DEIM Forum 2019 G7-5

A Scheme for Factoid Question Answering over Knowledge Base

Buzaaba HAPPY† and Toshiyuki AMAGASA††

† Systems and Information Engineering University of Tsukuba
†† Center for Computational Sciences University of Tsukuba
1 Chome- 1 − 1 Tennodai, Tsukuba, Ibaraki Prefecture 305-8577, Japan
E-mail: †happy-b@kde.cs.tsukuba.ac.jp, ††amagasa@cs.tsukuba.ac.jp

Abstract Large scale knowledge bases like Freebase or DBpedia, consist of a large pool of information that is used to answer questions expressed in natural language. The question answering over knowledge graphs aims at providing the answers to natural language questions by looking up facts that are already stored in the knowledge base. In this work, we focus on simple questions which require the extraction of a single fact from the knowledge base to be answered. Although this might seem less challenging, recent attempts that have applied end-to-end approaches with complex neural networks have shown that it is still a challenging task especially when asked against a large knowledge base. We propose a combination of standard neural network and non neural network techniques for answering factoid questions over knowledge base. Our approach differs from previous end-to-end approaches by splitting the question answering problem into independent sub-problems namely; detecting entities, linking entities to the knowledge base and classifying the question into one of the relation types in the knowledge base. Our approach yields competitive results with state-of-art approaches that apply complex neural networks on a similar data set.

Key words Question Answering, Knowledge Base

1 Introduction

Question answering (QA) often refers to building systems that automatically answers questions expressed in natural language. It is one of the oldest research areas with a variety of applications in Natural Language Processing (NLP) tasks such as information retrieval and entity extraction. Recently, Question Answering has also been applied in developing chatbots[1] and dialogue systems designed to simulate human conversation. There are two major paradigms of question answering: Question Answering over free-text which aims at providing the answers to questions expressed in natural language without any domain restriction and On the other hand, Question Answering over knowledge base (KB) which provides the answer to the question by looking up the facts stored in the KB. Traditionally, research in the question answering domain used a pipeline of conventional linguistic-based NLP techniques, such as part-of-speech tagging, parsing and conference resolution. Many of the state-of-art QA systems like IBM Watson[2] have applied these methods.

However, the recent developments in deep learning has triggered a line of work tempting many researchers to investigate more language agnostic approaches together with complex neural network architectures that have outperformed traditional approaches on a variety of natural language processing tasks like opinion extraction[3], sentence classification[4] and dependency parsing[5].

This study focuses on simple factoid question answering (SimpleQA) based on simpleQuestions benchmark[6] in which answering a question requires the extraction of a single fact from the knowledge base.

Information in many knowledge bases is stored in the form of RDF triples (subject, predicate, object) [7], for example a freebase triple (/m/02mjmr, /People/Person.place of birth, /m/025vc42) where **/m/02mjmr** represent the freebase MID for **Barack Obama**, and **/m/025vc42** is the the MID for **Honolulu**. Therefore asking a question like *"where was Barack Obama born?"*, getting the correct answer from the knowledge base would require retrieving the subject entity **/m/02mjmr** in freebase that represents Barack Obama and the predicate **/People/Person.place of birth**. The retrieved entity and predicate would form a structured query represented as a pair of the subject entity and the relation that would be used to obtain the right answer **/m/025vc42** which represents the object entity **Honolulu** in freebase.

Although several studies on Simple question answering task have increasingly applied complex neural network architectures, our argument inspired by an emerging research area that aims at improving empirical rigor in the machine learning field by focusing on knowledge and insights as opposed to simply winning sculley et al[8], is of understanding what exactly contributes to the effectiveness of a particular neural network, we avoid the complex NN architectures and apply standard/baseline NN architectures on SimpleQuestions data set which generates results competitive to the state of the art. We take a step further to show that combining NN with non neural network techniques like conditional random field (crf) achieves reasonable results. Basing on our results, we seem to agree with the argument that standard neural network architectures when properly tuned, Outperform some recent complex models[9].

2 Related work

Question answering over knowledge base research has gradually developed from earlier domain-specific question answering (Tang and Mooney[10]), to open-domain question answering based on large scale knowledge bases. For several years research has been conducted to tackle this problem by directly parsing the natural language question into a structured query using semantic parsing Kwiatkowski et al[11], and more recent work includes designing knowledge specific logical representation and grammar parsing Berant et al[12].

Another line of research tackles the problem by deep learning powered similarity matching. In his work, Bordes et al[6] proposed the single-relation factoid question answering. This work introduces a new data-set SimpleQuestions with 108,000 questions built on Freebase and proposes a memory network to solve the simple question answering task. This data set prompted a new line of work and in the past years several researchers have applied even more complex neural network architectures to address this problem: Golub and He[13] proposed a character-level attention-based encoder-decoder model, Lukovinikov et al[14] applies a hierarchical word-level and character-level question encoder to train a neural network in an end-to-end manner. Dai et al[15] proposes a conditional probabilistic framework using BIGRUs to infer the target relation first and then the target subject associated with the candidate relations. Yin et al[16] used character-level convolutional Neural Network for entity linking and a separate word-level convolutional Neural Network with attentive max-pooling that models the relationship between the predicate and question pattern more effectively. Yu et al[17] applied a residual hierarchical BILSTM that performs hierarchical matching between questions and knowledge base relations for relation prediction, the results were then combined with the entity linking output. The above deep learning approaches, exploit increasingly complex techniques.

In this work we build on a more close related work by Ture and Jojic[18] which argues that baseline methods when fully explored can equally produce competitive results. His work formulates the question answer problem into two machine learning tasks: entity detection and relation classification which then applies simple recurrent neural network and urges that taking advantage of the problem structure yields accurate and efficient results compared to complex neural network methods. We extend on this work to explore the performance of non neural network techniques on a similar problem.

Table 1 A table showing a few samples from the back-projected entity detection train dataset

Question	Entity Labels
what is terry adamson 's nationality ?	O O I I O O O
which track is in the recording teleportation	O O O O O O I
what is ellen swallow richards 's nationality ?	O O I I I O O O
which country is albert bolender from	O O O I I O

3 Approach

The task of question answering over knowledge base can be represented as follows; Let G be the knowledge base representing a set of triples;

$$G = \{(Si, Pi, Oi)\} \quad (1)$$

where: *Si: Subject entity, Pi: Predicate or relation, and Oi: an Object entity.*

Therefore given a simple natural language question q represented as a sequence of words,

$$q = \{w1, w2, ..., wt\} \quad (2)$$

The task is to find a triple;

$$(\hat{s},\hat{p},\hat{o}) \in G \quad (3)$$

such that $\hat{o}$ is the intended answer to the question. We therefore formulate this task to finding the right subject $\hat{s}$ and predicate or relation $\hat{p}$ referred to in the question q that characterizes a set of triples in the knowledge base G that contain the answer $\hat{o}$ to the question.

As mentioned above, we split the question answering task into the following sub-tasks;

Entity detection To identify the entity in the question we formulate this as a sequence labeling problem where each word or token is tagged as entity or non-entity.

Each question word/token is represented with a word embedding, the input word representation is then combined with the hidden layer representation from the previous time step using either BiLSTM[19] or BiGRU[20] standard RNNs which then applies a non-linear transformation to compute the hidden layer representation at the current time step. The final hidden representation at the current time step is then projected to the output dimensional space and normalized into a probability distribution via a softmax layer. The standard RNNs used in our model, apply both LSTMs and GRUs for calculating the hidden states of the network. Below we briefly describe the gated recurrent units (GRU) shown in **Figure 2** because they are commonly used due to their ability to process longer sequences brought about by their additive manipulation of the state vector and explicit filtering using gates.

Given a sentence, as we read the sentence from left to right, the GRU is going to have a new memory variable called the memory cell/hidden state $C^{<t>}$ which provides a bit of memory to remember so that when the network gets further into the sentence it can still remember the subject of the sentence and so at time step t the GRU will output an activation function equivalent to the memory cell at the time step.

Therefore the current memory cell/hidden state $C^{<t>}$ at time step t is computed by interpolating between the previous hidden state $C^{<t-1>}$ at previous time step and the candidate state $\hat{C}^{<t>}$ at the current time step.

$$C^{<t>} = \Gamma_u \odot \hat{C}^{<t>} + (1 - \Gamma_u) \odot C^{<t-1>} \tag{4}$$

with Γ_u the update vector and $\odot$ the element wise vector product. For interpolation, the update gate which determines how much of the previous state is leaked into the current state Γ_u is computed using the current input $X^{<t>}$ and the previous state $C^{<t-1>}$. The update gate can decide to forget the previous state altogether or copy the previous state and ignore the current input.

$$\Gamma_u = \sigma(W_u[C^{<t-1>}, X^{<t>}] + b_u) \tag{5}$$

Where W_u and b_u are parameter metrics to be learned during training and σ the Sigmoid activation function $\sigma(x) = \frac{1}{1+e^{-x}}$ applied element wise to the vector entries.

The current candidate memory cell/hidden state $\hat{C}^{<t>}$ is computed based on the current input $X^{<t>}$ and the previous hidden state $C^{<t-1>}$.

$$\hat{C}^{<t>} = tanh(W_c[\Gamma_r \odot C^{<t-1>}, X^{<t>}] + b_c) \tag{6}$$

where W_c and b_c are parameter metrics, *tanh* the hyperbolic tangent activation function and Γ_r is the reset gate which determines the parts of the previous state ignored in computation of the candidate state and it is computed as;

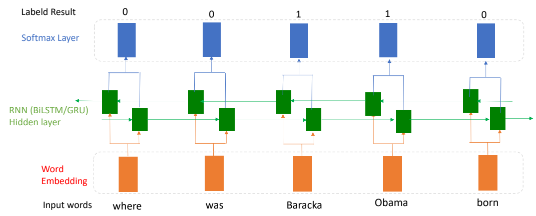

Figure 1 Entity detection architecture.

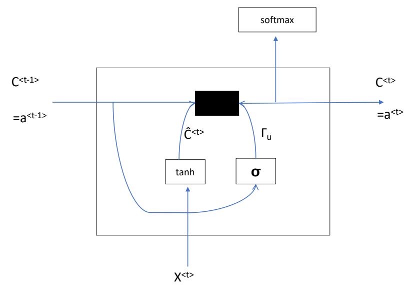

Figure 2 Schematic representation of GRU

$$\Gamma_r = \sigma(W_r[C^{<t-1>}, X^{<t>} + b_r]) \tag{7}$$

W_r and b_r are parameters.

We also apply conditional random fields (crf) to sequence labeling[21] to compare the entity detection performance with recurrent neural networks. The crf is a conditional sequence model which represents the probability of a hidden state sequence given some observations. We train the crf using Stanford Named Entity Recognizer (NER) a tool which can label word sequences into four classes; person, organization, location and non-entity. We tagged the question into four classes, where three of the classes were tagged as entity (person, organization, location) and so there were two classes entity and non-entity.

Entity linking The generated candidate entities are then linked to the actual knowledge base node. We use Freebase as our knowledge base where each node is represented with a machine identifier/MID. We treat this as a fuzzy string matching problem and follow a similar method proposed in [18]. For linking the extracted entity to the actual knowledge base node, we build three different indexes using dictionaris in python; a names index which maps all entity MID's in the Freebase subset to their names in the Freebase names file. The second index is the inverted entity index which maps all n-grams of an entity for $n \in \{1, 2, 3\}$ to the entity MID. The

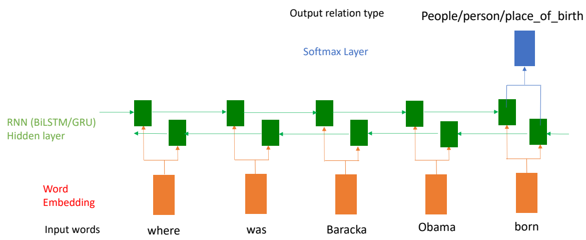

Figure 3 Relation Classification architecture.

final index is the reachability index that maps each entity node in the Freebase subset to all nodes reachable by a single relation and we append the retrieved entity MID's to the candidate list. An early termination proposed by Ture and Jojic[18] that stops searching for entities of smaller n value after candidate entities have been found is applied, and we calculate levenshtein distance to score entities in the candidate list and rank them in a descending order.

Relation classification The question is classified as one of the freebase knowledge base relations. We examine a model similar to that of entity detection, both BiLSTMs and BiGRUs are applied to model dependencies among words in the question. The difference is that relation classification is not a tagging task, we therefore base the classification decision on the output of the last hidden layer for prediction as shown in figure 2.

We also use CNNs for relation classification and following Kim et al[22], we modify the multi-channel model described in his paper to a single static channel instead, and apply the same model to our task of relation classification. CNNs are not recommended for sequence modeling but since they are likely to extract local features by sliding filters over the word embeddings we adopt them.

After generating the candidate entities, and relations in the previous steps, we come up with all possible (entity, relation) pairs, and using the reachability index we check the existence of the given combination in the knowledge base. Since entity linking and relation prediction are carried out separately, many combinations don't exist in the knowledge base. Those combinations that don't exist in the knowledge base are pruned. The final prediction is achieved by combining the scores of both entity linking and their relation prediction.

4 Experiments and Results

4.1 Experiments

We experiment on Freebase knowledge base[23] table

Table 2 A table showing a few Freebase samples

Subject	Predicate	Object
m/0n1vy1h	people/person/gender	m/05zppz
m/0n5xzdf	people/person/place_of_birth	m/0c_m3
m/0cz9079	soccer/football_player/position_s	m/02nzb8
m/04yltrc	book/book_edition/author_editor	m/034bs
m/09gl06d	music/album/album_content_type	m/02jbfk

Table 3 A table showing a few simplequestions samples in training set

Subject	Predicate	Object	Question
m/0ftqr	music/producer/tracks_produced	m/0p600l	what songs have nobuo uematsu produced?
m/01_d4	location/location/events	m/0b27c	Whats the name of a battle that happened in chicago
m/0j3yffd	people/person/nationality	m/0162v	What is terry adamson's nationality?
m/063vjv	automotive/model/automotive_class	m/03vnt4	in what automotive class is the hyundai santa fe
m/04wf9tq	book/written_work/subjects	m/019kbm	What is the subject of the book "Hearts West"?

Table 4 A table showing statistics of the freebase subsets

Item	2M-Subset	5M-Subset
No. Entities	2,150,604	4,904,397
No. Relations	6,701	7,523
No. Tripples	14,180,937	22,441,880

1, and the SimpleQuestions data-set[6] table 2. In freebase knowledge base, entities are connected by predefined predicates connecting from the Subject to the object. A triple (Subject, Predicate, Object) denoted as *(S, P, O)* describe a fact for example (Barack Obama, People/Person.place.of.birth, Honolulu) refers to the fact that Barack Obama was born in Honolulu.

The SimpleQuestions data-set consists of 108,442 natural language simple questions with their corresponding Freebase triples (subject, predicate, object) that provides an answer the question. We use the training, validation and test splits of 75,910, 10,845 and 21,687 questions respectively as provided by the data-set.

Following the previous work we use FB2M Freebase subset as the knowledge base which has 2 million entities and 6,701 relations table 3.

We compute precision, recall and F1 measure for evaluation in entity detection and evaluate recall for top results at

Table 5 Entity detection results for a given model

Model	Dataset	Precision (%)	Recall (%)	F1 (%)
LSTM	Validation	91.89	92.87	92.26
LSTM	Test	91.08	91.21	91.53
GRU	Validation	92.56	93	92.78
GRU	Test	92.09	92.92	92.5
CRF	Validation	90.71	89.92	90.36
CRF	Test	90.72	89.8	90.2

Table 6 Entity linking results for a given model

	LSTM		GRU		CRF	
R@k	Val	Test	Val	Test	Val	Test
1	0.679	0.662	0.676	0.661	0.663	0.649
5	0.827	0.811	0.825	0.808	0.809	0.796
10	0.863	0.849	0.86	0.848	0.845	0.834
20	0.889	0.876	0.885	0.876	0.871	0.861
50	0.912	0.903	0.909	0.903	0.895	0.889
100	0.927	0.921	0.925	0.92	0.912	0.907

k (R@k) for both entity linking and relation prediction. The prediction is marked as correct if both entity and relation match the ground truth in end-to-end evaluation.

We initialize the model word embeddings with a 300-dimensional pre-trained vectors provided by Glove[24]. The pre-trained word embeddings implicitly integrate word semantics inferred from large text corpus based on the distributional semantic hypothesis[25] which states that *"words with similar meanings occur in similar context"*. The pre-trained word embeddings allows to find better matches between words in the question and subject labels or relation URI's. It also allows to handle unseen words during training when it comes to training.

We implement the model in PyTorch v0.2.1 with a single CPU 3.3 GHz Intel core i5 macOS Sierra. and we use negative log likelihood loss function and Adam[26] for optimization, with the learning rate of 0.0001 in a mini-batch setting with batch size 32. To implement the conditional random field (crf), we use the Stanford Named Entity Tagger (NER)[27].

4.2 Results

In this section we present our results on the SimpleQuestions task and we begin with the results on individual components:

1. Entity Detection: Table 1 shows the models' results on the task of entity recognition. We evaluate the precision, recall and F1-score on the the token span level. This means that (a true positive span) the predicted entity token span exactly matches the ground truth from the back-projected dataset. The results reveal that RNN (LSTM GRU) perform better with F1-score of 92.5% for the GRU. It can also be noticed that the crf result of 90.2% is comparable.

2. Entity linking: Table 2 shows the model comparison of entity linking results. The CRF entity linking results accuracy is comparable to both LSTM and GRU. Although the crf may have performed slightly lower than the LSTM and GRU on entity detection, the bottleneck is entity linking since we see more entities in the knowledge graph with the same label that makes it difficult to identify the correct entity (MID).

Table 7 Relation prediction results for a given model

Model	Dataset	Precision	Retrieval@3	Retrieval@5
GRU	Validation	82.22	93.75	95.93
GRU	test	81.59	93.68	95.76
LSTM	Validation	81.76	93.73	95.85
LSTM	test	81.28	93.66	95.47
CNN	Validation	82.88	93.75	95.86
CNN	test	81.92	93.68	95.64

3. Relation Prediction: For the task of predicting the relation type of the question, the relation or predicate is given in the dataset. There are 1,837 unique relation types in the dataset. We conduct a large scale classification with 1,837 possible labels to assign a relation type to the question. from Table 7 we see that on precision, CNN out performs both RNN's (BiLSTM and BiGRUs). We however see that both RNN and CNN retrieval results (R@3) are essentially similar but RNN better at (R@3).

5. End-to-end: Table 4 shows end-to-end results for various combinations of entity detection and relation prediction on test set. The best model combination which achieves 74.64 accuracy is the BiLSTM for entity detection and BiGRU for relation prediction. We also compare our results with other state-of-the-art models on the SimpleQuestions test set . Our results outperform the complex neural network models like Bodes et al's memory network, Golub and He's attention-enhanced encoder-decoder framework and Lukovinikov et al's complex character and word-level encoding. Our model is however outperformed by Dai et al and Yin et al which apply a separately trained segmentation. Ture and Jojic [18] reported a much higher accuracy, we are not able to replicate their results since they do not release their source code due to company restrictions as mentioned

Table 8 Different models end-to-end test accuracy compared previous results mentioned in the related work

Our Approach		
Entity	**Relation**	**Accuracy**
BiLSTM	BiGRU	74.64
BiLSTM	CNN	74.63
BiLSTM	BiLSTM	74.59
BiGRU	BiGRU	74.54
BiGRU	CNN	73.92
crf	CNN	73.42
crf	BiLSTM	73.34
crf	BiGRU	73.39
Previous approaches		
Model	**Description**	
Ture and Jojic (2017) [18]	LSTM and GRU	88.3
Yin et al (2016) [16]	attentive max-pooling	76.4
Dai et al (2016) [15]	conditional probabilistic framework	75.7
Lukovinikov et al (2017) [14]	end-to-end learning using neural embeddings	71.2
Golub & He (2016) [13]	character based encorder-decorder	70.9
Bodes et al (2015) [6]	memory network	62.7

by the author when I contacted via email. Besides that, our best accuracy is less than 2 points away from the next highest reported result in the literature. When we replace BiLSTM with CRF for entity detection/linking the accuracy decrease by only 1.25 this shows that non neural network baselines can still perform well. Despite the immense contribution of neural networks to the meaningful improvements in the state of the art on the simple questions dataset, our results suggest that the improvements directly attributed to neural networks are modest than previous researchers may have led the readers to believe.

It is important to pay attention when interpreting the results in table 4 due to non-determinism associated with training neural networks that can yield differences in accuracy Reimers and Gurevych [28]. It was also demonstrated that for answer selection in question answering, issues ranging from software versions, can significantly impact the accuracy Crane [29].

5 Conclusion

In this work we explore simple yet effective approach for simple question answering. Our baseline NNs are less complex than the previous models described in the related work. Moving forward we are more interested in formally decomposing the simple question answering task in sub-problems of entity detection/linking and relation prediction and solve each separately.

In conclusion, some points to consider especially for the simple question answering over knowledge graphs, there is still need to adequately examine simple baselines rigorously before rushing to sophisticated deep learning techniques. secondary our deep learning results are exciting but it is also necessary to consider none neural network baselines since natural language processing existed before deep learning. Also as pointed out by Sculley [8] it is important to remember that the ultimate goal of science is knowledge and not owning the top entry in a leader board. Finally even though deep learning has opened potential for more generic solutions, taking advantage of the problem structure yields not only accurate but also efficient results.

Acknowledgments

This research was supported by 平成 30 年度共同研究 (SKY 株式会社) (CPE30034))「データエンジニアリングの知見の応用による SKYSEA Client View のログ及び資産情報の処理の高速化・軽量化・高度化」の助成を受けたものです。

References

[1] S. Quarteroni, "A chatbot-based Interactive Question Answering System," Proceedings of the 11th workshop on the Semantics and Pragmatics of Dialogue, pp. 83-90, 2007.

[2] D. A. Furrucci, "Introduction to this is watson" IBM Journal of Research and Development, vol. 56, no. 3.4, pp. 1-1, 2012.

[3] O. Irsoy and C. Cardie, " Opinion Mining with Deep Recurrent Nueral Networks," Proceedings of the conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1746-1751, 2014.

[4] Y.Kim, "Convolutional Neural Networks for sentence Classification," Proceedings of the conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 720-728, 2014.

[5] D.Chen and C.Manning, "A Fast and Accurate Dependency Parser using Neural Networks," Proceedings of the conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 740-750, 2014.

[6] Antoine Bordes, Sumit Chopra and Jason Weston, "Large-scale simple question answering with memory networks," (CoRR), 2015.

[7] H.Bast, F.Burle, B.Buchhold, E.Haumann, "Easy Access to the Freebase Dataset," Proceedings of the 23rd International Conference on World Wide Web (WWW), pp. 95-98, 2014.

[8] D. Sculley, J. Snoek, A. Rahimi, A. Wiltschko, "WINNER'S CURSE?, ON PACE, PROGRESS, AND EMPIRICAL RIGOR," Proceedings of the 6th International Conference on Learning Representations, Workshop Track (ICLR) 2018.

[9] G. Melis, C. Dyer, P. Blunsom, "On the State of the Art Evaluation in Neural Language Models," ICLR, 2018.

[10] Lappoon R. Tang and Raymond J. Mooney, "Using Multiple Clause Constructors in Inductive Logic Programming for Semantic Parsing," Proceedings of the 12th European Conference on Machine Learning (ECML), pp. 466-477, 2001.

[11] T. Kwiatkowski, E. Choi, Y. Artzi, L. Zettlemoyer, "Scaling Semantic Parsers with On-the-fly Ontology Matching," Proceedings of the Conference on Emperical Methods in Natural Language Processing (EMNLP), pp. 1545-1556, 2013.

[12] J. Berant, A. Chou, R. Frostig and P. Liang, "Semantic Parsing on Freebase from Question-Answer Pairs," Proceedings of the Conference on Empirical Methodes in Natural Language Processing (EMNLP), pp. 1533-1544, 2013.

[13] D. Golub and X. He, "Character-Level Question Answering with Attention," Proceedings of the conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1598-1607, 2016.

[14] D. Lukovinikov, A. Fisher, J. Lehmann, and S. Auer, "Neural Network-based Question Answering over Knowledge Graphs on Word and Character Level," Proceedings of the 26th International Conference on World Wide Web

(WWW), pp. 1211-1220, 2017.

[15] Z. Dai, L. Li, and W. Xu, "CFO: Conditional Focused Neural Question Answering with Large-scale Knowledge Bases," Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 800-810, 2016.

[16] W. Yin, M. Yu, B. Xiang, B. Zhou, and H. Schutze, "Simple Question Answering by Attentive Convolutional Neural Network," Proceedings of the 26th International Conference on Computational Linguistics (COLING), pp. 1746-1756, 2016.

[17] M. Yu, W. Yin, K.S. Hasan, C. dos Santos, B. Xiang, and B. Zhou, "Improved Neural Relation Detection for Knowledge Base Question Answering," Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), pp. 571-581, 2017.

[18] F. Ture and O. Jojic, "No Need to Pay Attention: Simple Recurrent Neural Networks Work! (for Answering "Simple" Questions)" Proceedings of the conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 2866-2872, 2017.

[19] A. Graves and J. Schmidhuber, "Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures," Neural Network, pp. 602-610, 2005.

[20] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, "Learning Phrase Representation using RNN Encoder–Decorder for statistical Machine Translation," Proceedings of the conference on Empirical Methods in Natural Language Processing (EMNLP), pp.1724-1734, 2014.

[21] J. Rose Finkel, T. Grenager, and C. Manning, "Incoporating non-local information extraction systems by Gibbs sampling," Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL), pp. 363-370, 2005.

[22] Y. Kim, "Convolutional Neural Networks for Sentence Classification," Proceedings of the conference on Empirical Methods in Natural Language Processing (EMNLP), pp.1746-1751, 2014.

[23] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, J. Taylor, "Freebase: a collectively created graph database for structuring human knowledge," Proceedings of the 2008 ACM SIGMOD international conference on Management of Data, pp. 1247-1250, 2008.

[24] J. Pennington, R. Socher, and C. Manning, "Glove: Global Vectors for Word Representation," Proceedings of the conference on Empirical Methods in Natural Language Processing (EMNLP), pp.1532-1543, 2014.

[25] M. Sahlgren. "The distributional hypothesis." Italian Journal of Linguistics, pp.33-54, 2008.

[26] Diederik P. Kingma, J. Lei Ba, "ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION," International Conference On learning Representation (ICLR), 2015.

[27] Jenny R. Finkel, Christopher D. Manning, "Nested named entity recognition," Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp.141-150, 2009.

[28] N. Reimers and I. Gurevych, "Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging," Proceedings of the Conference on Empirical Methodes in Natural Language Processing (EMNLP), pp. 338-348, 2017.

[29] M. Crane, "Questionable Answers in Question Answering Research: Reproducibility and Variability of Published Results," Transactions of the Association for Computational Linguistics (ACL), pp. 241-252, 2018.