge labels.

3 Approach

We designed the following pipeline to verify our two research questions, namely the effect of introducing type into free-text KG for pruning and integrating entity-oriented language model into DELFT. As displayed in fig. 3, the pipeline is consists of 4 modules. In order to train the classifiers following the manner of supervised learning, firstly we collect a labeled QA dataset using zero-shot classification. Afterward, in order to obtain a type label, we train two separate classifiers for each task of text classification. These labels are used for matching and deriving a pruned graph in the third phrase. By taken both the intention of question and semantic summary of an entity into account, the calibrated graph is aimed to remain question-asked entities only, and discard irrelevant entities along with the evidence sentences. Lastly, regarding the second research question, we adopt a new language model, which generates different embedding representation for GNN.

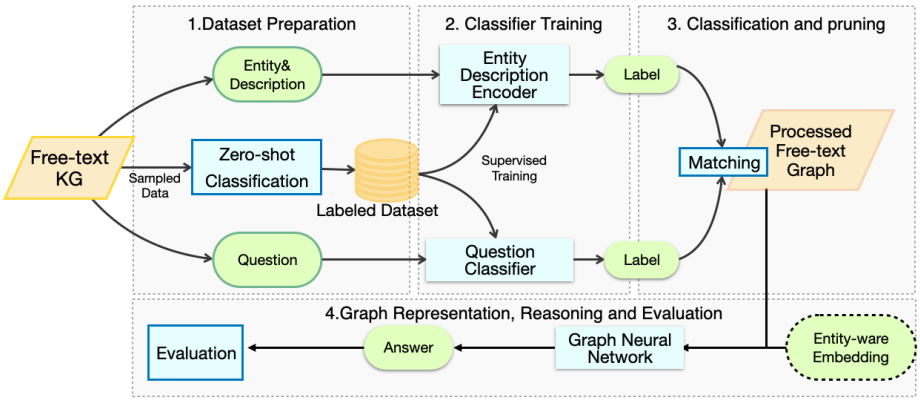


Fig. 3: The proposed pipeline on introducing pruning and new language model for DELFT

3.1 Task Definition

In this work we deal with open-domain QA: Given a question described in natural language, the desired answer is a commonly known entity that can be retrieved from a knowledge base such as Wikipedia. Need to mention that the answer is limited to the entity only and no other types such as a digit or reasoning process are being asked. For instance, giving the following question:

"Name this country composed of seven provinces which gained its independence from Spain in the Eighty Years War." The answer is an entity: "The Netherlands".

In the experiment setting, the task has been converted to select the correct answer from a candidate set: Given a question q and a set of candidate answers $V_{an} = v_p \cup V_n$, composed by a positive and a number of negative answers. After input question's free-text KG $\mathcal{G}$ into GNN model f, the goal is to select positive entity v_p from the candidate set V_{an}. Two research questions can be interpreted as following two sub-tasks that (1) limit the size of candidate answers V_{an}. (2) Training a new set of parameters f_θ for GNN model DELFT-LUKE.

3.2 Labelled Dataset Collection

In order to train classifiers for both entity description and questions by supervised learning, a high-quality labeled dataset is the first prerequisite. The current labeled-QA dataset, namely TREC-10[26] is insufficient for factoid QA, especially complex open-domain QA. This involves three concerns: (1) TREC-10 includes quantity and reasoning questions, which are not asked in the scenario of factoid QA. (2) Most questions in TREC-10 are less than 10 tokens and very few entities are available. In the following examples, there are 1 and 2 entities in question, respectively. Too simple questions will weaken the model's generalization ability when facing long and complex questions. (3) TREC-10 does not provide the supposed answer's entity description, which we needed for training entity classification.

Q: What Canadian city has the largest population?

Q: Which country gave New York the Statue of Liberty?

Since the existing labeled QA dataset does not fulfill our requirement, we choose to collect a new labeled QA dataset, only referencing the setting of TREC-10. We used text-corpora provided by Zhao et al.[54] to build up such a dataset, of which 2 processed free-text KGs are provided, constructed from the QA benchmarks TriviaQA and QBLink. For each question in the free-text KG, the graph includes question text, recognized question entities, candidate entities with their descriptions, and evidence linking question and candidate entities.

Before extracting labels from free-text KG, we examined two available routes: Either rely on external KG like Wikidata's rich attribute or exploit the existing text using NLP techniques. For the KG method, given each entity, we can search the entity's label using WikiMedia's API and derive the corresponding label by property *"isInstance_of"*. There are mainly three drawbacks. Firstly, entity would have multiple *"isInstance_of"* properties in Wikidata. In order to group labels according to the fixed schema, a clustering procedure is needed. For instance in Wikidata, Amsterdam(Q727), is an instance of both city and capital, which are concepts that semantically overlapped. Secondly, the ambiguity is an issue that can not be ignored during the retrieval of the Wikidata: If entities' tokens are exactly identical, the matching would lead to confusion. Common handling is to use heuristics[11]: When facing the issue of identical tokens between entities, select the entity with the lowest QID, which is normally the most common entity. However, this is not that reliable in the scenario of factoid QA, since rare entities are often asked as an indicator of difficulty. Thirdly, introducing new knowledge made the framework less comparable with the original model as new knowledge is introduced.

As for exploiting the existing graph and text, an NLP method named zero-shot classification is preferred: Given a sentence and a set of predefined labels, there is no need for training and the model will return the most semantically related topic to the text, either in the manner of single-label or multi-label classification. The disturbance is the noise in the text, which may influence the

result of classification. For instance, according to the schema of Wikipedia, each movie entity's gloss sentence, i.e. the first sentence must include its director. In zero-shot classification, a movie entity would be classified as a person due to the director as a disturbance. Secondly, using the predefined model is both time and resource-consuming compared to searching KG due to a large amount of computation.

In consideration of the standardization, accuracy, and convenience of matching in the further steps, we chose the NLP method for labeling. This would also make the comparison with the previous DELFT model fairly as no knowledge is newly introduced from KG. For the label's taxonomy, we adopt the FIGER[27] fine-grained label set as the predefined labels for zero-shot classification. As for the final classification output, we will only preserve the coarse-grained label according to FIGER's hierarchy setting. There are several reasons why we use fine-grained labels for classification in zero-shot classification but only coarse labels are preferred at the end. First, the coarse label like other do not have reasonable semantic meaning, thus it is not possible to directly utilize it in zero-shot classification. Second, the fine-grained concepts are usually overlapped with each other since an entity may have several identities. For instance, *Benjamin Franklin* is not only a scientist but also a politician and a writer. Under the coarse labels, all these identities can be grouped under a single label as a person. Third, using over 100 fine-grained labels will make further matching very hard compared to only matching 8 coarse labels, and the model performance barely benefit from it. This can be verified from the previous work where a question classifier is performed on both coarse and fine-grained labels[50]. The result showed that using fine-grained labels has almost identical output with the classifier adopting coarse labels.

Lastly, we adjusted the FIGER ontology with a few modifications, focusing on reducing the ambiguity and the custom of formulating the question. The refined label set is displayed in the fig. 4, labels of which differ from the original FIGER taxonomy are marked in bold on each sub-class tail position. These modifications

	Fine-grained label
1. Person	"doctor", 'actor', 'engineer', 'architect', 'monarch', 'artist', 'musician', 'athlete', 'politician', 'author', 'religious leader', 'coach', 'director', 'soldier', 'terrorist', **'scientist', 'criminal','god','fictional character'**
2. Organization	"terrorist organisation", "airline", "government agency", "company", "government","educational institution", "political party", "fraternity sorority", "educational department", "sports league", "military", "sports team", "news agency"
3. Location	"body of water", "city", "island", "country", "mountain", "county", "glacier", "province","astral body", "cemetery", "park", "bridge", **"state"**
4. Product	"camera", "engine", "mobile phone", "airplane", "computer", "car", "software", "ship", "game", "spacecraft", "instrument", "train", "weapon"
5. Art	"written work", "film", "newspaper", "play", "music"
6. Event	"natural disaster", "election", "sports event", "protest","terrorist attack", **"conflict"**
7. Building	"airport", "dam", "hospital", "hotel", "library", "power station", "restaurant", "sports facility",
8. Other	"time", "color", "award", "educational degree", "law", "ethnicity", "language","religion", "chemical thing", "biology thing", "medical treatment", "disease", "symptom", "drug","body part", "living thing", "animal", "food", "website", "broadcast network", "broadcast program","currency", "stock exchange", "algorithm", "programming language", "transit system", "transit line","system", **"noble title"**

Fig. 4: Hierarchical taxonomy based on FIGER tag set. Fine-grained labels different from original FIGER taxonomy are marked in bold on tail.

include adding more QA-related concepts such as *fictional character* and change labels with ambiguity such as *title*, which may refer to a wide range of concepts. Other modifications are based on the joint consideration of question schema and entity. Take label *god* as an example, in the original FIGER ontology, *god* belong to the coarse class *other*. However, most of the questions related to the god figures are asked by question begin with *Who* which is more in accordance with the human-related question, thus we adjusted it under the coarse *Person* label.

Referencing the size of existing labeled QA benchmarks TREC-10, through one-shot model Bart-large[3] we build up a comparable dataset of 5596 questions, tagged by 8 labels according to the coarse labels in FIGER ontology. To ensure accuracy, all results are human-calibrated. Each sample in the dataset set includes the question text, positive entity with description, and a type label.

[3] https://huggingface.co/facebook/bart-large-mnli

Questions are sampled from the datasets of TriviaQA and QBLink, thus the labels are able to reflect the variety of question topics. A statistical summary is displayed in table 1. Eight coarse labels can be group into 2 camps: Four majority labels are Person, Location, Art, and Other. Question regarding *Person* is the most prevailing type. The second place comes with the label *Other*, which consists of more than 30 varied kinds of concepts such as animal, chemical elements, etc. The third place comes with the label *Location*, where questions are mostly related to geography concepts such as countries and cities. The last majority one is *Art*, where questions are asking about written works, paintings, and music, etc. All together these four classes account for more than 88% of total questions. The second row's labels all combined is less than the last member's share in the majority group. We adopt this very simple FIGER ontology to explain why entities are classified into certain groups. Besides, due to the skewed label distribution, we have to select reasonable metrics for the classification evaluation on minority labels.

Person	Others	Location	Art
2012	1225	961	781
Organization	Event	Product	Building
261	218	86	52

Table 1: Labelled QA dataset by coarse labels in FIGER, 5596 samples in total

3.3 Entity Description Classifier

Entity classification is a task commonly used for evaluating a language model's representation quality: High entity classification accuracy indicates the better knowledge mastered by the model. Like many NLP tasks, ambiguity is an issue in entity classification. For instance, given Jordan in a sentence, it could either refer to a western Asian country or the family name of the basketball player

Michael Jordan, depending on the context. To overcome the issue, newly proposed embedding models like BERT adopted contextual representation, which took into the consideration of surrounding context. This is a major improvement compared to the statistic embedding method such as word2vec[29]. In order to reduce the ambiguity caused by entity tokens, recent work pointed out the benefits of performing classification over entity description instead of entity tokens[42]. This is plausible since even two entities that share identical tokens would have different explanations. Using description could reduce the ambiguity, also improve the classifier's robustness.

The entity description classification task can be viewed as a special case of text classification, though summarizing the description. Our pipeline aims to generalize each entity a label from its description. To this end, we follow the methodology using BERT Link Prediction(BLP) [7] to train a fine-tuned entity description encoder that specialized in discovering the relationship between entity description with defined labels. The goal is to improve the quality of the entity representation, and furthermore, the classification accuracy. We use fine-tuned description encoder to generate embedding for an entity's description. Afterward, the classifier is performed by a Logistic Regression, using generated embedding as the feature. Normally the encoder can be undertaken by the pre-trained language models such as BERT-base. However as we discovered previously from the zero-shot classification, the attention module could mislead by the noisy detail in the gloss sentence, thus it is necessary to introduce a specialized fine-tuned BERT encoder.

The procedure is introduced by fig. 5, which includes two steps, namely fine-tuned encoder training and entity label classification. The training is conducted in the following way: Given a knowledge graph $(\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{D})$, where $\mathcal{D}$ represents the description sentence of entity $\mathcal{E}$. Beforehand, a mapping between entity $\mathcal{E}$ description $\mathcal{D}$ is built, and the description's embedding $f_\theta(d_{ei}) = (w_1, ..., w_n)$ is used as representation for e_i . The goal of the training process is to obtain a set of parameters θ for a new fine-tuned BERT encoder f, which is used to

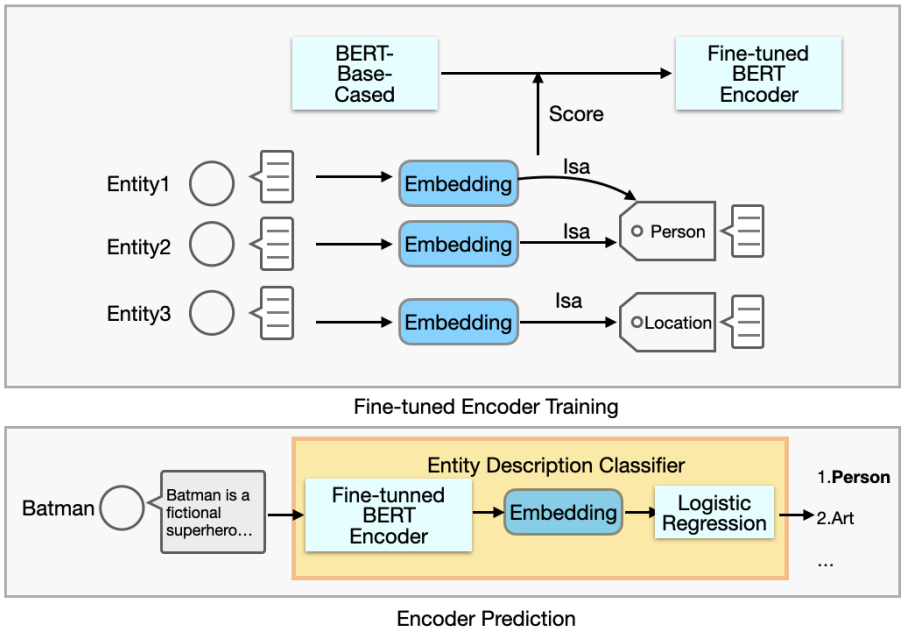


Fig. 5: Training and predicting an entity label by BERT Link Prediction

generate the embedding for the entity description. For each triple $(e_i, r_j, e_k) \in \mathcal{T}$, a positive score s_p is calculated according to the scoring function s. In addition, triple is corrupted to generate a negative sample by replacing either (e_i) or (e_k) with a random entity. The negative score is marked as s_n. The scoring processes are formulated in formula (3.1) and formula (3.2). During the training, positive triples shall be graded with high scores whereas corrupted negative triples with low scores. This is controlled by the scoring function s, and loss function $\mathcal{L}$, for which we used margin loss as displayed in formula (3.3). Finally, the encoder parameter θ is updated in one iteration, leveraging the calculated loss $\mathcal{L}$ and learning rate η, as elaborated in formula (3.4). Through adding regulation parameters, the model is able to obtain a generalized ability to predict the label for unseen entities, as long as the same type of entities share similar descriptions.

$$s_p = s(f_\theta(d_{ei}), r_j, f_\theta(d_{ek})) \tag{3.1}$$

$$s_n = s(f_\theta(d_{ei'}), r_j, f_\theta(d_{ek'})) \tag{3.2}$$

$$\mathcal{L} = max(0, 1 - s_p + s_n)) \tag{3.3}$$

$$\theta^{new} = \theta^{old} - \eta \bigtriangledown_\theta \mathcal{L} \tag{3.4}$$

As specified above, the scoring function s plays an important role in the training process, i.e. grant a high reward for positive samples while the low for the negative. In practice, this duty is fulfilled by the KGE methods, which are summarized in table 2. These KGE methods can be categorized by how scoring is conducted, i.e. translational model and tensor factorization model. As illustrated in fig. 6, the translational model embeds entities and relationships into the same vector space for representation. On the other hand, tensor factorization models take entities and relationship embedding as input to calculate a score through multiplication. Among three tensor factorization models, Distmult is the simplest one where the scoring function is continuous multiplication between matrices. ComplEx, on the basis of DistMult, extends the embedding from real space to complex space. Lastly, inspired by Canonical Polyadic, SimplE's scoring function is composed of two parts of matrix factorization.

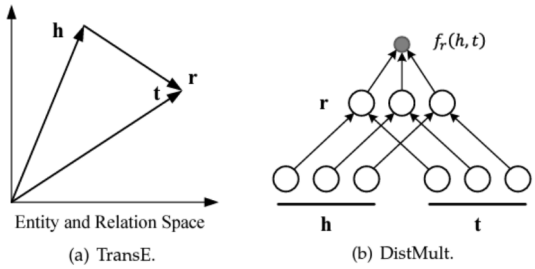


Fig. 6: Illustration of scoring process of translational and tensor factorization knowledge embedding models, adopt from [47],[31]

After obtaining embedding through a fine-tuned encoder, a multinominal Logistic Regression model is trained to predict the label. Taken the input embedding of entity description $\mathbf{Z}$, the possibility of each label i is generated through

Model	Type	Scoring function s
TransE [5]	Translational Model	$- \parallel \mathbf{h} + \mathbf{r} - \mathbf{t} \parallel$
DistMult[19]	Tensor Factorization	$\mathbf{h}^T diag(r)\mathbf{t}$
ComplEx[44]	Tensor Factorization	$Re(\mathbf{h}^T diag(r)\bar{t})$
SimplE[22]	Tensor Factorization	$\frac{1}{2}(h_{i1}^T diag(r_{j1})t_{k1} + h_{i2}^T diag(r_{j2})t_{k2})$

Table 2: Different scoring function given triple(h, r, t). $\parallel \cdot \parallel$ indicates n-norm, $Re(\cdot)$ indicates the real part of number, $\bar{t}$ is the complex conjugate of complex-valued vector t

the softmax function, where k represents the total classes.

$$Softmax(class = i|\mathbf{Z}) = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \; with \; i = 1, 2, ..., k \; and \; \mathbf{Z} = (z_1, ..., z_k)^T \in \mathbb{R}^k \tag{3.5}$$

During the fitting, a set of weight parameters are obtained for each classification label and the best regularization parameter λ is selected. Fine-tuned description encoder together with the logistic regression model formed our entity description classifier, which is marked in the orange box in fig. 5.

3.4 Neural Network Question Classifier

Determine the answer type is a common module in the traditional question answering system[30]. In the context of factoid question answering, since all questions are asking entities, Expected Answer Type(EAT) is used to represent the question-asked entity type like either person or location. For classifying EAT, we adopted the same FIGER taxonomy, which is compatible with the outcome of the previous entity classification.

With regards to question classification, we introduce two neural network models, namely CNN and LSTM. Due to the structure, CNN lacks the ability like RNN to catch the relationship between the long sequential terms. However, dealing specifically with the task of predicting EAT, the key is to focus on only a few terms related to the type instead of understanding sequential dependencies, which could easily mislead by the variety and noise in the question text. Based on the common knowledge that RNN models perform better with text format data, we also testify the popular RNN structure LSTM in our labeled QA dataset.

CNN Classifier As an illustration, the mechanism of CNN text classification is displayed in fig. 7. To apply CNN for the task of question classification, the initial step is to convert words into embeddings. In contrast to the computer vision tasks where kernels are applied on the pixels, here convolution is performed on either character level or token level. In the example displayed in the figure, kernel sizes varied from 2-4 on the token level to walk through the text and capture the useful information regarding EAT, this is formally presented in formula (3.6), where k indicates the size of the window. Here, K is a number list $[k_1, ...k_n]$,which considers both kernel length and list size. This is a hyperparameter that needs to be defined manually depending on performance. Afterward, parallel outputs of different kernels are passed to the following max-pooling layer, as displayed in formula (3.7). In the next step of formula (3.8), the results of each kernel's max-pooling are concatenated together. Lastly, the hidden states are passed through a fully connected layer, of which the output size is identical to the types of classification, and finally, a softmax function is applied to generate the most likely label, which is summarized in the formula (3.9) and (3.10).

$$X_{convulation\_k} = Conv(X_{Embedding}), k \in K \tag{3.6}$$

$$X_{maxpooling\_k} = Maxpool(X_{Conv_k}), k \in K \tag{3.7}$$

$$X_{maxpooling\_all} = Stack(X_{maxpooling\_k}), k \in K \tag{3.8}$$

$$X_{linear} = LinearLayer(X_{maxpooling\_all}) \tag{3.9}$$

$$X_{probability} = Softmax(X_{linear}) \tag{3.10}$$

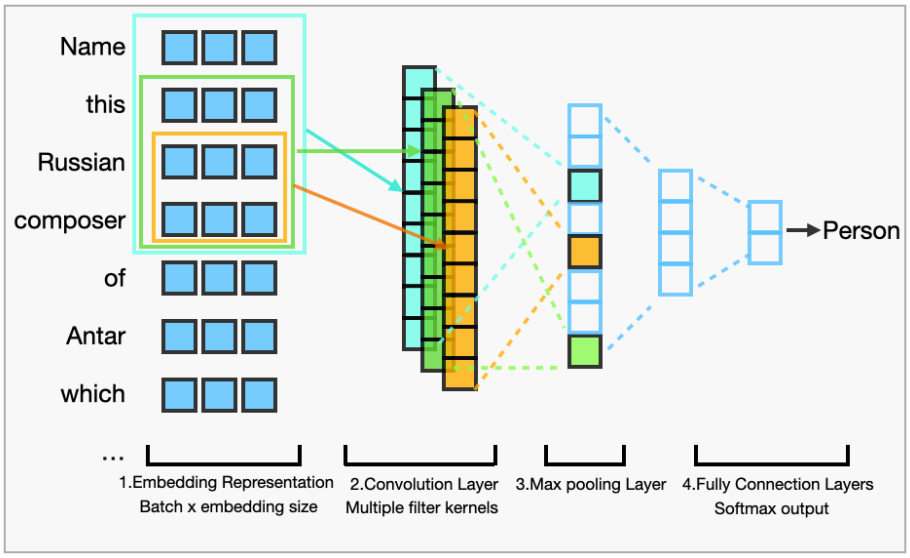


Fig. 7: CNN classifier structure

Long Short Term Memory Classifier We pick the representative Long Short Term Memory(LSTM) model for predicting each question's EAT. In the neural network, several LSTM layers are stacked together to capture the useful information passed from the embedding. In fig. 8 we displayed with two LSTM layers, formulated in equation (3.11), where K is the number of layers. To improve the ability of generalization, the dropout function is applied for each LSTM module as regulation, which is presented in equation (3.12). To fully exploit the text, we adopt LSTM in a bidirectional setting, i.e. not only sequence from front to end but also end to the front are used as input. In contrast to CNN, the output of the last LSTM layer is passed through a multi-head attention layer to weigh differently on each token, which is formalized in equation (3.13). Afterward, a fully connected linear layer cutting the dimension to the number of classification labels, and a softmax function is applied to generate the probability for each

potential label, as stated in equation (3.14).

$$X_{lstm\_n} = BiLSTM_k(X_{Embedding_{n-1}}), k \in K \tag{3.11}$$

$$X_{lstm\_n} = Dropout(X_{lstm\_n}) \tag{3.12}$$

$$X_{multi\_head} = Attention\_multi\_head(X_{lstm\_n}) \tag{3.13}$$

$$X_{probability} = Softmax(X_{linear(X_{multi\_head})}) \tag{3.14}$$

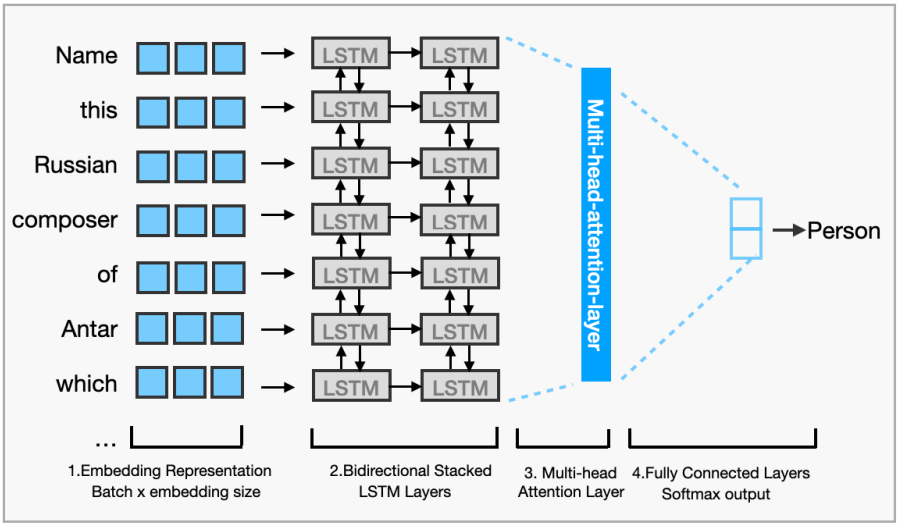


Fig. 8: LSTM classifier structure

3.5 Graph Pruning

Previously we introduced our methods on obtaining the classification models, trained from the labeled dataset. The pruning process is a trade-off between graph size and nodes quality: Stricter rules lead to a smaller graph with fewer entities, however, taking the risk that positive entity is also eliminated due to the unmatched result. Since most of the questions and entities are not labeled, we transfer the trained classifiers to perform label prediction on the entire free-text

KG, tagging each question and entity. Next, we prune the free-text KG according to these labels in a straightforward way: Only candidates with the same label as questions' EAT remain in the pruned graph. The generated pruned graph will be used as the evidence input for GNN.

3.6 Luke Language Representation Model

Besides externally modify DELFT's input considered by RQ1, for RQ2, we examine the language model's internal effect on DELFT, This hypothesis is based on the observation that even following the method of free-text KG which guaranteed a high volume of evidence from the text, it may still missing gold evidence for answering the question. In such circumstances, the model is relying on its own common knowledge instead of the given evidence to choose an answer.

Besides the original DELFT with popular BERT as language embedding, we integrated LUKE[48], a contextualized representation model based on RoBERTa[28], with DELFT. The key mechanism in LUKE is the entity self-attention during the pre-training. This is a natural extension from the BERT's masked language modeling, that is, predicting both masked word and entity in the masked sentence. As entities may be composed of several tokens, predicting the entity is a harder task than predicting a token, e.g. predicting *United States* instead of given *United* predicting the next token. This mechanism improves the LUKE's representation quality on entities.

3.7 Free-text Graph Representation and Reasoning

In the last step where the answer is derived from the input evidence graph, a GNN model named DELFT has been used as the reasoner. Following the schema of free-text KG, the main components of the graph are question, edge, and entities. In order to perform reasoning, the initial step is to properly represent the graph, which includes question representation, edge representation, and node representation, respectively.

Node Representation Both questions and nodes are formulated by a sequence of tokens in free-text KG. First, we convert either question or description sequence X into embedding E according to different language embedding model f_θ as formula (3.15) presented. Afterward, the generated Embedding $X(E_{x_1}, ...E_{x_n})$ is input into an RNN where a hidden state h_{x_u} is generated according to formula (3.16), u refers to the token's position in the sequence. Beyond RNN's output, a self-attention layer is applied to weight over the hidden states, as displayed in formula (3.17), where w_x is a learnable parameter from the training. Lastly, the output is the weighted average of all hidden states by formula (3.18), and in order to build the connection between the question and candidate entity, the candidate entity representation is the sum of question representation $h_q^{(0)}$ and its gloss representation $h_g^{(0)}$, as elaborated in formula (3.19).

$$(E_{x_1}, ...E_{x_n}) = f_\theta(X_1, ..., X_n) \tag{3.15}$$

$$h_{x_u} = RNN(E_{x_1}, ...E_{x_n}) \tag{3.16}$$

$$a_{x_u} = Softmax(w_x, h_{x_u}) \tag{3.17}$$

$$h_x^{(0)} = \sum_u a_{x_u} h_{x_u} \tag{3.18}$$

$$h_v^{(0)} = h_q^{(0)} + h_g^{(0)} \tag{3.19}$$

Edge Representation The edge representation's formulation is similar to the node representation, where the difference is that there might be multiple evidence sentences on a single edge. This indicates strong evidence that the candidate is related to the question and likely to be an answer. Also, in order to emphasize the relationship between question and evidence sentence, an inter attention layer replaced the self-attention layer in the node representation as displayed in the formula (3.21). After applying the attention module between question and edge sentence, the final edge representation is the average of all k sentence's hidden representation, as stated in formula (3.22) and (3.23).

$$h_{s_u} = RNN(E_{s_1}, ...E_{s_n}) \tag{3.20}$$

$$a_{s_u} = Softmax(h_{s_u}, h_q^{(0)}) \tag{3.21}$$

$$h_s^{(0)}(k) = \sum_u a_{s_u} h_{s_u} \tag{3.22}$$

$$h_e^{(0)} = Avg_k(h_s^{(0)}(k)) \tag{3.23}$$

Graph Update After converting the free-text KG from text to numerical representation, we have question representation $h_q^{(0)}$, node representation $h_v^{(0)}$ and edge representation $h_e^{(0)}$. The goal of DELFT is to do a series of manipulations to calculate the correct answer based on these representations. For each layer l in GNN, sequences like questions and entities are updated through Feed-Forward Network(FFN). Then evidence edge is scored by relevance, passed from the question entity to the candidate entity, and the candidate representation is updated.

1. Question Sentence Update: Question sentence is updated through a FFN, i.e. the next layer's question representation is the output of previous layer's FFN, as stated in formula (3.24).
2. Question Entity Update: Question entity is updated by combining question representation $h_q^{(l)}$ with previous layer's question entity representation $h_{v_q}^{(l-1)}$ into a FFN, as stated in formula (3.25).
3. Evidence Edge Update: Similar to question update, the evidence edge also updated through FFN. Since there might multiple edges link between question entity and candidate entity, DELFT take average of all edges as final edge representation, as stated in formula (3.26) and (3.27)
4. Edge Scoring: In order to emphasize the quality of the evidence respect to the question, an edge score $a_e^{(l)}$ is calculated based on the similarity between question and evidence, as stated by formula (3.28)
5. Information Passing: The final step before updating the candidate entity is to fuse previous representation to make dimension consistent. This is implemented by concatenate edge representation $h_e^{(l)}$ and question representation $h_{v_q}^{(l)}$, through FFN and weighted by edge score $a_e^{(l)}$, as stated by formula (3.29).

6. Candidate Entity Update: The candidate entities for answer are updated by combining previous layer's representation, question representation and Information Passing together, as stated by formula (3.30)
7. Answer Score: After multi layers of transformation, the answer is score is generated by using the final layer L's layer into a Multi Layer Perception(MLP) layer, of which the entity with the highest score is the predicted answer, as shown by formula (3.31).

$$h_q^{(l)} = FFN(h_q^{(l-1)}) \tag{3.24}$$

$$h_{v_q}^{(l)} = FFN(h_{v_q}^{(l-1)} + h_q^{(l)}) \tag{3.25}$$

$$h_s^{(l)}(k) = FFN(h_s^{(l-1)}(k)) \tag{3.26}$$

$$h_e^{(l)} = Avg_k(h_s^{(l)}(k)) \tag{3.27}$$

$$a_e^{(l)} = Sigmoid(h_q^{(l)} \cdot h_e^{(l)}) \tag{3.28}$$

$$f^{(l)}(v_q \xrightarrow{e} v_a) = a_e^{(l)} FFN([h_{v_q}^{(l)}; h_e^{(l)}]) \tag{3.29}$$

$$h_{v_a}^{(l)} = (h_{v_a}^{(l-1)} + h_q^{(l)} + \underbrace{\sum_{v_q \in V_q} f^{(l)}(v_q \xrightarrow{e} v_a)}_{InformationPassing}) \tag{3.30}$$

$$p(v_a; G) = Sigmoid\left(MLP(h_{v_a}^{(L)})\right) {v_a}^{(l)} \tag{3.31}$$

In summary, this chapter presented methods we adopt in the proposed pipeline. We applied zero-shot classification on free-text KG to build up a labeled dataset as preparation. We presented two types of methods, namely language model-based and neural network-based classifiers on the two variants of sentence classification tasks to derive labels. We proposed DELFT-LUKE, which integrates DELFT with an entity-oriented language model and lastly we elaborated how DELFT's reasoning is processed on the basis of free-text KG.

4 Question and Entity Classification

In this chapter, we introduce text classification with two types of models, namely neural network classifiers and language model-based classifiers to derive labels from free-text KG. For question EAT classification, two neural networks, i.e. CNN and LSTM are adopted in the experiment. In addition, two BERT-based models, namely BERT-base classifier and BERT-fine-tuned classifier are adopted for comparison. As for entity description classification, we only compare the performance between two BERT alike models. After question and entity classification, a statistical study is performed to discover how many questions' correct answers are falsely discarded during the process, and how many irrelevant candidate entities and edges are excluded as result.

4.1 Dataset

We use the labeled QA dataset constructed in section 3.1 for evaluating both classification tasks. The questions with gold labels are used for training the question classifier whereas the entity description text with gold labels for the description classification. 5596 samples are divided by train, valid, and test set according to the ratio of 6:2:2. To meet the issue of unbalanced sub-classes indicated by the statistics in table 1, we also apply the ratio of train-dev-test split over 8 sub-class labels to make each set's formation with more variety. Other textual modifications include (1) repeated entities are removed from the dataset and (2) description file is further processed to discard symbols and non-English content.

4.2 Implementation

The neural network models are implemented through the library of PyTorch, which mainly help with building the neural network structure for both CNN and LSTM as well as conducting the training, the parameters of both models can be found in the appendix. The language model classifiers are implemented

on the basis of BLP [7]. We trained two separate fine-tuned BERT models for each task. The fine-tuned BERT encoder is trained on the basis of BERT-base-uncased. During the training, firstly the labeled QA dataset is converted into a simplified KG, where the only type of triple is either *entity isa label* or *question isa label* depending on the task. After obtaining the fine-tuned BERT model for the purpose of question or entity classification, we apply fine-tuned model to encode either question or description text. The generated embeddings are used for fitting two logistic regression models for each classification task.

4.3 Metrics

We use accuracy as the main metric, that is, the number of truly predicted labels divides all its predictions. Since the labeled dataset's sub-class distribution is rather skewed, in order to reflect the model's generalization ability over all classes instead of focus on majority labels, we adopt balanced accuracy as another, which is the average of each sub-class accuracy. The formal definition is given in the following formulas (4.1) and (4.2), where K represents the number of classes.

$$Accuracy = \sum \frac{Truly\ predicted\ labels}{All\ predicted\ labels} \tag{4.1}$$

$$Balanced\_Accuracy = Avg(\sum_{k \in K} Accuracy_k) \tag{4.2}$$

4.4 Result

Entity Description Classification table 3 summarised the entity description classification by different scoring functions. Comparing the BERT base model with fined-tuned models, the fine-tuned models outperform the base model with a margin over 10% on accuracy. When it comes to balanced accuracy, the difference is enlarged to over 20%. Besides, fine-tuned models' balanced accuracy is close to the general accuracy. This indicates the fine-tuned models are better at summarizing labels from descriptions with inadequate samples.

	Train	Test	Balanced-Train	Balanced-Test
Bert-base-cased	0.829	0.799	0.715	0.634
BLP-DistMult	**0.958**	0.956	0.926	0.913
BLP-TransE	0.953	0.942	0.908	0.860
BLP-ComplEx	**0.958**	0.951	**0.930**	**0.918**
BLP-Simple	0.957	**0.957**	0.923	0.907

Table 3: Result of entity description classifiers based on different scoring method

	Train	Test	Balanced-Train	Balanced-Test
CNN kernels	0.823	0.691	0.534	0.404
BiLSTM	0.939	0.636	0.802	0.382
Bert-base-cased	0.742	0.740	0.518	0.506
BLP-ComplEx	**0.958**	**0.897**	**0.845**	**0.648**

Table 4: Result of question text classification on EAT

Question Classification For question classification, given a question we would like to figure out EAT that question is asking. Compare to the previous classifying entity with description, this task is harder since it engages more understanding of question's intention. We compared two camps of methods, i.e. neural network based methods and the language model methods. The result is displayed in table 4. Overall, neural network methods could achieve high accuracy during the training, however this may lead by overfitting as the BERT alike methods outperform the neural networks on the test set. Looking at two neural network models, CNN based model achieved better performance compared to the BiLSTM, which is the generally thought well-performance model on the text tasks. This result emphasized the structure's great impact on neural networks' performance on different tasks. In the setting of question EAT classification,

few keywords and specific patterns are more important than remembering the long-term dependency in the text.

Comparing two BERT-based models, again the model trained through BLP outperforms the base model with a wide margin, on both accuracy and balanced accuracy. The fine-tuned BERT encoder guided by the link prediction process could successfully build up the relationship between the question's text and EAT. Comparing the best-performed BLP-ComplEx with the previous work of entity description classification, the accuracy decreased by 5%, while the balanced accuracy is significantly lower than the accuracy and previous experiment's counterpart, which reflected the task difficulty to some extent.

4.5 Label Based Graph Pruning

In the second module of the pipeline, we conducted training on both question and entity description classification, obtained two corresponding classifiers. From the previous results, BLP-ComplEx based classifier achieved the best performance on both tasks. In the next pruning procedure, we apply best-performed classifiers over the entire free-text KG to examine the classifier's generalization ability. We prune the free-text KG according to both classifier's output by the end.

To set up prediction, first, we converted the free-text graph's entities into a simple knowledge graph the same way we constructed it for training. The only difference is questions and entities are labeled with null instead of the gold label as we previously owned in the labeled dataset. Afterward, we generated each question and entity a label with respect to its sentence. For the pruning process, we concern about both effectiveness and efficiency: (1) Whether the positive answer still exists in the graph and (2) how many irrelevant nodes and edges are excluded from the free-text KG.

The result is summarized in table 5. From the first three rows, we inspect the effectiveness of the pruning. The result shows that dataset QBLink has generally higher matched results compare to the TriviaQA. Since both datasets share the same format of entity description, thus this indicates the trained question clas-

	TriviaQA			QBLink		
	Train	Dev	Test	Train	Dev	Test
Question number	38689	4009	5159	40782	2861	5243
Correct matched question	30862	3207	4693	34403	2533	4505
Correct matched percentage	79.89%	80%	90.1%	84.4%	88.5%	85.9%
Candidates per question before	19.6	49.6	49.8	19	45.8	45.4
Candidates per question after	10.7	18.9	18.6	11.1	30.2	11.4
Percentage entities excluded	45.4%	61.9%	62.7%	41.6%	34.1%	74.9%
Evidence edges per question before	113	301	302	199	351	354
Evidence edges per question after	64	120	118	109	233	85
Percentage edge excluded	43.4%	60%	59.9%	45.2%	33.6%	76%

Table 5: Pruning result by matching the question's EAT and entity's label

sifier performs better with the longer and richer context in QBLink, compared to short and plain questions in TriviaQA.

Secondly, we focus on the efficiency of the pruning, which includes numbers of entities in the free-text KG, stated from table 5's row4 to row6. Before processing, the free-text KG training set's candidate entities are limited up to 20 in compromised of the graph size, while the validation and test set's candidates' boundaries are up to 50. Looking at both datasets' training sets, by introducing entity type as a general property, the pruning process is able to eradicate over 40% of unmatched candidates, while for validation and test set the eradicated percentage is up to over 60% because of the larger initial candidate set. Lastly, looking at the evidence edges in the graph, the effectiveness is in accordance with the eliminated candidates. This can be interpreted from most candidates entities have 1 to 2 connected edges to the question entities.

In summary, after semantically matching EAT with entity labels, we trade-off a relatively small percentage of correct entities for reducing the graph size by half. As a reward, when DELFT is applied to perform reasoning, the model would have a higher probability of choosing the correct answer from the shrunken candidates,

though also a number of positive entities excluded at the same time. Moreover, labels provide us an intermediate step in the QA process to evaluate why the answering succeed or failed. The unmatched result between EAT and entity label indicates the model's lack of understanding of the question's intention. The pruned graph will be used as input for the next chapter to evaluate whether the pruning process is able to improve the QA benchmark, or only enhanced the QA process's explainability.

5 Question Answering Validation

In this chapter, we verify our two hypotheses by applying DELFT in 2 experiments. For RQ1 which focuses on external modification on free-text KG, we compare DELFT-BERT's performance between the original graph and pruned graph. For RQ2 which focuses on the DELFT's internal representation, we compare the proposed DELFT-LUKE with other DELFT versions to testify whether the entity-based language representation model benefited DELFT.

5.1 Dataset

We evaluate the DELFT with two QA benchmarks, namely TriviaQA[20] and QBLink[13]. As its name indicates, TriviaQA mainly adopts questions from quiz games. The questions are rather short and straightforward, thus fewer entities are included in the question text as supporting evidence. On the contrary, QBLink is originally a sequential QA benchmark, thus the question is composed of multiple sentences, which provides diverse details as descriptions. As a previous statistical study shows [54], 87% of TriviaQA's questions have less than 3 entities while more than 90% of QBLink's questions have more than 3 entities. Though the question in TriviaQA is shorter and generally easier, however, due to the lack of evidence, in the experiments setting it could be harder compared to the QBLink.

5.2 DELFT over Pruned Graph

To study research1 we compare the result by input both original and pruned free-text KG into the DELFT QA model. In this experimental setting, we adopt DELFT-BERT for evaluation in accordance with the previous BLP based classifier. We testify the DELFT-BERT on the test set of two datasets. For the pruned free-text KG, we remained the positive entity in the graph for the sake of verification. However, some positive answers in the pruned graph should have been already excluded due to previous matching failure, we exclude such kinds of correct answered questions and adjust the metrics accordingly.

The QA result is summarized in table 6. Overall, our pruning method does improve the DELFT's performance with a clear margin. Looking horizontally, before adjusting the accuracy due to the unmatched label, there is an 11% and 22% improvement for TriviaQA and QBLink, respectively. After we adjusted accuracy for the pruned graph to exclude discarded positive answers in pruning, the improvements are narrowed down by 6% and 12%. Beyond metrics, following the gap between raw and adjusted accuracy gave us more insights on how many percentages of questions are failed due to inability in DELFT's reasoning process, and how many percentages of questions are due to insufficient understanding of question and entity at the very beginning.

	TriviaQA		QBLink	
	Original	Pruned	Original	Pruned
Question Numbers	5159		5243	
Correct Answers	3103	3679	3300	4438
Correct After Exclusion	-	3329	-	3812
Raw Accuracy	60.15%	71.31%	62.94%	84.65%
Adjusted Accuracy	-	**64.53%**	-	**72.71%**
Improvement	+4.43%		+9.77%	

Table 6: DELFT-BERT's performance on the test set of TriviaQA and QBLink, comparing between original free-text KG and pruned free-text KG

5.3 DELFT by Different Language Model

RQ2 wondering whether the adjustment in the language model's pretraining would benefit DELFT. To do so, we integrated DELFT-LUKE, which uses the pre-trained language model LUKE -base[4] to generate the embedding for elements in free-text KG.

[4] https://huggingface.co/studio-ousia/luke-base

In this experiment, we compared the performance of DELFT-LUKE with the other two DELFT versions, namely DELFT-BERT and DELFT-Glove on both datasets' dev and test set. The result is summarized in table 7. Among 3 language models, DELFT-BERT achieved the best accuracy on 3 out of 4 terms. DELFT-Glove scored closely to the DELFT-BERT. The proposed DELFT-LUKE is lagged behind the previous two versions. Looking at the results within DELFT-LUKE, like the other two versions, DELFT-LUKE's performance on QBLink is better than TriviaQA, with a larger margin. There may several reasons why

	TriviaQA		QBLink	
	Dev	Test	Dev	Test
DELFT-Glove	58.60%	59.43%	**64.17%**	61.93%
DELFT-BERT	**59.64%**	**60.15%**	63.82%	**62.94%**
DELFT-LUKE	52.66%	53.17%	61.65%	59.26%

Table 7: The performance of DELFT integrating with different language models on dataset TriviaQA and QBLink. Best performance is marked with bold

DELFT-LUKE's performance does not meet the expectation. First, due to the GPU size limit, we are not able to utilize the LUKE-large as the pretraining model and thus adopt LUKE-base. The parameter in LUKE-large is double times bigger than the LUKE-base, in other words, has better performance. Second, similar to DELFT-BERT, DELFT-LUKE needs tremendous time to train for computing the contextual representation. To compromise, we cut down the training epochs from 5 to 2, which may indicate insufficient training. We added the loss curve's plot in the appendix. Lastly, LUKE was specialized in entity-orientated tasks, such as Named Entity Recognition and Entity Classification. In the hypothesis, we are hoping the model would be able to benefit from this characteristic and perform well when there is no gold evidence available. However like several previous models, LUKE may also struggle with the enormous

amount of noise in the text, which undermined its performance. This claim can be supported from two aspects. Firstly, the result shows that if we also provide noise-reduced pruned free-text KG to DELFT-LUKE on QBLink-test, the performance could improve by 10.1%. Secondly, the above experiment showed that DELFT-LUKE performed better on the QBLink with more available questions entities, compared to TriviaQA, which lack supporting entities.

In summary, for QA we adopt DELFT performed two experiments to testify our research questions. As the result, we further improved DELFT's performance with our pruning methods while integrating an entity-focused language model did not. This may indicate that despite powerful ability, the contextual language model is still not a silver bullet for complex reasoning tasks. However, there is room for improvement on GNN by providing a smaller and better quality graph.

6 Results Analysis

6.1 Question and Entity Description Classification

In Chapter 4, leveraging BLP our best classifier is able to successfully match 80% question with respect to EAT. In the following section, we focus on giving explanations on why matching between question and entity based on label could be failed.

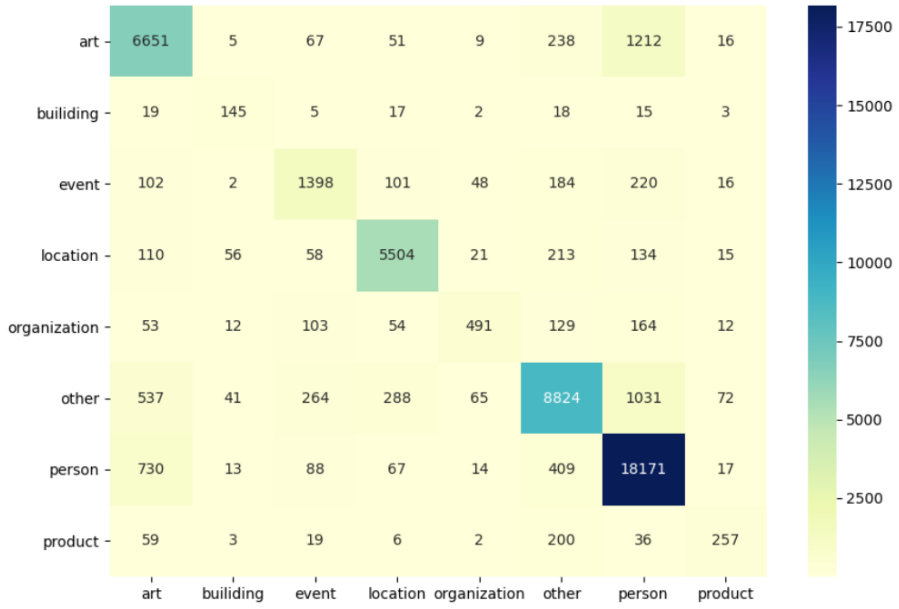


Fig. 9: Confusion matrix on QBLink's questions and entities' predicted labels

We study the output from the confusion matrix between the predicted question label and the desired positive entity's label, which is represented in the fig. 9. In the plot, the x-axis represents the positive entity's predicted label while the y-axis represents the predicted EAT label. First of all, numbers on the diagonal line suggest the correct matching dominate in every subclass. This indicates the majority of labels can be successfully matched. Looking at the four prevailing

labels with over 5000 samples per each, namely art, location, other, and person. Label location has the least number of unmatched pairs, while label person has the most. Since location has more questions compared to label art, thus this result indicates the QA model is likely to be well performed on geography questions due to its certainty and clear intention of questions.

There are several patterns that can be concluded from the confusion matrix. Firstly, the unmatched labels between label art and person is a major issue with more than 1000 failures. This problem can be explained from both sides: A question regarding a person usually includes its work as an indicator, while an artwork's gloss sentence commonly mentions its creator by convention. Secondly, for label other, it confuses with all the others at the same time, proportional to each label's amount. This can be inferred from two aspects that (1) other label includes more than 30 concepts according to the FIGER taxonomy, thus classifier didn't figure out semantic features. (2) Compare to some types of questions with the obvious patterns, e.g. *Where* to geography questions and *Who* to questions regarding the person, the questions to type other has the most variety due to the content. These difficulties lead to relatively poor performance in QA.

The other group is consists of 4 minority labels, which are building, event, organization, and product. Mismatching patterns can be discovered semantically, such as the label building's biggest mismatch is with label location, which makes sense since the boundary between two labels could be very narrow sometimes. The same happens to the mismatch between label product and other. In addition, label event and label organization are often mismatched. This is similar to the above issue between label person and art, confused by what is subject and what is indicator in either question or a description sentence.

6.2 QA Result by Topics

Previous work mainly focuses on the characteristics of data's influence on the QA result. For instance, DELFT perform better with larger number of question entities and supporting evidence sentences. Leveraged by the label we introduced,

we are able to gain insight from a semantic view. For some QA topics, the question is formed in a more directional way, and supporting evidence can be better understood by the model. From the above confusion matrix, we clearly see the matching perform less well on abstract questions in label group other. Besides general accuracy, we also studied the accuracy by each subclass. Since the classification conducted by the BLP is not perfectly reliable and topic may not correctly labeled, we used the label of each positive answer in following displays.

	Art	Person	Location	Other	Organization	Event	Building	Product
	383	1100	945	599	164	85	18	35
TriviaQA	559	1580	1441	1055	263	152	40	69
	68,52%	69,62%	65,58%	56,78%	62,36%	55,92%	45,00%	50,72%
	638	1723	545	671	49	134	20	32
QBLink	883	2098	714	1144	114	202	32	56
	72,25%	82,13%	76,33%	58,65%	42,98%	66,34%	62,50%	57,14%

Table 8: Correctly answered questions' percentage according to different sub class label on TriviaQA and QBLink

The performance of each label is statistically summarized in table 8 and graphically illustrated by fig. 10. Firstly we examine the 4 major labels, jointly account for 91% of the total questions. For both datasets, only label other is largely behind the general accuracy. For minority labels, the accuracy is generally lower than the major counterparts. We have the following observations from the above phenomenon. First, adequate training examples do affect the performance, as the long-tail effect summarize. In our experiments person labeled question has the highest share of all question types and obtained best accuracy, while on the other hand, 4 minority labels are mostly behind the general accuracy. Secondly, the definition of the taxonomy does not affect the final result to a great extent.

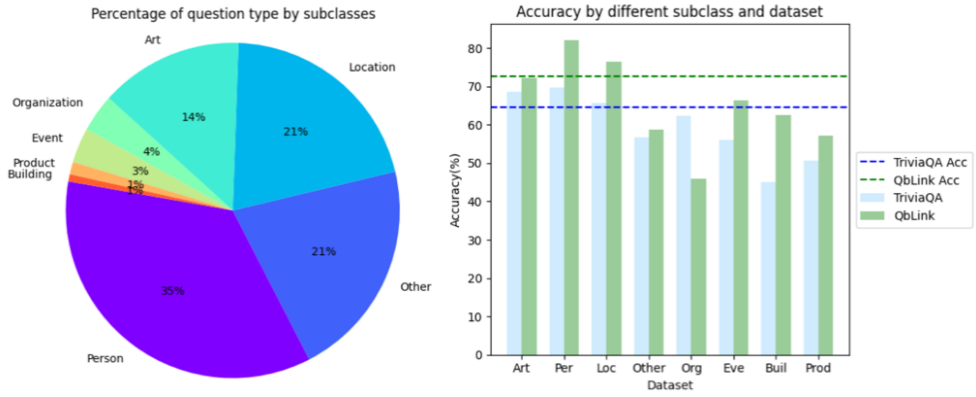


Fig. 10: Percentage of each type of question(Left) and accuracy for each label by dataset(Right)

If we treat 4 less popularized labels under a same label, the overall accuracy would be very close to label other. Lastly, concerning the size and semantic, the accuracy on moderate size of concepts, e.g. person and location are better compared to either too-big concept (other) and too small concept (building). Introducing an appropriate ontology for QA is not a trivial task, as the FIGER taxonomy we adopt was originally used for fine-grained classification.

6.3 Case Study

To find out the details beneath the general accuracy, we performed a case study on several specific questions. We selected 4 questions from the test set of QBLink and TriviaQA. Each of them belongs to a different type of question. The result is displayed in fig. 11. From examples (I) and (II), the pruning method is able to constrain top likely answers within the EAT that question asked, namely coutry and disease. Although the example (I) failed, the predicted answer is still reasonable with *Yugoslavia*: A divided European country, also engaged in WWII. Correctly referencing entities in question is essential for answering the question. For instance, example (III) successfully referred Civil War to Spanish Civil War, which is key evidence for answering the question. Example (I) went wrong largely

	Example	Explanation
I(-)	Name this European nation which was divided into Eastern and Western regions after World War II. Prediction: Yugoslavia Answer: Germany	By pruning the model is able to identify a location is being asked (All top 3 predictions are countries). However missing the key evidence Eastern and Western Regions are referring BDR and DDR, respectively.
II(+)	Name this disease in which patients are missing blood clotting factors, resulting in even small cuts being potentially fatal. This disease is most often caused by mutations in factor VIII and factor IX. Prediction:haemophilia Answer:haemophilia	The disease question is successfully classified under the coarse label. Also, top three predictions are all diseases and the correct answer is selected.
III(+)	Name this party that merged with the Monarchist factions after the Nationalist victory in its nation's Civil War. It was founded and initially led by Primo de Rivera. Prediction: FET y de las JONS Answer: FET y de las JONS	Spanish Civil War is successfully referred by the entity in question. Key evidences for the connection between positive answer to both Spanish Civil War and party leader Primo de Rivera.
IV(-)	In May 1999, after over 20 years of restoration work, which painting by Leonardo da Vinci was placed back on display in Milan?" Prediction:The Last Supper Answer:The Last Supper	Question is correctly answered, however, labels are unmatched. The positive entity is successfully marked with label Art while the questions has been falsely marked by label Other

Fig. 11: DELFT-BERT's prediction on questions from TriviaQA and QBLink. The correct answered questions are marked as(+) while the false answers are marked with(-). Explanations are given on details with respect to each question

because of failed to refer to eastern and western regions in the question correctly, instead, falsely pointed to the identical entity of the Western Region in China, which lead to the wrong answer Yumen Pass by original free-text KG. Lastly, question (IV) is correctly answered, however, the matching between question and positive entity is failed. In such a scenario, the positive answer would be very different from other candidates, which may explain why the accuracy is so high before we excluding the unmatched positive answer from the pruned graph.

7 Conclusion

This study is designed to improve and investigate the modifications' influence on QA system DELFT. To this end, we proposed a framework by externally pruning DELFT's input free-text KG, and a new DELFT version which internally integrated with an entity-oriented language model. Returning to the two research questions at the beginning, for the first question, we can state that introducing label as a general property into free-text KG and the further pruning do benefits the DELFT's performance. This is largely contributed by two classifiers, which are capable of understanding the intention of question and summarizing the label from entity description. For the second question regarding the effect of language model, the performance of DELFT-LUKE did not meet our expectation. Regarding the topic of QA by GNN, this project demonstrated the gaining of excluding irrelevant nodes in a graph overcame the risk that we may loose the positive answer. In the experiment, DELFT acquired significant improvement from the smaller and higher quality free-text KG despite the imperfect classification result.

The proposed framework of labeling a dataset through zero-shot classification and further training a classification model through BLP, demonstrated its potential. The general idea is similar to the encoder-decoder structure. In this study, facing the issue that there is no factoid QA-oriented labeled dataset available, we collected a dataset, which is used as ground standing for supervised training and encode the basic knowledge. In order to perform pruning between questions and entities in QA, we performed two classification tasks under the same taxonomy to justify the effectiveness of learning by classification. From the result, among several classification techniques, BLP based classier yield the best outcome on both tasks. BLP model's sufficient performance and generalization ability made the pruning process possible with a relatively high accuracy. Along the same method, the pipeline can also be applied to other similar NLP tasks such as sentiment analysis.

Multiple findings were revealed during the project. First, the attention module can be easily misled by adding some text noise in the sentence. This is implied from several aspects in the study, either the falsely predicted topic by zero-classification, or the BLP-classifier's wrong label. We tried to minimize this noise effect with a fine-tuned BERT model which was trained through a simple KG as guidance. In this way, our classifier obtained better performance compared to the neural network models, however, still room for improvement. Second, by error analysis, we discovered multiple patterns on how the QA process may fail. The semantic information summarized by the label gave us insight into the model's understanding intermediately, which provided us whether the model failed with understanding the question's intent, or was due to a lack of supporting evidence. Third, the performance of the newly introduced DELFT-LUKE suggests that the embedding model is not a direct solution for complex QA, while a high-quality evidence graph is, as the pruning methods indicated.

Finally, several limitations need to be taken into account. First of all, the size of the labeled dataset is considerably small, which has the problem that several rare labels are lack training samples. The effect can be found by the relatively low performance on each of these labeled questions. Another limitation comes from the pruning process, where we used simple but brutal direct matching. However, during the process, much information has been discarded, such as the possibility of the label. A method that takes both label's ranking and possibility may make pruning more robust. Lastly, we adopt label as the single and general property for both questions and entities in this study, future work may adopt an ontology that balance the variety and universality, further increase the QA process and result explainability.

References

1. Asai, A., Hashimoto, K., Hajishirzi, H., Socher, R., Xiong, C.: Learning to retrieve reasoning paths over wikipedia graph for question answering. CoRR **abs/1911.10470** (2019), http://arxiv.org/abs/1911.10470
2. Balakrishnan, S., Halevy, A., Harb, B., Lee, H., Madhavan, J., Rostamizadeh, A., Shen, W., Wilder, K., Wu, F., Yu, C.: Applying webtables in practice. In: CIDR (2015)
3. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: A collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data. p. 1247–1250. SIGMOD '08, Association for Computing Machinery, New York, NY, USA (2008). https://doi.org/10.1145/1376616.1376746, https://doi.org/10.1145/1376616.1376746
4. Bonifati, A., Martens, W., Timm, T.: Navigating the maze of wikidata query logs. The World Wide Web Conference (2019)
5. Bordes, A., Usunier, N., Garcia-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2. p. 2787–2795. NIPS'13, Curran Associates Inc., Red Hook, NY, USA (2013)
6. Chen, D., Fisch, A., Weston, J., Bordes, A.: Reading wikipedia to answer open-domain questions. In: ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers). pp. 1870–1879. Association for Computational Linguistics (ACL) (2017). https://doi.org/10.18653/v1/P17-1171
7. Daza, D., Cochez, M., Groth, P.: Inductive entity representations from text via link prediction. In: Proceedings of The Web Conference 2021 (2021). https://doi.org/10.1145/3442381.3450141
8. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR **abs/1810.04805** (2018), http://arxiv.org/abs/1810.04805
9. Diefenbach, D., Both, A., Singh, K.D., Maret, P.: Towards a question answering system over the semantic web. CoRR **abs/1803.00832** (2018), http://arxiv.org/abs/1803.00832

10. Ding, M., Zhou, C., Chen, Q., Yang, H., Tang, J.: Cognitive graph for multi-hop reading comprehension at scale. arXiv preprint arXiv:1905.05460 (2019)
11. Dogan, C., Dutra, A., Gara, A., Gemma, A., Shi, L., Sigamani, M., Walters, E.: Fine-grained named entity recognition using elmo and wikidata. CoRR **abs/1904.10503** (2019), http://arxiv.org/abs/1904.10503
12. Dong, X., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S., Zhang, W.: Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 601–610 (2014)
13. Elgohary, A., Zhao, C., Boyd-Graber, J.: A dataset and baselines for sequential open-domain question answering. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 1077–1083. Association for Computational Linguistics, Brussels, Belgium (Oct-Nov 2018). https://doi.org/10.18653/v1/D18-1134, https://aclanthology.org/D18-1134
14. Galkin, M., Trivedi, P., Maheshwari, G., Usbeck, R., Lehmann, J.: Message passing for hyper-relational knowledge graphs. arXiv preprint arXiv:2009.10847 (2020)
15. Graves, A., Schmidhuber, J.: Framewise phoneme classification with bidirectional lstm and other neural network architectures. Neural Networks **18**(5), 602–610 (2005). https://doi.org/https://doi.org/10.1016/j.neunet.2005.06.042, https://www.sciencedirect.com/science/article/pii/S0893608005001206, iJCNN 2005
16. Hao, Y., Zhang, Y., Liu, K., He, S., Liu, Z., Wu, H., Zhao, J.: An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 221–231. Association for Computational Linguistics, Vancouver, Canada (Jul 2017). https://doi.org/10.18653/v1/P17-1021, https://aclanthology.org/P17-1021
17. Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., de Melo, G., Gutiérrez, C., Gayo, J.E.L., Kirrane, S., Neumaier, S., Polleres, A., Navigli, R., Ngomo, A.N., Rashid, S.M., Rula, A., Schmelzeisen, L., Sequeda, J.F., Staab, S., Zimmermann, A.: Knowledge graphs. CoRR **abs/2003.02320** (2020), https://arxiv.org/abs/2003.02320
18. Huang, X., Zhang, J., Li, D., Li, P.: Knowledge graph embedding based question answering. In: Proceedings of the Twelfth ACM International Conference on Web

Search and Data Mining. p. 105–113. WSDM '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3289600.3290956, https://doi.org/10.1145/3289600.3290956

19. Huang, Z., Li, B., Yin, J.: Knowledge graph embedding via multiplicative interaction. In: Proceedings of the 2nd International Conference on Innovation in Artificial Intelligence. p. 138–142. ICIAI '18, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3194206.3194227, https://doi.org/10.1145/3194206.3194227
20. Joshi, M., Choi, E., Weld, D.S., Zettlemoyer, L.: Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. CoRR **abs/1705.03551** (2017), http://arxiv.org/abs/1705.03551
21. Kalchbrenner, N., Grefenstette, E., Blunsom, P.: A convolutional neural network for modelling sentences. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 655–665. Association for Computational Linguistics, Baltimore, Maryland (Jun 2014). https://doi.org/10.3115/v1/P14-1062, https://aclanthology.org/P14-1062
22. Kazemi, S.M., Poole, D.: Simple embedding for link prediction in knowledge graphs. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. p. 4289–4300. NIPS'18, Curran Associates Inc., Red Hook, NY, USA (2018)
23. Kim, Y.: Convolutional neural networks for sentence classification. CoRR **abs/1408.5882** (2014), http://arxiv.org/abs/1408.5882
24. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. CoRR **abs/1609.02907** (2016), http://arxiv.org/abs/1609.02907
25. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998). https://doi.org/10.1109/5.726791
26. Li, X., Roth, D.: Learning question classifiers. Proceedings of the 19th international conference on Computational linguistics - (2002). https://doi.org/10.3115/1072228.1072378
27. Ling, X., Weld, D.S.: Fine-grained entity recognition. In: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence. p. 94–100. AAAI'12, AAAI Press (2012)

28. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized BERT pretraining approach. CoRR **abs/1907.11692** (2019), http://arxiv.org/abs/1907.11692
29. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems. vol. 26. Curran Associates, Inc. (2013), https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf
30. Moldovan, D.I., Harabagiu, S.M., Girju, R., Morarescu, P., Lacatusu, V.F., Novischi, A., Badulescu, A., Bolohan, O.: Lcc tools for question answering. In: TREC (2002)
31. Nickel, M., Rosasco, L., Poggio, T.: Holographic embeddings of knowledge graphs. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. p. 1955–1961. AAAI'16, AAAI Press (2016)
32. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. CoRR **abs/1802.05365** (2018), http://arxiv.org/abs/1802.05365
33. Petroni, F., Rocktäschel, T., Lewis, P.S.H., Bakhtin, A., Wu, Y., Miller, A.H., Riedel, S.: Language models as knowledge bases? CoRR **abs/1909.01066** (2019), http://arxiv.org/abs/1909.01066
34. Poerner, N., Waltinger, U., Schütze, H.: E-bert: Efficient-yet-effective entity embeddings for bert. arXiv preprint arXiv:1911.03681 (2019)
35. Radford, A., Narasimhan, K.: Improving language understanding by generative pre-training (2018)
36. Ramachandra Rao, S.K.: Question Answering with Hybrid Data and Models. Theses, Université Paris-Saclay (Feb 2020), https://tel.archives-ouvertes.fr/tel-02890467
37. Razniewski, S., Balaraman, V., Nutt, W.: Doctoral advisor or medical condition: Towards entity-specific rankings of knowledge base properties [extended version]. CoRR **abs/1709.06907** (2017), http://arxiv.org/abs/1709.06907
38. Ren, H., Hu, W., Leskovec, J.: Query2box: Reasoning over knowledge graphs in vector space using box embeddings. CoRR **abs/2002.05969** (2020), https://arxiv.org/abs/2002.05969

39. Riedel, S., Yao, L., McCallum, A.: Modeling relations and their mentions without labeled text. In: Balcázar, J.L., Bonchi, F., Gionis, A., Sebag, M. (eds.) Machine Learning and Knowledge Discovery in Databases. pp. 148–163. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
40. Schuster, M., Paliwal, K.: Bidirectional recurrent neural networks. Trans. Sig. Proc. **45**(11), 2673–2681 (Nov 1997). https://doi.org/10.1109/78.650093, https://doi-org.vu-nl.idm.oclc.org/10.1109/78.650093
41. Shen, D., Lapata, M.: Using semantic roles to improve question answering. In: Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL). pp. 12–21 (2007)
42. Shi, B., Weninger, T.: Open-world knowledge graph completion. CoRR **abs/1711.03438** (2017), http://arxiv.org/abs/1711.03438
43. Singh, A.: Entity based Q&A retrieval. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. pp. 1266–1277. Association for Computational Linguistics, Jeju Island, Korea (Jul 2012), https://aclanthology.org/D12-1116
44. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. CoRR **abs/1606.06357** (2016), http://arxiv.org/abs/1606.06357
45. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks (2018)
46. Wang, H., Zhang, X., Ma, S., Sun, X., Wang, H., Wang, M.: A neural question answering model based on semi-structured tables. In: Proceedings of the 27th International Conference on Computational Linguistics. pp. 1941–1951. Association for Computational Linguistics, Santa Fe, New Mexico, USA (Aug 2018), https://aclanthology.org/C18-1165
47. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence. p. 1112–1119. AAAI'14, AAAI Press (2014)
48. Yamada, I., Asai, A., Shindo, H., Takeda, H., Matsumoto, Y.: LUKE: deep contextualized entity representations with entity-aware self-attention. CoRR **abs/2010.01057** (2020), https://arxiv.org/abs/2010.01057

49. Yamada, I., Asai, A., Shindo, H., Takeda, H., Takefuji, Y.: Wikipedia2vec: An optimized tool for learning embeddings of words and entities from wikipedia. CoRR **abs/1812.06280** (2018), http://arxiv.org/abs/1812.06280
50. Yamada, I., Tamaki, R., Shindo, H., Takefuji, Y.: Studio ousia's quiz bowl question answering system. CoRR **abs/1803.08652** (2018), http://arxiv.org/abs/1803.08652
51. Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W.W., Salakhutdinov, R., Manning, C.D.: Hotpotqa: A dataset for diverse, explainable multi-hop question answering. CoRR **abs/1809.09600** (2018), http://arxiv.org/abs/1809.09600
52. Zhang, X., Shou, L., Pei, J., Gong, M., Wen, L., Jiang, D.: A graph representation of semi-structured data for web question answering. CoRR **abs/2010.06801** (2020), https://arxiv.org/abs/2010.06801
53. Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., Liu, Q.: Ernie: Enhanced language representation with informative entities. arXiv preprint arXiv:1905.07129 (2019)
54. Zhao, C., Xiong, C., Qian, X., Boyd-Graber, J.: Complex factoid question answering with a free-text knowledge graph. In: Proceedings of The Web Conference 2020. pp. 1205–1216 (2020)
55. Zhao, C., Xiong, C., Rosset, C., Song, X., Bennett, P., Tiwary, S.: Transformer-xh: Multi-evidence reasoning with extra hop attention. In: International Conference on Learning Representations (2019)

Appendix

Layer	Description
Convolution Layer	Kernel size(2,3,4,5,6)
Maxpooling Layer	1 dimensional max-pooling
Fully Connected Layers	Composing hidden size from 500 to 8
Softmax	LogSoftmax, 8 possible labels

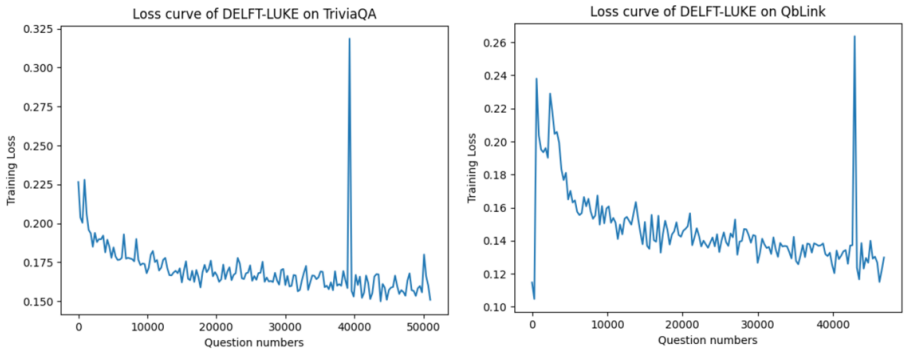
The graph structure and parameters for CNN model

Layer	Description
LSTM Layer1	bidirectional, 100->250
LSTM Layer2	bidirectional, 250->500
Attention Module	Multi-head attention, head=8
Fully Connected Layer	Composing hidden size from 500 to 8
Softmax	LogSoftmax, 8 possible labels

The graph structure and parameters for RNN model

Layer	Description
RNN	1-layer Bi-GRU, 300 dimension
FFN	600 dimension, ReLU activation
MLP	2 Layers from 600 ->300dimension, ReLU activation
Attention	600 dimension Bi-linear
Self-attention	600 dimension linear
Layers	3

The graph structure and parameters for GNN reasoner DELFT[54]



DELFT-LUKE's training loss curve on TriviaQA(Left) and QBLink(Right)