

Question Answering over Knowledge Base using Language Model Embeddings

Sai Sharath, Japa
School of Computing
Southern Illinois University
Carbondale, IL
sharath.japa@siu.edu

Banafsheh, Rekabdar
School of Computing
Southern Illinois University
Carbondale, IL
brekabdar@cs.siu.edu

Abstract—Knowledge Base, represents facts about the world, often in some form of subsumption ontology, rather than implicitly, embedded in procedural code, the way a conventional computer program does. While there is a rapid growth in knowledge bases, it poses a challenge of retrieving information from them. Knowledge Base Question Answering is one of the promising approaches for extracting substantial knowledge from Knowledge Bases. Unlike web search, Question Answering over a knowledge base gives accurate and concise results, provided that natural language questions can be understood and mapped precisely to an answer in the knowledge base. However, some of the existing embedding-based methods for knowledge base question answering systems ignore the subtle correlation between the question and the Knowledge Base (e.g., entity types, relation paths, and context) and suffer from the Out Of Vocabulary problem. In this paper, we focused on using a pre-trained language model for the Knowledge Base Question Answering task. Firstly, we used Bert base uncased for the initial experiments. We further fine-tuned these embeddings with a two-way attention mechanism from the knowledge base to the asked question and from the asked question to the knowledge base answer aspects. Our method is based on a simple Convolutional Neural Network architecture with a Multi-Head Attention mechanism to represent the asked question dynamically in multiple aspects. Our experimental results show the effectiveness and the superiority of the Bert pre-trained language model embeddings for question answering systems on knowledge bases over other well-known embedding methods.

Index Terms—knowledge base question answering, BERT, Language Model, KBQA, Multi-Head Attention

I. INTRODUCTION

Question Answering (QA) systems enable natural language platforms to interact with the Knowledge Bases. These QA systems return direct and more specific answers to the asked questions. In recent years, with the increase in the popularity and the use of knowledge bases (KB) such as Google’s Knowledge Graph [1], YAGO [2], DBpedia [3], and Freebase [4] people are more interested in seeking effective methods to access these knowledge bases. Most of such knowledge bases adopt Resource Description Frame-

work (RDF) as their data format and they also contain billions of SPO (subject, predicate, object) triples [5].

There are several languages designed for querying such large KBs, including SPARQL [6], Xcerpt [7], RQL [8]. However, learning these languages adds a limitation, as one needs to be familiarized with the query language, it’s syntax, it’s semantics, and the ontology of the knowledge base.

By contrast, KB-QA which takes natural language as it’s query is a more user-friendly solution and became a research focus in recent years [9]. There are two main research streams for the task of QA: Semantic Parsing-based systems (SP-based) [10], [11], [12], and Information Retrieval-based systems (IR-based) [13], [14], [15], [16]. SP-based methods address the QA problem by constructing a semantic parser that converts the natural language question into a conditionally structured expressions like the logical forms [17] and then run the query on the Knowledge base to obtain the answer. SP- based methods consists of three modules: (1) Entity linking, recognizes all entity mentions in a question and links each mention to an entity in KB; (2) Predicate mapping, finds candidate predicates in KB for the question; (3) Answer selection, converts the candidate entity-predicate pairs into a query statement and queries the knowledge base to obtain the answer. IR-based methods [13], [14], [15], [16] focus on mapping answers and questions into the same embedding space, where one could query any KB, independent of its schema without requiring any grammar or lexicon. IR-based methods are more flexible and require less supervision compared with the SP-based approaches [10], [11], [12].

With the advancement in the embedding techniques which can capture the semantics, deep neural networks are applied in several areas of Natural Language Processing (NLP) [18] and Natural Language Understanding (NLU) [19]. In the field of KB-QA, under IR-based umbrella, embedding-based approaches [20] [21] have been proposed. However, these approaches face two limitations. First, these models encode different components separately without learning the representation of the whole KB. Hence, are not able to capture the compositional

semantics in a global perspective. Second, the performance of Long Short Term Memory [22](LSTM)s, Bi-directional LSTM [23](BiLSTM)s, and Convolutional Neural Network [24](CNN)s are heavily dependent on large training data sets which is often not available in practice. In recent years, pre-trained Language Models [25] [26] [27] [28] on large-scale unsupervised corpus has shown its advantages on mining prior linguistic knowledge automatically, it indicates a possible way to deal with above problems.

In comparison to previous embedding-based approaches, we focus on exploiting pre-trained language models for IR based KB-QA task. We use BERT [25] pre-trained language model embeddings to encode our question and candidate answer contexts. We exploit a multi head attention encoder based on Convolution Neural Network (CNN) encoder [29] [30] to fine tune Bert pre-trained embeddings in particular for the KB-QA task. Since we use a Language Model trained on the English Wikipedia and Brown Corpus [25] for our question and KB representations, Out Of Vocabulary (OOV) problem can be negated. The interrelationships between the questions and the underlying KB are stored as the context for our model. This context is used to filter out the candidate answers by implementing cross attention between the question being asked and the KB.

The contribution of this paper is summarized as follows: (1) We propose a method called *Language Model based Knowledge Base Question Answering* (LM-KBQA). It exploits the BERT pre-trained language model embeddings [25] (which eliminates the need for using Recurrent Neural Network(RNN) architectures (LSTM, Bi-LSTM) to capture the contextual representations); (2) it is based on CNN encoder with self multi-head attention mechanism to fine tune the Bert embedding for the KB-QA task; (3) our results demonstrate the effectiveness of the proposed approach on the open Web-Questions data-set [31].

II. RELATED WORK

Over the past few years we have seen a growing research on KB-QA, shaping interactive paradigm. This enables user to take advantage of the communicative power of linguistic web knowledge while hiding their complexity behind an easy-to-use interface. At the same time the abundance of information has led to a heterogeneous data landscape where QA systems struggle to stay up with the quantity, variety, and truthfulness of the underlying knowledge. In general, the most popular methods for KB-QA can be mainly divided into two classes: SP-based and IR-based.

Semantic parsing(SP) based approaches focuses on constructing a semantic parsing tree or equivalent query structure that represents the semantic meaning of the question. In terms of logical representation of natural language questions, many methods have been implemented, such as query graph [32], [33] or RDF query language [12] [34].

Information retrieval(IR) based system try to obtain target answer directly from question information and KB knowledge without explicit considering interior query structure. There are various methods [13], [16], [35], [36] to select candidate answers and rank results.

[20] Were the pioneers to use embedding-based models to resolve the KB-QA problem. The queries and KB triples were represented as vectors in an exceedingly low dimensional vector space. Subsequently the cosine similarity is applied to find the most accurate answer. Also the Bag Of Words (BOW) methodology [37] is used to generate one vector for all the query answers. The Pairwise training methodology is utilized and the negative samples are randomly extracted from the KB. Authors in [15] improved their work based on sub-graph embeddings which is capable of answering complex questions. Their main motive was to include the maximum amount of information within the answer context which encodes the surrounding sub-graph of the KB(e.g., answer path and context). The proposed sub-graph embeddings accommodate all the entities and relations that are connected to the answer entity. The resulting vector was obtained using BOW [37] representations. In follow-up work [16] [38], memory networks [39] were used to store candidate set, and could be iteratively accessed to mimic multi-hop reasoning. Unlike the above methods that mainly use a bag-of-words (BOW) approach to encode questions and KB contexts, [21], [35] apply more advanced network modules (e.g., CNNs and LSTMs). The approach proposed in [32] is fixated on single-relation queries. The KB-QA task was sub-divided into 2 stages. First and foremost, the topic entity of the question was found. Then, the later question was considered by CNNs and utilize to match relations. Authors in [35], thought of the diverse features of answers to represent questions respectively using 3 columns of CNNs. Authors in [36], [40] proposed to use multi-channel CNNs to extract the relations of KB triples while exploiting the free text of the Wikipedia.

Most embedding-based approaches encode question and KB contexts independently. In NLP, [41] was the first to apply attention model. By cooperatively learning to align and translate, they enhanced the encoder-decoder Neural Machine Translation (NMT) framework. They argued that having source sentence by a fixed vector is illogical, and proposed a soft-align method, which could be understood as attention mechanism. [35] represented questions using three CNNs with different parameters when dealing with different answer contexts including answer path, answer context and answer type. With this inspiration [21] proposed a cross-attention mechanism to encode questions according to various candidate answer aspects. Further more [42] proposed bidirectional attention similar to those applied in machine reading comprehension [43] [44] [45] by modeling the interactions between questions and KB contexts. We take a stride forward incorporating BERT [25] pre-trained language model embeddings to encode

our question and candidate answer contexts from KB. We fine tuned BERT for the KB-QA problem with a Multi-Head Attention mechanism [29] which is based on a Convolution Neural Network (CNN) [30] encoder. Our proposed architecture is also based on the bi-directional cross attention mechanism between the asked question and KB answer contexts [42].

III. OVERVIEW

The goal of the KB-QA task can be formalized as follows: given a natural language question q , the model should return an entity set A as answers. The general architecture of a KB-QA system is illustrated in Figure 1. First, the candidate entity of the question is identified, then the candidate answers are generated from the Freebase [4] knowledge base.

The questions are then encoded using pre-trained-Bert model [25] which is fine tuned with a CNN encoder with Multi-Head attention [29] for relationship understanding. Then, cross attention neural networks [21] are employed to represent question under the influence of candidate answer set aspects (like entity types, relation paths) and vice versa. Finally, the similarity score between the question and each corresponding candidate answer set is calculated, and the candidates with the highest score will be selected as the final answer.

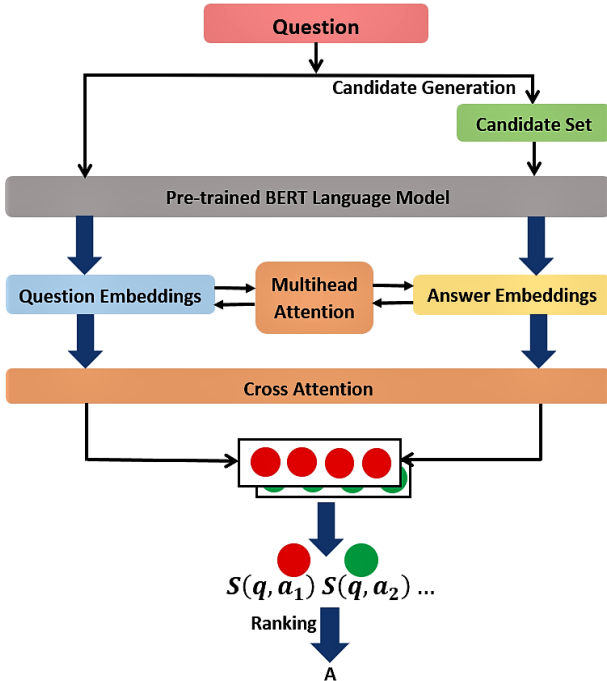


Fig. 1: General Architecture of a KB-QA.

We refer to Freebase [4] as our knowledge base. It has more than 3 billion facts, and is used as the supporting knowledge base for many Question Answering systems. In Freebase, the facts are represented as subject-predicate-object triples (s,p,o).

Following is an example triple in Freebase: (/m/01428, /language/human_language/countries_spoken_in, /m/03_r3) which relates to the fact that the language spoken in the country of Jamaica is Jamaican English where "/m/03_r3" denotes the country Jamaica while "/m/01428" denotes the language Jamaican English, and "/language/human_language/countries_spoken_in" is a relationship between the Freebase entities.

IV. OUR APPROACH

A. Candidate Generation

Ideally all entities in the Freebase should be considered for candidate answers, but practically, this is computationally expensive and can be avoided. We use Freebase API [4] to find a named entity for each question q which acts as the primary entity for the given question. For general understanding, Freebase API methodology resolved 86% of questions in open Web-Questions data-set when top-1 accuracy criteria [13] is used.

For example if the question "what language Jamaican people speak?" is passed through the Freebase API, it returns "Jamaica" as the main entity. After the named entity is identified with Freebase API, we gather all the other entities that are directly associated to the named entity with in 2-hop. These entities create the candidate set.

B. Question Representation

Initially, we need to collect the representations for each word in the question. These representations contain all the information of the question, and could serve as follows: For example, the question Q is represented as $Q = (x_1, x_2, \dots, x_n)$ where x_i stands for i^{th} word. The input natural language question $Q = \{q_i\}_{i=1}^{|Q|}$ as a sequence of word embeddings q_i fine tuned from Bert pre-trained language model using an encoder layer.

C. Context Representation

For each candidate set from the KB, we generate context for each question focusing on three aspects: answer type, answer path, and answer context from KB.

Answer type contains entity type information and helps in narrowing down entities while ranking the answers. If a question uses the word "who", the candidates answers that are relevant to a person are more likely to be correct. Answer path is a sequence of relations from a candidate set to a named entity. For example "/language/human_language/countries_spoken_in" is a relation path between Jamaica and Jamaican English which is stored as [human, language, countries, spoken, in] in the Freebase.

Answer context is defined as the surrounding entities (e.g., neighbour nodes) of the candidate which help to answer the asked questions with constraints. We only consider the context nodes that have overlap with the asked questions. In particular, for each context node (i.e.

a sequence of words) of a candidate, we first compute the longest common sub-sequence between the neighbors and the question. If there exists some common sub-sequence between question and neighbor entities we store them as Answer context which will help in answering questions with multiple entities. All this information is stored as the context and will be fed into the pre-trained Bert model to generate embeddings and are fine tuned using an encoder layer.

D. BERT Language Model

BERT is a multi-layer bidirectional transformer encoder. The given input is a token sequence at the character-level. This means it is either a single sequence or a special token [SEP] separated sentence pair. Each token in the input sequences is the aggregate of the token embeddings, the segment embeddings, and the position embeddings. The initial token in every sequence is consistently a special classification symbol ([CLS]), and for the classification tasks it uses the hidden state of the token. After fine tuning, the pre-trained BERT [25] representations are used in several natural language processing tasks.

1) *Notation*: For an input sequence of word or sub-word tokens $X = (x_1, \dots, x_n)$, BERT trains an encoder that generates a contextualized representations for each token: $x_1, \dots, x_n = \text{enc}(x_1, \dots, x_n)$. Each token in the sequence, positional embeddings p_1, \dots, p_n are used to label the absolute position of input sequence because deep transformer is used to implement the encoder.

2) *Masked Language Modeling (MLM)*: Masked Language Modeling (MLM) or "Cloze test", is used to predict the missing tokens from their placeholders in a given sequence. When a subset of tokens $Y \subseteq X$ sampled and substituted with placeholder set of tokens. In Bert's MLM implementation, Y accounts for 15% of the tokens in X; of those, 80% are replaced with [MASK], 10% are replaced with a random token (according to the unigram distribution), and 10% are kept unchanged. The main idea is to predict the modified input from the original tokens in Y. BERT selects each token in Y independently by randomly selecting a subset.

3) *Next Sentence Prediction (NSP)*: The Next Sentence Prediction (NSP) predicts whether X_B is immediate continuation of X_A when two sequences X_A and X_B are given as input. In BERT methodology, it initially takes X_A from corpus. Then it either reads X_B from where X_A has terminated or it randomly samples X_B from different point in corpus. A special token [SEP] is used to separate the two sequences. Also, a special token [CLS] is appended to X_A and X_B to express as an input. The main idea of [CLS] is to truly find whether X_B follows X_A in the corpus.

E. Embedding Layer

To embed question and KB context we use pre-trained BERT [25] language model. Without the need of Recurrent

Architecture BERT uses positional embeddings to encode the word sequences. By using unsupervised learning based on Masked Language Models (MLM) and Next Sentence Prediction (NSP), BERT embeddings are generated. These embeddings contain bidirectional attention from Masked Language Model(MLM) section of BERT.

BERT model takes sentences in one or two formats. If we have a context of [location, language, human, language, countries, spoken, in, Jamaican] and a question [what does Jamaican people speak?], we can either encode it as a single input or two separate inputs to the BERT Model as shown below.

<CLS>[question]<SEP> [context] Or
 <CLS>[question]
 <CLS>[context]

We provide Context and Query together with query first followed by the context as shown in the figure above. Since BERT is trained on "next sentence prediction" outcome, we believe that using this formulation will provide richer query and context embeddings. The maximum length of the context and the question are 96 and 18 respectively. We use the concatenation of all 12 layers of BERT model as our embedding layer.

F. Encoder Layer

The encoder layer is a stack of following layers:

[CNN-layer + self-attention-layer + feed-forward-layer]

Similar to [44], the input of this layer at each position is $[c, a, c \odot a, c \ominus b]$, where a and b are respectively a row of attention matrix A and B. For the self-attention-layer, we adopted multi-head attention mechanism defined in [29] which is the improvement of the basic attention mechanism.

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \quad (1)$$

$$Multi-Head(Q, K, V) = Concat(head_1, \dots, head_h) \quad (2)$$

$$SelfMulti-Head = Multi-head(X, X, X) \quad (3)$$

These three equations are the formation process of the self multi-head attention mechanism. The matrix of W is the weight matrix. The Q, V, K should multiply its corresponding weight matrix before getting into the attention function. Repeat this process h(number of heads) times and connect each result then we can get a new vector matrix that reflects the relationship between Q, V. Especially, in the self multi-head attention mechanism, to look for internal connections within words, the $Q = V = K = X$, and X represents the word vector matrix.

The multi-head attention mechanism helps the model learn the words relevant information in different presentation sub-spaces. The self-attention mechanism can extract the dependence in words. As the name shows, the self multi-head attention mechanism integrates the benefits of both, creates a context vector for each word. Then we don't need to depend on additional information to get a matrix that reflects the context relationship between

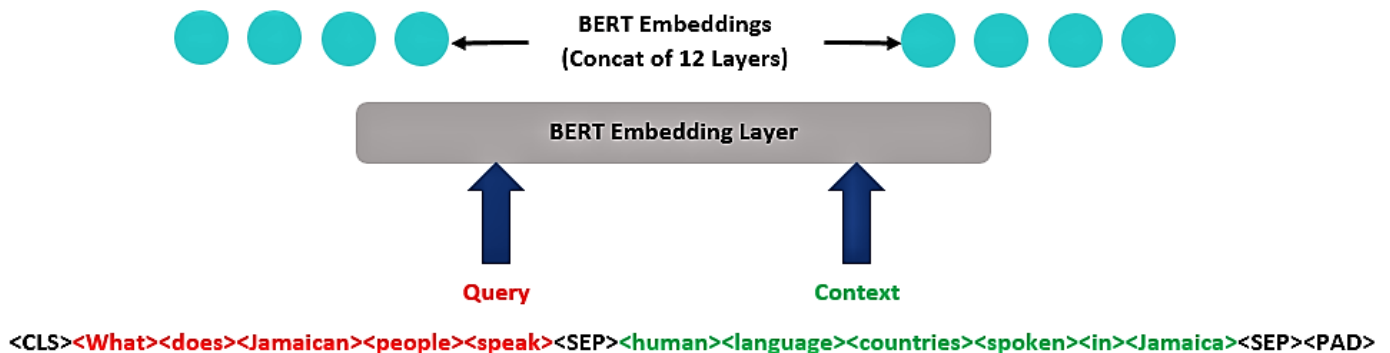


Fig. 2: BERT Embedding Layer

current word and other words in a sequence. Each of these basic operations (cnn/self-attention/ffn) is placed inside a residual block

G. Attention Layer

We use a focused, Context-Query attention layer on top of the pre-trained BERT embeddings identical to that of the QANet [46] model. Such attention modules were standard in other KB-QA models such as [21] [42].

C is used to denote context from KB and Q for natural language question. The context-to-query attention is constructed as follows: (1) Compute similarities between each pair of context and query words that generates a similarity matrix $S \in R^{n \times m}$. (2) Normalize each row of S by applying soft-max function. (3) Compute context-to-query attention $A = S \cdot QT \in R^{n \times d}$

The similarity function used here is the tri-linear function as mentioned in [44] : $f(q, c) = W_0[q, c, q \odot c]$

V. EXPERIMENTS

A. Dataset

We use the Web-Questions [31] dataset. Web-Questions dataset is built by using the Google Suggest API to obtain questions that begin with a wh-word and contain exactly one entity. Specifically, they queried the question excluding the entity, the phrase before the entity, or the phrase after it. Each query generates 5 candidate questions, which are added to the queue. Further they iterated until

1 million questions were visited; a random 100K were submitted to Amazon Mechanical Turk (AMT). The AMT task requested that the distributed workforce should answer the question using only the Freebase page of the questions' entity, or otherwise mark it as unanswerable by Freebase. The answer was restricted to be one of the possible entities, values, or list of the entities on the page. Thus this combination of Web-Questions dataset with Freebase KB is used as the base line.

B. Comparison with State of the art

To assess the proposed methodology, experiments were conducted on the Freebase KB and the Web-Questions [31] dataset. Freebase is large-scale KB [4] that has organised

general facts as subject-predicate-object triples. It has 41M non-numeric entities with 19K unique properties and 596M assertions. The Web-Questions [31] dataset has 3,778 question-answer pairs for training and 2,032 for testing. The questions are gathered from Google Suggest API, and the answers from Amazon Mechanical Turk which are labelled manually. All of the answers are from the Freebase KB. we use 80% of the training data as training set and 20% as validate set. The evaluation metric we use is F1 score, and the average result is calculated by [31] script.

1) *Settings*: For KB-QA training, we use pre-trained Bert embeddings base uncased version. During tokenization, BERT [25] code uses a word-piece algorithm to split words into sub-words and all less frequent words will be split into two or more sub-words. The vocab size of Bert is 30522. We adopt delexicalization strategy as mentioned in [42]. For each question, the candidate entity mentions those belonging to date, ordinal, or number are replaced with their type. Same is applied on answer context from KB text, if the overlap belongs to above type. This assures that the query matches up with answer context in the embedding space. The dropout rates for both question and answer encoder side is set to 0.3. The batch size is set as 4 and answer module threshold is set to 0.7 to allow multiple answers for questions with list of answers. We use Adam optimizer [47] to train the model. The initial learning rate is set to 0.01. Further learning rate is reduced by a factor of 10 if no improvement is observed in validation process for 3 successive epochs. The hyper-parameters are tuned on validation-set.

2) *Results*: In this section, we compare the performance of our method with other IR-based approaches. The results are shown in Table 1. Based on this table our method (LMKB-QA) obtained an F1 score of 50.9 on Web-Questions using the topic entity predicted by Freebase API. As Table 1 shows our method achieves better results or even competes with state-of-the-arts. This demonstrate the effectiveness of our idea to use Bert pre-trained language model embeddings in the Question Answering on Knowledge Base problem.

It is important to note that our proposed approach is

Methods	Avg F1
Bordes 2014b [20]	29.7
Bordes 2014a [15]	39.2
Yang 2014 [48]	41.3
Dong 2015 [35]	40.8
Bordes 2015 [16]	42.2
Xu 2016 [36]	42.2
Hao 2017 [21]	42.2
Chen 2019 [42]	49.7
Chen 2019 [42] topic entity predictor	51.8
Our Approach	52.7

Table I: Evaluation results on Web-Questions

based on a CNN architecture, and only depends on the training data (it does not depend on wiki text [40]). [20] applies BOW method to obtain a single vector for both questions and answers. [15] further improve their work by proposing the concept of sub-graph embeddings. Besides the answer path, the sub-graph contains all the entities and relations connected to the answer entity. The final vector is also obtained by bag-of-words strategy. [48] follows the SP-based manner, but uses embeddings to map entities and relations into KB resources, then the question can be converted into logical forms. [35] use three columns of Convolution Neural Networks (CNNs) to represent questions corresponding to three aspects of the answers, namely the answer context, the answer path and the answer type. [16] put KB-QA into the memory networks framework [49], and achieves the state-of-the-art performance of end-to-end methods. Our approach incorporates pre-trained Bert language model embeddings fine tuned for the KB-QA task using a Encoder architecture with multi head attention. [15], [16], [20] all utilize BOW model to represent the questions, while ours takes advantage of pre-trained language model embeddings. Also note that [16] uses additional training data such as Reverb [50] and their original data set Simple-Questions.

The proposed method in [35] employs three fixed CNNs to represent questions. [21] implemented the mutual influence between the representation of questions and the corresponding answer aspects. [42] further enhanced the result by using memory networks [39] to control the mutual influence between the representation of questions and the corresponding answer aspects. The approach in [42] achieved an F1 score of 0.518 using a custom topic entity predictor while an F1 score of 0.497 has achieved via the Freebase Search API. The [33], [51] have higher F1 scores than other methods. Their approach was able to address more questions with constraints and aggregations. But, their methods apply number of manually designed rules and features, which come from the observations on the training questions set. This is a manual process which reduces the versatility of their proposed approach. Integrated systems like [36], [40] generates higher F1 score leveraging Wikipedia free text as external knowledge, which are based on semantic parsing. Therefore the sys-

tems are not directly compared to ours.

VI. CONCLUSION

In this paper, we focused on using a pre-trained language model for KB-QA task. Firstly, we used *Bert base uncased* for the initial experiments. We further fine tuned these embeddings with two way attention mechanism from the knowledge base to the asked question and from the asked question to the knowledge base answer aspects. Our method is based on a simple CNN architecture with Multi-Head Attention mechanism to represent the asked question dynamically in the multiple aspects. Our experimental results show the effectiveness of the Bert pre-trained language model embeddings.

In the future, we would like to explore other language models including GPT-2, RoBERTa, Transformer-XL and XLNet. We will also further investigate to answer even more complex questions on the knowledge bases.

VII. ACKNOWLEDGEMENTS

We would like to thank Hugging Face [52] for their open-source code to use pretrained BERT embeddings for downstream tasks using PyTorch, BAMnet [42] for open-source code to process Freebase knowledge base dump, and Web-Questions [31] for the train, test dataset, and F1 metric script for comparison.

REFERENCES

- [1] Amit Singhal. Introducing the knowledge graph: things, not strings. *Official google blog*, 16, 2012.
- [2] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web, WWW 2007*, page 697–706, New York, NY, USA, 2007. Association for Computing Machinery.
- [3] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. Dbpedia—a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195, 2015.
- [4] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. AcM, 2008.
- [5] Graham Klyne, Jeremy J Carroll, and B McBride. Resource description framework (rdf): concepts and abstract syntax, 2004. *February*. URL: <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210>, 2009.
- [6] Andy Seaborne, Geetha Manjunath, Chris Bizer, John Breslin, Souripriya Das, Ian Davis, Steve Harris, Kingsley Idehen, Olivier Corby, Kjetil Kjernsmo, et al. Sparql/update: A language for updating rdf graphs. *W3c member submission*, 15, 2008.
- [7] François Bry, Paula-Lavinia Patrânjan, and Sebastian Schaffert. Xcerpt and xchange – logic programming languages for querying and evolution on the web. In Bart Demeo and Vladimir Lifschitz, editors, *Logic Programming*, pages 450–451, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.
- [8] Brice Chardin, Emmanuel Coquery, Marie PAILLOUX, and Jean-Marc Petit. Rql: A sql-like query language for discovering meaningful rules. volume 2014, pages 1203–1206, 12 2014.
- [9] Christina Unger, André Freitas, and Philipp Cimiano. An introduction to question answering over linked data. In *Reasoning Web International Summer School*, pages 100–140. Springer, 2014.

- [10] Luke S Zettlemoyer and Michael Collins. Learning context-dependent mappings from sentences to logical form. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 976–984. Association for Computational Linguistics, 2009.
- [11] Luke S Zettlemoyer and Michael Collins. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. *arXiv preprint arXiv:1207.1420*, 2012.
- [12] Wanyun Cui, Yanghua Xiao, Haixun Wang, Yangqiu Song, Seung-won Hwang, and Wei Wang. Kbcqa: learning question answering over qa corpora and knowledge bases. *Proceedings of the VLDB Endowment*, 10(5):565–576, 2017.
- [13] Xuchen Yao and Benjamin Van Durme. Information extraction over structured data: Question answering with freebase. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 956–966, 2014.
- [14] Yanchao Hao, Yuanzhe Zhang, Kang Liu, Shizhu He, Zhanyi Liu, Hua Wu, and Jun Zhao. 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)*, pages 221–231, 2017.
- [15] Antoine Bordes, Sumit Chopra, and Jason Weston. Question answering with subgraph embeddings. *arXiv preprint arXiv:1406.3676*, 2014.
- [16] Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. Large-scale simple question answering with memory networks. *arXiv preprint arXiv:1506.02075*, 2015.
- [17] Hiyan Alshawi and Jan van Eijck. Logical forms in the core language engine. In *27th Annual Meeting of the Association for Computational Linguistics*, pages 25–32, 1989.
- [18] Ivan A. Sag. Linguistic theory and natural language processing. In Ewan Klein and Frank Veltman, editors, *Natural Language and Speech*, pages 69–83. Berlin, Heidelberg, 1991. Springer Berlin Heidelberg.
- [19] Terry Winograd. Language as a cognitive process: Volume 1: Syntax. 1983.
- [20] Antoine Bordes, Jason Weston, and Nicolas Usunier. Open question answering with weakly supervised embedding models. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 165–180. Springer, 2014.
- [21] Yanchao Hao, Yuanzhe Zhang, Kang Liu, Shizhu He, Zhanyi Liu, Hua Wu, and Jun Zhao. 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)*, pages 221–231, Vancouver, Canada, July 2017. Association for Computational Linguistics.
- [22] Hasim Sak, Andrew W Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. 2014.
- [23] Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. Bidirectional lstm networks for improved phoneme classification and recognition. In *Proceedings of the 15th International Conference on Artificial Neural Networks: Formal Models and Their Applications - Volume Part II, ICANN 2005*, page 799. Springer, Berlin, Heidelberg, 2005. Springer-Verlag.
- [24] Kunihiro Fukushima. Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural networks*, 1(2):119–130, 1988.
- [25] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [26] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.
- [27] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. [URL https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/language_understanding_paper.pdf](https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/language_understanding_paper.pdf), 2018.
- [28] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.
- [29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.
- [30] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.
- [31] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.
- [32] Wen-tau Yih, Xiaodong He, and Christopher Meek. Semantic parsing for single-relation question answering. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 643–648, Baltimore, Maryland, June 2014. Association for Computational Linguistics.
- [33] Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1321–1331, Beijing, China, July 2015. Association for Computational Linguistics.
- [34] Sen Hu, Lei Zou, Jeffrey Xu Yu, Haixun Wang, and Dongyan Zhao. Answering natural language questions by subgraph matching over knowledge graphs. *IEEE Transactions on Knowledge and Data Engineering*, 30(5):824–837, 2017.
- [35] Li Dong, Furu Wei, Ming Zhou, and Ke Xu. Question answering over Freebase with multi-column convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 260–269, Beijing, China, July 2015. Association for Computational Linguistics.
- [36] Kun Xu, Siva Reddy, Yansong Feng, Songfang Huang, and Dongyan Zhao. Question answering on Freebase via relation extraction and textual evidence. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2326–2336, Berlin, Germany, August 2016. Association for Computational Linguistics.
- [37] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.
- [38] Sarthak Jain. Question answering over knowledge base using factual memory networks. In *Proceedings of the NAACL Student Research Workshop*, pages 109–115, 2016.
- [39] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *arXiv preprint arXiv:1410.3916*, 2014.
- [40] Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao. Hybrid question answering over knowledge base and free text. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2397–2407, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.
- [41] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.
- [42] Yu Chen, Lingfei Wu, and Mohammed J Zaki. Bidirectional attentive memory networks for question answering over knowledge bases. *arXiv preprint arXiv:1903.02188*, 2019.
- [43] Caiming Xiong, Victor Zhong, and Richard Socher. Dynamic coattention networks for question answering. *arXiv preprint arXiv:1611.01604*, 2016.
- [44] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*, 2016.

- [45] Shuohang Wang and Jing Jiang. Machine comprehension using match-*lstm* and answer pointer. *arXiv preprint arXiv:1608.07905*, 2016.
- [46] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. Qanet: Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv:1804.09541*, 2018.
- [47] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [48] Min-Chul Yang, Nan Duan, Ming Zhou, and Hae-Chang Rim. Joint relational embeddings for knowledge-based question answering. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 645–650, Doha, Qatar, October 2014. Association for Computational Linguistics.
- [49] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448, 2015.
- [50] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics.
- [51] Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–206, Berlin, Germany, August 2016. Association for Computational Linguistics.
- [52] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R’emi Louf, Morgan Funtowicz, and Jamie Brew. Hugging-face’s transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019.