HybQA: Hybrid Deep Relation Extraction for Question Answering on Freebase

Reham Mohamed, Nagwa El-Makky and Khaled Nagi
Department of Computer and Systems Engineering, Alexandria University, Egypt

Keywords: Question Answering, Relation Extraction, Linked Data.

Abstract: *Question Answering* over knowledge-based data is one of the most important *Natural Language Processing* tasks. Despite numerous efforts that have been made in this field, it is not yet in the mainstream. *Question Answering* can be formulated as a *Relation Extraction* task between the question focus entity and the expected answer. Therefore, it requires high accuracy to solve a dual problem where the relation and answer are unknown. In this work, we propose a *HybQA*, a *Hybrid Relation Extraction* system to provide high accuracy for the *Relation Extraction* and the *Question Answering* tasks over Freebase. We propose a hybrid model that combines different types of state-of-the-art deep networks that capture the relation type between the question and the expected answer from different perspectives and combine their outputs to provide accurate relations. We then use a joint model to infer the possible relation and answer pairs simultaneously. However, since *Relation Extraction* might still be prone to errors due to the large size of the knowledge-base corpus (Freebase), we finally use evidence from Wikipedia as an unstructured knowledge base to select the best relation-answer pair. We evaluate the system on WebQuestions data and show that the system achieves a statistical significant improvement over the existing state-of-the-art models and provides the best accuracy which is 57%.

1 INTRODUCTION

With the massive increase of structured data on the web, as Freebase, DBpedia, Yago, etc, these data sources can be considered one of the most important and accurate sources of information to serve users' needs. However, querying these knowledge-based (KB) sources requires query languages such as SPARQL, which is difficult for an ordinary user to use. Therefore, there is a great need of providing accurate natural-language question answering systems to answer users questions over structured knowledge-based data. Generally, natural languages are complex and ambiguous, where there are no specific rules or terminology is restricted on the users. Moreover, natural languages are more prone to human errors, such as: grammatical, spelling or punctuation mistakes. This makes the problem of understanding and answering natural language questions more challenging and requires many steps to account for these challenges.

Recently, a lot of systems have emerged to provide Question Answering (QA) over structured and unstructured data. Unstructured QA systems such as: (Cui et al., 2005; Kaisser, 2012; Abdelnasser et al., 2014) are text-based systems where there are no restricted rules on the arrangement of data. These systems depend on Information Retrieval (IR) techniques to extract answers. While structured question answering systems such as: (Jain, 2016; Yih et al., 2015; Xu et al., 2016) depend on structured data sources and relational networks such as: DBpedia (Auer et al., 2007), Freebase (Bollacker et al., 2008), etc.

A structured knowledge-based question answering system is complex because it relies on other NLP techniques. Mainly, structured QA can be divided into two stages: (1) Question Analysis and (2) Answer Retrieval. Furthermore, the question analysis can be subdivided into other NLP problems, such as: text parsing, part-of-speech tagging (POS), dependency parsing, named-entity tagging, relation extraction, among others. The retrieval stage concerns with translating the extracted information from the question into a query language (as SPARQL) to retrieve the answer from the knowledge-base. This hierarchical chain of NLP stages causes more error propagation in the final output which contributes to more obstacles for accurate QA systems.

In this work, we focus on improving the accuracy of *Relation Extraction (RE)* for providing accurate structured question answering. In order to pro-

Mohamed R., M. El-Makky N. and Nagi K.
HybQA: Hybrid Deep Relation Extraction for Question Answering on Freebase.
DOI: 10.5220/0006589301280136
In *Proceedings of the 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (KEOD 2017)*, pages 128-136
ISBN: 978-989-758-272-1

vide a KB question answering, we need to construct a query with a focus entity that is mentioned in the question and a relation between this entity and the expected answer in the form of $(e, r, ?)$. However, in this scheme relation extraction is more complex because one entity is missing and only information about it is available such as: the answer type. This leads to a dual problem where the answer entity and the relation are missing. Some efforts have been proposed to solve this dual problem, such as: (Xu et al., 2016) which provided a new system for question answering over Freebase (FB) by extracting the relation between the question and the unknown answer using evidence from the question itself. The extracted relations are jointly detected with the entities in the question to predict a FB query to retrieve the answer.

We introduce *HybQA* a Hybrid-Deep Relation Extraction based system for Question Answering over Freebase. The system builds on previous efforts made in structured question answering (Xu et al., 2016) and proposes a new Hybrid Relation Extraction technique which combines different features to improve the accuracy of Relation Extraction. Our system builds several RE classifiers based on different types of features. Each classifier is built using a suitable Deep Network type with the parameters that best fit with these features. The outputs of these classifiers are then combined using a Hybrid Relation Ranker based on the accuracy of each classifier. Our goal is to mimic different human judgements of relations between entities by using Relation Extractors based on different variations of NNs.

More precisely, first the entities in the question are detected. Then for each entity the system extracts the most suitable relation between this entity and the expected answer. In order to extract this relation, *HybQA* builds three RE classifiers that use Lexical, Sentential and Dependency parse features. The classifiers are built using Convolution and Recursive Neural Networks. Each classifier works independently and in parallel with the other classifiers to extract the $topk$ relations that could be represented by the question and the unknown answer. The Hybrid Relation Ranker then combines the outputs of the classifiers by ranking the extracted relations using linear weights based on the classifier accuracy. The final output of the relation extraction module is a list of k ranked relations.

The ranked list resulting from the RE module is then fed to the rest of *HybQA* to provide a final answer to the input question from Freebase. First, the list is inferred with the entities in the question using Joint Inference, resulting in a set of ranked entity-relation pairs. Finally, the best entity-relation pair is selected using a Wikipedia Inference module, which extracts the answer using lexical based search in Wikipedia documents.

We evaluate *HybQA* using the WebQuestions data. Our results show that our hybrid RE approach outperforms the single RE classifier. We also show that the accuracy of our overall QA system is 57% which is better than the state-of-the-art QA systems.

In summary, the contributions of this paper are as follows:

1. Addressing the structured-QA problem by proposing a new *Relation Extraction* technique that aims to mimic different human opinions about the relation type by combining different types of features in three parallel *Deep Networks*, then integrate their outputs to reach one ranked list of relations.
2. Integration of the new RE scheme with the structured-based Question Answering framework to provide *HybQA* system, which processes natural questions and extracts candidate answer triples from Freebase, then uses text-based information retrieval to select the best candidate.
3. Implementation and evaluation of *HybQA* using WebQuestions data and comparison with the state-of-the-art systems, showing that *HybQA* provides the best answering on this data (57%).

The rest of this paper is organized as follows: Section 2 shows some of the related work to our system. Section 3 shows the corpora used in our system. Section 4 shows the details of the system architecture. In Section 5, we show the results of the system evaluation. Finally, we conclude the paper in Section 6.

2 RELATED WORK

Several attempts to solve the Question Answering problem have been proposed in literature. Over the years, QA has been of the main focus of NLP research due to its importance in the semantic web. QA systems can be divided into structured and unstructured systems. Unstructured systems are QA systems which try to solve questions against unstructured data sources, such as Wikipedia. In (Ravichandran and Hovy, 2002), question answering is done by retrieving information from textual data using surface text patterns. (Iyyer et al., 2014) uses RNNs to answer factoid questions over textual paragraphs. Some systems use semi-structured data sources, such as (Ryu et al., 2014) which uses some sources of Wikipedia, including article content, infoboxes, article structure, etc.

On the other hand, structured-based QA systems tend to answer user questions over structured linked data sources, as DBpedia, Freebase, etc. These data sources are usually formulated as triplets of two entities and a relationship between them (e_1, e_2, r). Recently, many approaches have been proposed in the structured QA domain. (Zhu et al., 2015) formulates the QA problem into two subproblems: Semantic Item Mapping which focuses on recognizing the semantic relation topological structure in Natural language question, and Semantic Item Disambiguation which instantiates these structures according to a given KB. (Usbeck et al., 2015) combines linked data from DBpedia and textual data from Wikipedia to form SPARQL and text queries which are used to retrieve answer entities. Linked and textual data are also combined in (Xu et al., 2016) which uses Relation Extraction to retrieve relevant answers from Freebase, then filters these answers using textual data from Wikipedia. In (Jain, 2016), Factual Memory Network are used which learn to answer questions by extracting and reasoning over relevant facts from a Knowledge Base. In (Aghaebrahimian and Jurcıcek, 2016), a different approach has been proposed which addresses open-domain question answering with no dependence on any data set or linguistic tool, using the Constrained Conditional Models (CCM) framework.

Some campaigns have also been constructed to evaluate question answering systems based on linked data, such as: Question Answering on Linked Data (QALD) (Dong et al., 2015), BioASQ (Tsatsaronis et al., 2012; Balikas et al., 2015), TREC LiveQA (Agichtein et al., 2015), among others. Neural Networks have proven good accuracy for the structured QA task. Most of the recent systems rely on different variations of Neural Networks, such as (Xu et al., 2016; Jain, 2016; Dong et al., 2015; Bordes et al., 2015). Despite these efforts, structured knowledge-based question answering is not yet in the mainstream. This is because QA systems rely on a chain of NLP tasks which might have low accuracy.

In this paper, we focus on the Relation Extraction step as a critical building block in the structured QA architecture and the most challenging step. Therefore, we propose our hybrid RE model to provide better overall QA performance.

3 *HybQA* CORPORA

In this section, we describe the data corpora that we use to build our system.

3.1 Wikipedia

Wikipedia is one of the most commonly used resources in computational linguistics. The attraction to Wikipedia returns to its large size, its diversity and for being always up to date. The English version of Wikipedia is the largest Wikipedia dump. It has over 5M articles, and over 12% of the total Wikipedia articles belong to the English edition [1].

3.2 Freebase

Freebase is a large collaborative knowledge based data source, which was composed by Freebase community members. It was developed by the American software company Metaweb, which was then acquired by Google. Freebase is an online collection of structured data collected from many sources, such as: Wikipedia, NNBD, Fashion Model Directory, etc. Freebase offered an entity-relationship model and provided an interface that allowed ordinary users to fill data and connect data items in semantic ways. Moreover, a JSON-based API was provided to be used by programmers for commercial and non-commercial purposes [2].

Freebase was built using a graph model which is composed of nodes and links to represent the relation between nodes. Using this model, it could represent more complex relationships than a conventional database. The Freebase version we use in this paper is the version of (Berant et al., 2013) , which contains 4M entities and 5,323 relations. Its RDF triples were loaded into Virtuoso server.

4 SYSTEM ARCHITECTURE

The system architecture is shown in Figure 1. The system operates on the input question and links the entities in the question to Freebase entities using the *Entity Linking* module. Then, the *Hybrid Relation Extraction* module operates on the question to extract the relation between the topic entity in the question and the answer entity. This module is composed of three Relation Extraction approaches that work concurrently to extract the top-k relations, then their output is combined usng *Hybrid Relation Ranking*. The tagged entities and the extracted relations are then merged using *Joint Inference* module resulting in ranked candidate FB triples. Finally, the best answer is selected using *Wikipedia Inference* module which

[1] https://en.wikipedia.org/wiki/English_Wikipedia

[2] https://en.wikipedia.org/wiki/Freebase

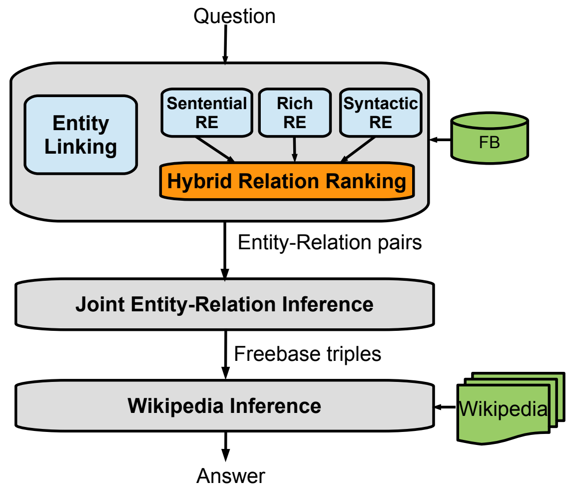


Figure 1: System architecture.

operates on the Wikipedia page of the topic entity to filter out incorrect relations using lexical search. In the next subsections, we discuss the different modules of the system architecture in detail.

4.1 Entity Linking

Entity linking is done using S-MART[3] entity linking tool (Yang and Chang, 2016). The tool takes entity spans marked by POS tags and retrieves the top 5 matching entities from Freebase. Entity spans are identified by hand-built POS sequences (Bao et al., 2014). S-MART is a tree-based structured learning framework based on multiple additive regression trees. It first uses surface matching to retrieve all possible entities of Freebase, and then ranks them using a statistical model.

4.2 Relation Extraction

For the Relation Extraction task, we use several classifiers which add different levels of features. All classifiers are based on variations of Neural Networks. The output of the classifiers is then combined in the Hybrid Relation Ranking module. The goal is to mimic different human judgements for relations modelling different features in the most suitable NN classifier. The approach is illustrated in Figure 2.

4.2.1 Sentential Relation Extractor

The first Relation Extractor encodes sentential features represented as the words in the question excluding the question word and the entity mention. Their word vectors are convoluted using CNN to give one vector representation of the sentence. In relation classification, we concentrate on learning discriminative word embeddings, which carry more syntactic and semantic information. This classifier utilizes word embeddings of the words in the question to capture the sentence words information that discriminates the relation. It uses a Convolution Neural Network (CNN) where the convolution layer tackles the input of varying length returning fixed length vectors which are then fed into softmax classifier. This simple RE approach have proven to be effective in extracting relations and therefore is built separately to exploit this discriminative feature and to avoid noise caused by increasing dimensionality.

4.2.2 Rich Relation Extractor

The second Relation Extractor combines more rich lexical and sentence level clues from diverse syntactic and semantic structures in a sentence. In this model, we use a convolutional DNN to extract lexical and sentence level features for relation classification. The extractor first takes word tokens and transforms them into vector word embeddings. Then, extracts lexical-level features according to the given entities. Sentence level features are learnt according to a convolution module. The two level features are concatenated and fed into a Softmax classifier to predict the relationship between two marked entities. This classifier adds the Position Encoding feature which is used to measure the distance between the two entities. In our system, we use the output of the entity linking task and the question word to mark the two related nouns (entities), provided that a question as *"who played Y role?"* is expected to be answered with *"X played Y role"*. Since we do not have the answer phrase, we assume that the entity X is substituted by the question word *"Who" in the question*, so we mark the two nouns in the question *"Who"* and *Y*. The system extracts the features based on these two nouns.

1. ***Lexical-level Features:*** represented by the word embeddings: question word, entity, right tokens of question word, left and right tokens of the entity, in addition to hypernyms from WordNet.

2. ***Sentence-level Features:*** obtained by a max pooled convolutional neural network. Each token is represented as word features (WF) and Positional Features (PF), then convolution is applied and sentence level features are obtained by non-linear transformation.

   The *Word features* (WF) combines a word's vector representation and the vector representations of the words in its context, which most probably have related meanings.

[3] http://msre2edemo.azurewebsites.net/

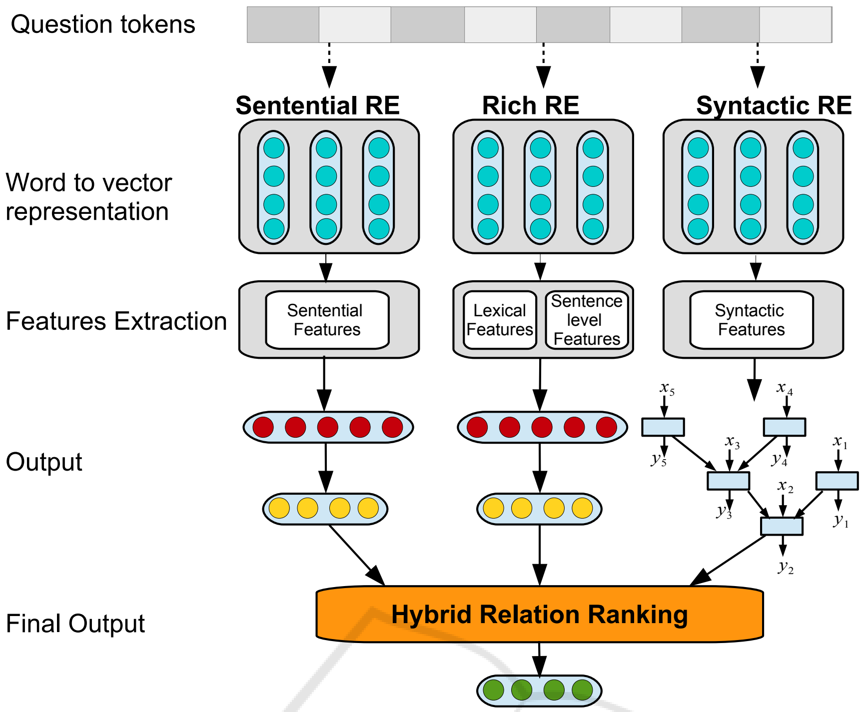


Figure 2: Illustration of the Hybrid Relation Extraction.

The *Position features* (PF) is used to capture the position of the word in the sentence with respect to the given entities. Thus, for each word the relative distance is calculated from the two entities. Each distance value is mapped to a vector randomly initialized with dimension d_e forming two vectors: d_1 and d_2 which are combined into $PF = [d_1, d_2]$.

These local features of each word are merged using a convolutional neural network to get a global feature vector.

4.2.3 Syntactic Relation Extractor

There are two ways to represent relations between entities using neural networks: recurrent/recursive neural networks (RNNs) and convolutional neural networks (CNNs). RNNs can directly represent essential linguistic structures, and dependency trees (Tai et al., 2015). In this RE classifier, we combine word sequence and dependency parse tree features to extract relations between entities. Like the previous classifier, entities must be tagged to get the dependency path between them. We substitute the first entity with the question word and the second entity is detected in the entity linking module.

The dependency parse is used to represent the relation between two target entities. This classifier forms a NN layer using the shortest path between a pair of target words in the dependency tree which are shown to be effective in relation classification (Xu et al., 2015). The classifier employs bidirectional tree-structured LSTM- RNNs (i.e., bottom-up and top-down) to represent a relation candidate. The relation candidate vector is constructed as the concatenation $d_p = [\uparrow \text{hpA}; \downarrow \text{hp1}; \downarrow \text{hp2}]$, where $\uparrow$ hpA is the hidden state vector of the top LSTM unit in the bottom-up LSTM-RNN (representing the lowest common ancestor of the target word pair p), and $\downarrow$ hp1 , $\downarrow$ hp2 are the hidden state vectors of the two LSTM units representing the first and second target words in the top-down LSTM- RNN.

The text is parsed using Stanford neural dependency parser [4].

4.2.4 Hybrid Relation Ranking

The outputs of the three relation extractors are combined to select the best relation candidates. For each relation extractor RE$_j$, the top k relations are selected. The weight of a relation r_i is computed as follows:

$$p(r_i) = w_1.p_1(r_i) + w_2.p_2(r_i) + w_2.p_3(r_i) \quad (1)$$

Where $p_j(r_i)$ is the probability of relation r_i using the j^{th} relation extractor and w_j's are linear weights proportional to the relation extractors accuracy.

[4] http://nlp.stanford.edu/software/stanford-corenlp-full-2015-04-20.zip

4.3 Joint Entity-Relation Inference

Detecting entities and relations separately is more error prone. In this module, we jointly detect entity-relation pairs that are globally optimal. We follow the approach proposed in (Xu et al., 2016). The pairs of entities and relations are formed $\{(e_1,r_1),(e_2,r_1),..,(e_n,r_m)\}$ and ranked using SVM Rank classifier. Scores for the training data is created as follows: if both entity and relation are correct, the score is 3. If only one of them is correct, the score is 2, if both are wrong, the score is 1. A set of features are extracted to train the classifier, including: (a) Entity features, such as the score of each entity, number of overlapping words with FB entity, number of word overlap between the question and the entity description. (b) Relation features, such as: the relation score from the Hybrid Relation Ranking, the sum of the tf-idf scores of the question words with respect to the relation. (c) Answer features: the result of the query $(e,r,?)$ is used as the answer, features from the answer are such as: matching between answer type and question word.

4.4 Wikipedia Inference

The Joint Entity-Relation Inference outputs triples with entities, relations and the retrieved candidate answers from FB. To filter the incorrect triples, Wikipedia is used as a source of evidence. The system retrieves the Wikipedia page of the topic entity and searches for sentences containing the candidate answers. A binary classifier is used to tag each candidate answer as correct or incorrect. The features used are the set of all possible pairs of words in the question q_i and words in the sentence s_j in the form of (q_i,s_j) for all values of i,j.

5 PERFORMANCE EVALUATION

5.1 Training and Testing Data

We use the WebQuestions (Berant et al., 2013) dataset. The dataset is composed of 5,810 questions crawled via Google Suggest service. The answers are annotated on Amazon Mechanical Turk. The questions are split into training and test sets, which contain 3,778 questions (65%) and 2,032 questions (35%), respectively. We use the same data split for our training and testing tasks. Furthermore, 20% of the training set is used as the development set.

For the relation extraction task, since gold-relations are not provided with the dataset, we use the surrogate gold-relations as shown in (Xu et al., 2016), which are the relations that produce answers with highest overlap with the gold-answers. For each question, the 1-hop and 2-hop relations connected to the topic entity are selected as relation candidates. For each relation candidate, FB is queried using the this relation and the topic entity to get the answer. The relation that produces the answer with minimal F1-loss against the gold answer, is selected is the surrogate gold-relation.

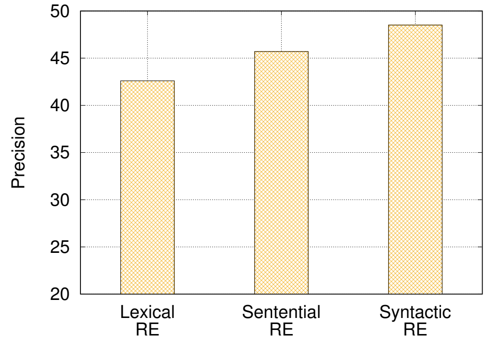


Figure 3: Effect of Relation Extractors.

5.2 Experimental Setup

We use the Freebase version of Berant et al. (Berant et al., 2013), containing 4M entities and 5,323 relations. We use the word embeddings of Turian et al. (Turian et al., 2010).The system was implemented on an Amazon EC2 Ubuntu instance with 32 GB RAM.

5.3 Results

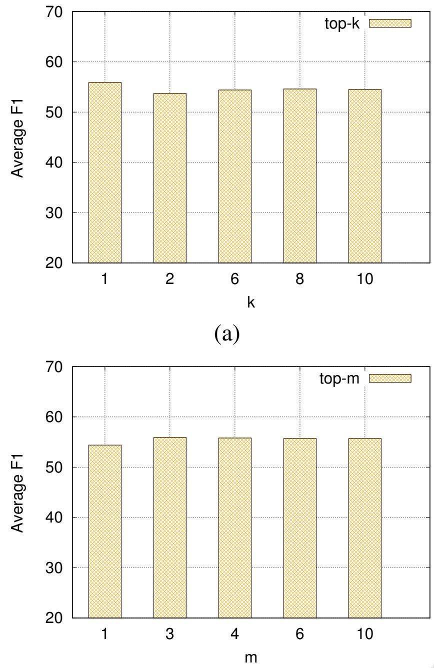
We evaluate the system using WebQuestions data. We use the average-F1 as the evaluation metric.

5.3.1 Effect of Hybrid Relation Extraction

Figure 3 shows the accuracy of each relation extractor on the relation extraction task using top-1 relation. We tag the training and test data with the surrogate gold-relations, and use them for evaluation. The sentential and syntactic extractors rely on the output of the entity linking, therefore they are more exposed to error propagation. Also, the syntactic extractor relies on the output of the dependency parser which further increases the error. Combining the three extractors improves the accuracy to 49% as it captures the relation from different perspectives.

5.3.2 Effect of Joint Inference Parameters

In the Joint Entity-Relation Inference module, we select the *top-k* relations and *top-m* entities which are



(a)

(b)

Figure 4: Effect of Joint Entity-Relation Parameters: (a) Effect of using *top-k* relations. (b) Effect of using *top-m* entities.

jointly ranked using SVM Rank classifier. We show the accuracy of the system with changing the value of k and m in Figure 4. The best accuracy is achieved with $k = 1$ and $m = 3$. This shows that our Hybrid Relation Extraction model is good enough to provide the best relation, where introducing other relations degrades the accuracy.

5.4 Overall Performance Evaluation

In this section, we compare *HybQA* with a baseline model that uses one CNN relation extractor to perform QA. Figure 5 compares the accuracy and the latency of *HybQA* and the baseline model. We show that since *HybQA* uses parallel NNs, it does not drastically increase the latency. The small delay in the latency is due to using the tree-based LSTM-RNN. However, using this network is essential as it best fits the dependency features. This delay could be improved by implementing the RNN on GPU. This also results in a significant improvement in accuracy with t-test of $p < 0.05$.

5.4.1 Comparison with other Systems

Table 1 compares our system with the related systems. We compare four variations of our system: using lexical features only for relation extraction, using lexical

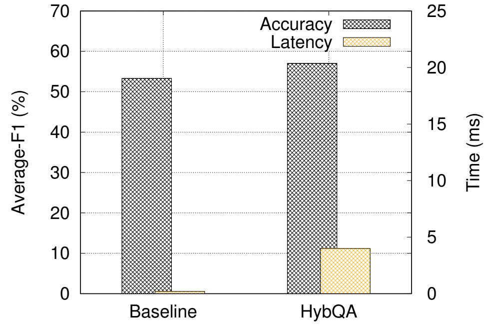


Figure 5: Overall performance of *HybQA* compared with a baseline QA system with CNN relation extractor.

features with sentential features, using lexical features with syntactic features and the Hybrid RE which uses the three types of features. Our system outperforms the other variations and achieves the best accuracy for the WebQuestions data (57%).

Table 1: Comparison with other systems.

Setup	**Average-F1 measure**
Berant et al. (Berant et al., 2013)	35.7
Yao and Van Durme (Yao and Van Durme, 2014)	33.0
Xu et al. (Xu et al., 2014)	39.1
Berant and Liang (Berant and Liang, 2014)	39.9
Bao et al. (Bao et al., 2014)	37.5
Bordes et al. (Bordes et al., 2015)	39.2
Dong et al. (Dong et al., 2015)	40.8
Yao (Yao, 2015)	44.3
Bast and Haussmann (Bast and Haussmann, 2015)	49.4
Berant and Liang (Berant and Liang, 2015)	49.7
Reddy et al. (Reddy et al., 2016)	50.3
Yih et al. (Yih et al., 2015)	52.5
Xu et al. (Xu et al., 2016)	53.3
Jain (Jain, 2016)	55.6
This work	
Lexical RE	49.8
Lexical+Sentential RE	53.8
Lexical+Syntactic RE	53.4
Hybrid RE	**57**

6 CONCLUSION

In this paper, we proposed a novel structured-based Question Answering system over Freebase. Our system builds a new *Hybrid Relation Extraction* model that combines different types of NNs to model different features resulting in a ranked list of candidate relations. The system then infers the list of relations with the question entities to produce candidate *relation-entity* pairs. The best of these pairs is then selected using lexical-based search in unstructured data.

Our experimental results on WebQuestions dataset show that the system achieves 57% average-F1 accuracy which outperforms the state-of-the-art systems.

REFERENCES

Abdelnasser, H., Ragab, M., Mohamed, R., Mohamed, A., Farouk, B., El-Makky, N., and Torki, M. (2014). Al-bayan: an arabic question answering system for the holy quran. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 57–64.

Aghaebrahimian, A. and Jurcıcek, F. (2016). Open-domain factoid question answering via knowledge graph search. In *Proceedings of the Workshop on Human-Computer Question Answering, The North American Chapter of the Association for Computational Linguistics*.

Agichtein, E., Carmel, D., Pelleg, D., Pinter, Y., and Harman, D. (2015). Overview of the trec 2015 liveqa track. In *TREC*.

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). Dbpedia: A nucleus for a web of open data. *The semantic web*, pages 722–735.

Balikas, G., Kosmopoulos, A., Krithara, A., Paliouras, G., and Kakadiaris, I. (2015). Results of the bioasq tasks of the question answering lab at clef 2015. In *CLEF 2015*.

Bao, J., Duan, N., Zhou, M., and Zhao, T. (2014). Knowledge-based question answering as machine translation. *Cell*, 2(6).

Bast, H. and Haussmann, E. (2015). More accurate question answering on freebase. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1431–1440. ACM.

Berant, J., Chou, A., Frostig, R., and Liang, P. (2013). Semantic parsing on freebase from question-answer pairs. In *EMNLP*, volume 2, page 6.

Berant, J. and Liang, P. (2014). Semantic parsing via paraphrasing. In *ACL (1)*, pages 1415–1425.

Berant, J. and Liang, P. (2015). Imitation learning of agenda-based semantic parsers. *Transactions of the Association for Computational Linguistics*, 3:545–558.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). 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.

Bordes, A., Usunier, N., Chopra, S., and Weston, J. (2015). Large-scale simple question answering with memory networks. *arXiv preprint arXiv:1506.02075*.

Cui, H., Sun, R., Li, K., Kan, M.-Y., and Chua, T.-S. (2005). Question answering passage retrieval using dependency relations. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 400–407. ACM.

Dong, L., Wei, F., Zhou, M., and Xu, K. (2015). Question answering over freebase with multi-column convolutional neural networks. In *ACL (1)*, pages 260–269.

Iyyer, M., Boyd-Graber, J. L., Claudino, L. M. B., Socher, R., and Daumé III, H. (2014). A neural network for factoid question answering over paragraphs. In *EMNLP*, pages 633–644.

Jain, S. (2016). Question answering over knowledge base using factual memory networks. In *Proceedings of NAACL-HLT*, pages 109–115.

Kaisser, M. (2012). Answer sentence retrieval by matching dependency paths acquired from question/answer sentence pairs. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 88–98. Association for Computational Linguistics.

Ravichandran, D. and Hovy, E. (2002). Learning surface text patterns for a question answering system. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 41–47. Association for Computational Linguistics.

Reddy, S., Täckström, O., Collins, M., Kwiatkowski, T., Das, D., Steedman, M., and Lapata, M. (2016). Transforming dependency structures to logical forms for semantic parsing. *Transactions of the Association for Computational Linguistics*, 4:127–140.

Ryu, P.-M., Jang, M.-G., and Kim, H.-K. (2014). Open domain question answering using wikipedia-based knowledge model. *Information Processing & Management*, 50(5):683–692.

Tai, K. S., Socher, R., and Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*.

Tsatsaronis, G., Schroeder, M., Paliouras, G., Almirantis, Y., Androutsopoulos, I., Gaussier, E., Gallinari, P., Artieres, T., Alvers, M. R., Zschunke, M., et al. (2012). Bioasq: A challenge on large-scale biomedical semantic indexing and question answering. In *AAAI fall symposium: Information retrieval and knowledge discovery in biomedical text*.

Turian, J., Ratinov, L., and Bengio, Y. (2010). Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics.

Usbeck, R., Ngomo, A.-C. N., Bühmann, L., and Unger, C. (2015). Hawk–hybrid question answering using linked data. In *European Semantic Web Conference*, pages 353–368. Springer.

Xu, K., Feng, Y., Huang, S., and Zhao, D. (2015). Semantic relation classification via convolutional neural networks with simple negative sampling. *arXiv preprint arXiv:1506.07650*.

Xu, K., Reddy, S., Feng, Y., Huang, S., and Zhao, D. (2016). Question answering on freebase via relation extraction and textual evidence. *Proceedings of the Association for Computational Linguistics (ACL 2016)*.

Xu, K., Zhang, S., Feng, Y., and Zhao, D. (2014). Answering natural language questions via phrasal semantic parsing. In *Natural Language Processing and Chinese Computing*, pages 333–344. Springer.

Yang, Y. and Chang, M.-W. (2016). S-mart: Novel tree-based structured learning algorithms applied to tweet entity linking. *arXiv preprint arXiv:1609.08075*.

Yao, X. (2015). Lean question answering over freebase from scratch. In *HLT-NAACL*, pages 66–70.

Yao, X. and Van Durme, B. (2014). Information extraction over structured data: Question answering with freebase. In *ACL (1)*, pages 956–966. Citeseer.

Yih, S. W.-t., Chang, M.-W., He, X., and Gao, J. (2015). Semantic parsing via staged query graph generation: Question answering with knowledge base.

Zhu, C., Ren, K., Liu, X., Wang, H., Tian, Y., and Yu, Y. (2015). A graph traversal based approach to answer non-aggregation questions over dbpedia. In *Joint International Semantic Technology Conference*, pages 219–234. Springer.