Hybrid Question Answering over Knowledge Base and Free Text

Kun Xu[1], **Yansong Feng**[1,*], **Songfang Huang**[2] and **Dongyan Zhao**[1]
[1]Institute of Computer Science & Technology, Peking University, Beijing, China
[2]IBM China Research Lab, Beijing, China
{xukun, fengyansong, zhaody}@pku.edu.cn
huangsf@cn.ibm.com

Abstract

Recent trend in question answering (QA) systems focuses on using structured knowledge bases (KBs) to find answers. While these systems are able to provide more precise answers than information retrieval (IR) based QA systems, the natural incompleteness of KB inevitably limits the question scope that the system can answer. In this paper, we present a hybrid question answering (hybrid-QA) system which exploits both structured knowledge base and free text to answer a question. The main challenge is to recognize the meaning of a question using these two resources, i.e., structured KB and free text. To address this, we map relational phrases to KB predicates and textual relations simultaneously, and further develop an integer linear program (ILP) model to infer on these candidates and provide a globally optimal solution. Experiments on benchmark datasets show that our system can benefit from both structured KB and free text, outperforming the state-of-the-art systems.

1 Introduction

Recently, with the emergence of large structured knowledge bases (KBs) like DBpedia (Auer et al., 2007), Freebase (Bollacker et al., 2008) and Yago (Suchanek et al., 2007), increasing research efforts on automatically answering natural language questions has shifted from using text corpora only to large scale structured KBs like DBpedia, Freebase (known as KB-QA). Compared to pure text resources used in IR-based QA systems, structured knowledge bases may help to provide users with more accurate and concise answers, especially for factoid questions.

Generally, the traditional KB-QA paradigm assumes that world knowledge can be encoded using a closed vocabulary of formal predicates. In this paradigm, the system is given a knowledge base as input, and the question answering problem reduces to semantic parsing, i.e., mapping from text to logical forms containing the predicates from the given knowledge base. However, the closed predicate vocabulary assumed by the traditional KB-QA paradigm has inherent limitations. First, a closed predicate vocabulary has limited coverage, as such vocabularies are typically powered by community efforts. Second, a closed predicate vocabulary may abstract away potentially relevant semantic differences. Third, even a logical form was produced, the answers may be incomplete due to the imperfection of the KB, which has been addressed by (Riedel et al., 2013; Chen et al., 2014). For example, no logical form could be produced for the question *who is the front man of the band that wrote Coffee & TV*. Because the semantics of *front man* cannot be adequately encoded using Freebase or DBpedia predicates.

On the other hand, knowledge bases like DBpedia capture real world facts, and web resources like Wikipedia may provide a large repository of sentences that complement those facts. For instance, we can find in Wikipedia a sentence *In August 2009, Debelle performed at Africa Express in Paris, an event set up by Blur and Gorillaz front-man Damon Albarn*, which indicates the front man of the band in the example question is *Damon Albarn*[1]. Moreover, text corpora is also shown effective in refining the answers retrieved from the KBs (Xu et al., 2016). Motivated by these observations, we tackle the

[1]The Blur band wrote the Coffee & TV song.

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 2397–2407, Osaka, Japan, December 11-17 2016.

question answering task by integrating these two types of heterogeneous data, i.e., structured knowledge bases and free text, while is rarely investigated before.

This task involves three main challenges. The first is how to represent the meaning of a question by the clues from two types of heterogeneous resource. Secondly, for each phrase, there exist multiple grounded candidates over the KB and text corpora, how to perform inference on these candidates itself is a problem. The third challenge is how to properly incorporate the coherence of two types of heterogeneous resource, KB predicates and textual relations, into the inference model.

In this paper, we propose a joint inference approach to simultaneously solve these disambiguations. Specifically, our method consists of two main steps as outlined in (§2). In the first step, we employ preliminary models to perform the entity linking and relation extraction (§3). Next, we develop an integer linear program (ILP) model, where the candidate mapping of phrases to KB items and textual relations are the variables restricted by several designed constraints, and they could be determined simultaneously through joint inference (§4). The main contributions of this paper are two folds:

- We introduce a new task paradigm of the question answering community, and present a novel hybrid-QA framework to accommodate the structured KB and free text.
- We propose a joint inference model to solve the disambiguation among entities and relations across text and KBs.

Our evaluation results on benchmark datasets show that our system benefits from the integration of the KB and free text outperforming the state-of-the-art systems.

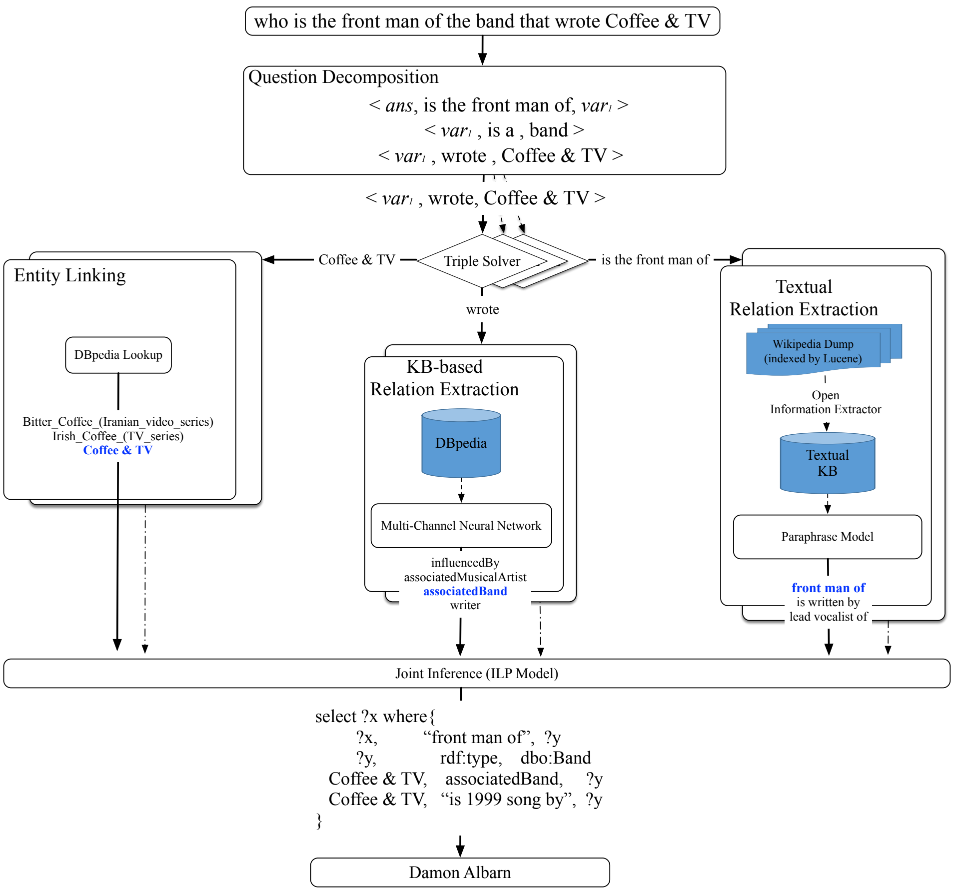

Figure 1: A running example of our hybrid-QA system for the question *who is the front man of the band that wrote Coffee & TV*, where the blue annotations are correct.

2 Our Method

Figure 1 gives an overview of our method for the aforementioned question "*who is the front man of the band that wrote Coffee & TV*". We have two main steps: (1) perform the local predictions, i.e., Entity Linking (EL) and Relation Extraction (RE); and (2) further infer on the retained candidate entities, KB predicates and textual relations to find an optimal assignment under certain constraints.

Let us take a close look into step 1. Here we first perform entity linking to identify possible KB entities in the question. Then we employ two types of relation extractors to predict both KB predicates and textual relations existing between two entities or question word and entities in the question. Specifically, we propose a neural network based method to map relational phrases to KB predicates, and apply a paraphrase model to find most likely textual relations that describe the phrases. In Step 2, we perform a joint inference over the local predictions of EL and RE models to find a best configuration through an ILP model.

As shown in Figure 1, it is often the case that a question may involve multiple relations. Consider the example question, the answers of this question should satisfy the following two constraints: (1) the person is the front man of a band (textual relation); and (2) the band wrote the song *Coffee & TV* (KB predicate). We use the 6 syntax-based rules as introduced in (Xu et al., 2016) to preprocess such *multi-relational* questions, i.e., decomposing them into a set of simple questions formulated as *ungrounded* triples. For instance, the example question can be decomposed into three ungrounded triples: $<$*ans*, *is the front man of*, $var_1$$>$, $<$$var_1$, *is a*, *band* $>$ and $<$$var_1$, *wrote*, *Coffee & TV*$>$[2].

3 Prelimiary Models

Since we represent the meaning of a question using clues from two types of heterogeneous resources, we tackle the QA problem in an IE-based fashion involving entity linking and relation extraction. In particular, we simultaneously map relational phrases to KB predicates and textual relations.

3.1 Entity Linking

The preliminary entity linking model can be any approach which outputs a score for each entity candidate. Note that a recall-oriented model will be more than welcome, since we expect to introduce more potentially correct local predictions into the inference step. In this paper, we adopt DBpedia Lookup[3] and S-MART (Yang and Chang, 2015) to retrieve top 10 entities from DBpedia and Freebase, respectively. These entities are treated as candidate entities that will be eventually disambiguated in the joint inference step.

3.2 KB-based Relation Extraction

The choice of KB-based relation extraction model is also broad. In this paper, we employ the **M**ulti-**C**hannel **C**onvolutional **N**eural **N**etworks (MCCNNs) model presented in (Xu et al., 2016) to learn a compact and robust relation representation. This is crucial since there exist thousands of relations in a KB, using lexicalized features inevitably suffers from the sparsity problem and their poor generalization ability on unseen words (Gormley et al., 2015).

The MCCNN model treats the conjunction of three parts in a ungrounded triple as a sentence (**subject** relational phrase **object**). The first channel takes the shortest path between the subject and object in the dependency tree[4] as input, while the other channel takes the relational phrase itself as input. Each channel uses the network structure described in (Collobert et al., 2011), which uses a convolutional layer to project the word-trigram vectors of words within a context window of 3 words to a local contextual feature vector, followed by a max pooling layer that extracts the most salient local features to form a fixed-length. The global feature vector is then fed to feed-forward neural network layers to output the final non-linear semantic features, as the vector representation of the relational phrase.

[2]Here ans denotes the answer and var_1 denotes an intermediate variable.

[3]http://wiki.dbpedia.org/projects/dbpedia-lookup.

[4]We use Stanford CoreNLP dependency parser.

Learning The model is learned using pairs of relational phrase and its corresponding KB predicate. Given an input phrase, the network outputs a distribution vector over the predicates o. We denote t as the target distribution vector, in which the value for gold relation is set 1, others are set 0. We compute the cross entropy error between t and o as the loss function. The model parameters can be efficiently computed via back-propagation through network structures. In experiment, we train two distinct relation extractors over DBpedia and Freebase, respectively. For DBpedia, we use the PATTY dataset (Nakashole et al., 2012) which consists of 127,811 pairs of relational phrases and DBpedia predicates involving 225 DBpedia predicates. For Freebase, we use 3,022 phrase-predicate pairs of WEBQUESTIONS used in (Xu et al., 2016), which involves 461 Freebase predicates.

3.3 Open Relation Extraction

Despite huge amounts of precise knowledge facts, structured KBs still have natural limitation in the coverage of knowledge domains compared to the vast information on the web. For example, out of 500,000 relations extracted by the ReVerb Open IE system (Fader et al., 2011), only about 10,000 can be aligned to Freebase (Berant et al., 2013). To alleviate this problem, we propose a paraphrase based method that can map relational phrases to proper textual relations. Specifically, we first apply an open information extractor (Angeli et al., 2015) on the English Wikipedia to construct a repository of <*argument*$_1$, *relation*, *argument*$_2$> triples, where the *arguments* are entity phrases found in the input sentence and the *relation* represents certain relationship between the arguments. By linking these arguments to KB entities, we can obtain a textual knowledge repository.

Paraphrasing Once the candidate set of textual relations $TR = \{tr_1, tr_3, ..., tr_{|TR|}\}$ are constructed, given a relational phrase rp, our goal is to find the tr that has the same meaning as rp, which can be treated as a paraphrase task. Our framework accommodates any paraphrasing method, such as the method based on dynamic pooling and recursive autoencoders (RAE) (Socher et al., 2011), which we adopt in our framework. Generally, the RAEs are based on a novel unfolding objective and learn feature vectors for phrases in syntactic trees. These features are used to measure the word-wise and phrase-wise similarities between two sentences. Since sentences may be of arbitrary length, the resulting matrix of similarity measures is of variable size. Then a dynamic pooling layer is introduced to compute a fixed-sized representation from the variable-sized matrices. Finally the pooled representation is used as input to a classifier $\mathcal{C}_p$.

Learning In our experiment, we directly used the pre-trained RAE which is trained on a subset of 150,000 sentences from the NYT and AP sections of the Gigaword corpus. To train the classifier $\mathcal{C}_p$, we use the PARALEX corpus (Fader et al., 2013), which is a large monolingual parallel corpora, containing 18 million pairs of question paraphrases from `wikianswers.com`, which were tagged as having the same meaning by the users of the website.

4 Joint Inference

The goal of the inference step is to find a global optimal configuration of entity phrases and relational phrases with semantic components. As the result of disambiguating one phrase can influence the mapping of other phrases, we consider all phrases jointly in one disambiguation task. Now, we will first describe three key criteria that are used to evaluate the configuration in details.

KB Predicate and Entity's Coherence If the relational phrase rp is grounded to a KB predicate kr, we should examine whether the semantic types of the entities fulfill the expectations of KB predicates. Particularly, we first obtain the type of subject entity e, which is collected from the KB's schema, and examine whether there exists another entity with the same type taking the subject position of this predicate in the KB. If such an entity exists, it indicates this entity is compatible with the KB predicate, $Coh_{e,kr} = 1$, otherwise 0.

Textual Relation and Entity's Coherence Similarity, we also need to capture the coherence, $Coh_{e,tr}$, between a textual relation tr and entity e. Since the textual relation does not have well-defined schemas like the KB, we practically treat the types of collected entities that take the subject and object position of tr as the type expectations of tr. For instance, *written by* takes `Coffee&TV` (a song) and `Blur` (an English band), which indicates the type expectations of *written by* should include `Song` and `Band`. We then determine whether e is compatible with tr by examining whether the type of e fulfills the type expectations of tr. If e is compatible with tr, $Coh_{e,tr} = 1$, otherwise 0.

KB Predicate and Textual Relation's Coherence Notice that, we allow a relational phrase to be simultaneously mapped to a KB predicate and a textual relation. In this case, the KB predicate kr and textual relation tr should be compatible with each other. For this purpose, we first determine if kr and tr have the same argument expectations. If so, we use the trained MCCNN to capture the coherence of a KB predicate kr and textual relation tr, $Coh_{kr,tr}$. In practice, we treat this problem as a variant of relation classification, i.e., the coherence score is the probability of mapping word sequence tr to KB predicate kr. Otherwise, $Coh_{kr,tr}$ is set to -1.

Integer Linear Program Formulation Now we describe how we aggregate the above components, and formulate the joint inference problem into an ILP framework. Given the above definitions, our objective function is to maximize the score of entity linking, relation extraction and their coherence among them:

$$\max \quad \alpha \times conf^e + \beta \times conf^r + \delta \times conf^{er} \tag{1}$$

where α, β and δ are weighting parameters tuned on development set. $conf^e$ is the overall score of entity linking:

$$conf^e = \sum_d \sum_{ep \in d, e \in C_e(ep)} w_{ep,e} Y_{ep,e} \tag{2}$$

where d is the ungrounded triple, $C_e(ep)$ is the candidate entity set of the entity phrase ep, $w_{ep,e}$ is the entity linking score, and $Y_{ep,e}$ is a boolean decision variable that indicates if entity phrase ep maps to entity e. $conf^r$ represents the overall score of relation extraction:

$$conf^r = \sum_d \sum_{rp \in d, kr \in C_{kr}(rp)} q_{rp,kr} Z_{rp,kr} + \sum_d \sum_{rp \in d, tr \in C_{tr}(rp)} v_{rp,tr} W_{rp,tr} \tag{3}$$

where $C_{kr}(rp)$ is the set of candidate KB predicates of relation phrase rp, $C_{tr}(rp)$ is the set of candidate textual relations corresponding to rp, $q_{rp,kr}$ and $v_{rp,tr}$ are the scores of relational phrase rp mapped to KB relation kr and textual relation tr. We define two boolean decision variables $Z_{rp,kr}$ and $W_{rp,tr}$ to denote whether rp is mapped to kr and tr. coh^{er} evaluates the coherence between the candidate entities and relations in the framework:

$$conf^{er} = \sum_d \sum_e \sum_{kr} o_{e,kr} Coh_{e,kr} + \sum_d \sum_e \sum_{tr} o_{e,tr} Coh_{e,tr} + \sum_d \sum_{kr} \sum_{tr} o_{kr,tr} Coh_{kr,tr} \tag{4}$$

where $o_{e,kr}$, $o_{e,tr}$ and $o_{kr,tr}$ are the coherence scores among entities, KB predicates and textual relations. We introduce three boolean decision variables $Coh_{e,kr}, Coh_{e,tr}, Coh_{kr,tr}$ to denote whether two semantic components are both selected.

Constraints Now we describe the constraints used in our ILP problem. The first kind of constraints is introduced to ensure that each entity phrase should be disambiguated to only one entity:

$$\forall d, \forall e \in C_e(ep), \sum_{ep \in d, e \in C_e(ep)} Y_{ep,e} \leq 1 \tag{5}$$

The second type of constraints ensure that each relational phrase should be disambiguated to only one KB relation or one textual relation *at most*:

$$\forall d, \forall kr \in C_{kr}(rp), \sum_{rp \in d, kr \in C_{kr}(rp)} Z_{rp,kr} \leq 1 \tag{6}$$

$$\forall d, \forall tr \in C_{tr}(rp), \sum_{rp \in d, tr \in C_{tr}(rp)} W_{rp,tr} \leq 1 \tag{7}$$

The third constraint ensures the decision variable $Coh_{e,kr}$ equals 1 if and only if both the corresponding variables $Y_{ep,e}$ and $Z_{rp,kr}$ equal 1.

$$\forall d, \forall e \in C_e(ep), \forall kr \in C_{kr}(rp), \forall tr \in C_{tr}(rp) \tag{8}$$

$$Coh_{e,kr} \leq Y_{ep,e} \quad Coh_{e,kr} \leq Z_{rp,kr} \quad Y_{ep,e} + Z_{rp,kr} \leq 1 + Coh_{e,kr} \tag{9}$$

Similarly, we further add the following constraints for $Coh_{e,tr}$ and $Coh_{kr,tr}$:

$$Coh_{e,tr} \leq Y_{ep,e} \quad Coh_{e,tr} \leq W_{rp,tr} \quad Y_{ep,e} + W_{rp,tr} \leq 1 + Coh_{e,tr} \tag{10}$$

$$Coh_{kr,tr} \leq Z_{rp,kr} \quad Coh_{kr,tr} \leq W_{rp,tr} \quad Z_{rp,kr} + W_{rp,tr} \leq 1 + Coh_{kr,tr} \tag{11}$$

We use Gurobi[5] to solve the above ILP problem.

Method	WebQ	QALD-6
Bordes et al. (2014)	39.2	-
Dong et al. (2015)	40.8	-
Yao (2015)	44.3	-
Bast (2015)	49.4	-
Berant (2015)	49.7	-
Reddy et al. (2016)	50.3	-
Yih et al. (2015)	52.5	-
Xu et al. (2016)	53.3	-
This work		
KB	44.1	10.1
KB + Joint	47.1	14.3
Text	40.3	28.7
Text + Joint	45.5	37.4
KB + Text + Joint	**53.8**	**40.9**

Table 1: Results on the test set of QALD-6 and WEBQUESTIONS.

QALD-6
What is the most common language in norway
What currency do they use in switzerland
When olympic games 2012 opening ceremony
What countries does queen elizabeth ii reign
What is the best sandals resort in st lucia
What did the islamic people believe in
WEBQUESTIONS
What is the largest city in the county in which Faulkner spent most of his life
Under which pseudonym did Charles Dickens write some of his books
Where was the Father of Singapore born
Which German mathematicians were members of the von Braun rocket group
Who is the architect of the tallest building in Japan

Table 2: Example questions from WEBQUESTIONS and QALD-6.

5 Experiment

In this section we evaluate our system on two benchmark datasets, QALD-6 and WEBQUESTIONS. After describing the setup, we present our main empirical results and analyze the components of our system.

The QALD-6 task[6] includes a hybrid QA dataset which contains 50 training questions and 25 test questions. We select 15 questions from the training set as the development set and use the remaining 60 ones to evaluate our system.

We also use the WEBQUESTIONS dataset (Berant et al., 2013), which contains 5,810 question-answers pairs. We further split this dataset into the same training and test sets as other baselines, which contain 3,778 questions (65%) and 2,032 questions (35%), to evaluate the system.

As shown in Table 2, these two datasets vary significantly in both syntactic and semantic complexity. For example, 85% questions of WEBQUESTIONS can be directly answered via a single Freebase predicate. However all questions of QALD-6 involve at least one DBpedia predicate and one textual relation, thus can not be accurately answered using DBpedia only.

5.1 Experimental Settings

We have 6 dependency tree patterns based on Bao et al. (2014) to decompose a question into sub-questions. We initialize the word embeddings with Turian et al. (2010)'s word representations with dimensions set to 50. The hyper parameters in our model are tuned using the development set. The window size of MCCNN is set to 3. The sizes of the hidden layer 1 and the hidden layer 2 of the two MCCNN channels are set to 200 and 100, respectively. For each relational phrase, we retain 20 candidate KB predicates and textual relations to the ILP model. The hyper parameters of the ILP objective function (i.e., α, β and δ) are set to 1, 3 and 4, respectively.

[5]http://www.gurobi.com/

[6]http://qald.sebastianwalter.org/index.php?q=6

5.2 Results and Discussion

We use the average question-wise F_1 as our evaluation metric. To give an idea of the impact of different configurations of our method, we consider the following variations with existing methods.

KB. This method involves prediction relying on the KB only in a pipelined fashion. First the entity linking system is run to predict the entity. Then we run the KB-based relation extraction system (described in §3.2) and select the best relation that can cooccur with the entity. We choose this entity-relation pair to predict the answer.

KB + Joint. In addition to selecting local optimal results, we further perform the joint inference over entity and KB predicates.

Text. Instead of applying a KB-based RE method, we map the relation phrase to textual relations as described in §3.3 and find a local optimal solution.

Text + Joint. This method augments the above method with a joint inference step.

KB + Text + Joint. This is our main model. We perform the entity linking, map the relation phrase to KB predicates and textual relations simultaneously, and then infer on the local predications to find a global optimal assignments of the phrases.

Table 1 summarizes the results on the test data along with the results from the literature[7]. We can see that the joint inference gives a performance boost of at least 3% (from 44.1% to 47.1%) regardless of using which type of relation extractor. In addition, text corpora can significantly improve the system performance when using the KB only, and vice versa. The combination of structured KB and free text along with the joint inference outperforms the default model by at least 3.5% (from 37.4% to 40.9%). On the WEBQUESTIONS, our method achieves a new state-of-the-art result beating the previous reported best result of Xu et al. (2016) (with one-tailed t-test significance of $p < 0.05$). And our results on QALD-6 also establishes a new baseline.

5.3 Impact of Textual Relations and KB Predicates

As shown in Table 1, KB-based relation extractor performs better than textual relation extractor on WEBQUESTIONS, but worse on QALD-6. This is due to the fact that WEBQUESTIONS is designed to evaluate the KB-QA systems, therefore the involved relations are guaranteed to be explicitly mapped to KB predicates. In contrast, QALD-6 is proposed to evaluate hybrid-QA systems, and almost no question can be answered using a KB only. Although different datasets have different appetites for the relation extractors, we find the combination of them significantly improves the overall performance.

We also compared our paraphrase model (RAE) with two baselines: EDIT-based and VECTOR-based paraphrase models. Specifically, the former computes the token edit distance between the textual relation tr and relation phrase rp as the similarity score, obtaining 43.6% and 35.4% F_1 on the development set of WEBQUESTIONS and QALD-6, respectively.

The latter obtains the vector representations of tr and rp by summing the word vectors (Turian et al., 2010), and compute the cosine similarity as the similarity score, obtaining 45.7% and 39.3% F_1 on the development set of WEBQUESTIONS and QALD-6, respectively. We find the RAE paraphrase model boosts the performance at least by 6% on QALD-6 and 2% on WEBQUESTIONS.

5.4 Impact of ILP's Constraints

One question of interest is when the ILP model prefers to mapping relational phrases to KB predicates and textual relations simultaneously. We mainly rely on the coherence score between KB predicates and textual relations, i.e., $Coh_{kr,tr}$, to guide the inference model to find a proper assignments. Specifically, if kr and tr have the same argument type expectations, we compute the $Coh_{kr,tr}$ as the probability of mapping tr to kr using the neural network as described in §3.2. Otherwise, $Coh_{kr,tr}$ is set to -1. The

[7]We list several recent results on WEBQUESTIONS. We use development data for all our ablation experiments. Similar trends are observed on both development and test results.

intuition behind is that the selected pair of KB predicates and textual relations should first be coherent, and then semantically similar. If there does not exist such a coherent pair, the model prefers to choosing the one which has higher overall score and neglects the other.

5.5 Error analysis

We analyze the errors of *KB + Text + Joint* model. Around 2% of the errors are caused by incorrect entity linking, and around 5% of the errors are due to incorrect question decomposition. The remaining errors are due to the relation extraction: (i) unbalanced distribution of KB predicates heavily influences the performance of MCCNN model towards frequently seen relations as observed in (Xu et al., 2016); (ii) the RAE model can hardly find proper assignments of textual relations for short-length relational phrases.

5.6 Limitations

While our inference on the structured KB and free text allows the system to answer more open questions to some extent, we still fail at answering some semantically complex questions such as *what is the second longest river in USA* involving aggregation operations. Our current assumption that free text could provide useful textual relations may work only for frequently typed queries or for popular domains like movies, politics and geography. We note these limitations and hope our result will foster further research in this area.

6 Related Work

Over time, the QA task has evolved into two main streams – QA on unstructured data, and QA on structured data. TREC QA evaluations (Voorhees and Tice, 1999) have been explored as a platform for advancing the state of the art in unstructured QA (Wang et al., 2007; Heilman and Smith, 2010; Yao et al., 2013; Yih et al., 2013; Yu et al., 2014; Yang et al., 2015; Hermann et al., 2015). While initial progress on structured QA started with small toy domains like GeoQuery (Zelle and Mooney, 1996), recent trend in QA has shifted to large scale structured KBs like DBPedia, Freebase (Unger et al., 2012; Cai and Yates, 2013; Berant et al., 2013; Kwiatkowski et al., 2013), and on text repository (Banko et al., 2007; Carlson et al., 2010; Krishnamurthy and Mitchell, 2012; Fader et al., 2013; Parikh et al., 2015). An exciting development in structured QA is to exploit multiple KBs (with different schemas) at the same time to answer questions jointly (Yahya et al., 2012; Fader et al., 2014; Zhang et al., 2016).

Our model combines the best of both worlds by inferring over the structured KB and unstructured text. Our work is closely related to Joshi et al. (2014) who aim to answer noisy telegraphic queries using both structured and unstructured data. Their work is limited in answering single relation queries. Our work also has similarities to Sun et al. (2015) who does question answering on unstructured data but enrich it with Freebase.

Joint inference methods over multiple local models has been applied to KB-QA systems (Yahya et al., 2012). In contrast to this prior work concentrating on the structured KB, our constraints are more complex, as we address the joint mapping of relational phrases onto KB predicates and textual relations.

7 Conclusion and Future Work

We have presented a hybrid-QA framework that could infer both on structured KBs and unstructured text to answer natural language questions. Our experiments reveal that integrating structured KB and unstructured text along with a joint inference method improves the overall performance. Our main model achieves the state-of-the-art results on benchmark datasets. A potential application of our method is to improve open domain question answering using the documents retrieved by a search engine.

Since we recognize the query intention inherent in the question using shallow methods, our method is less expressive than the deep meaning representation methods like semantic parsing. Our future work involves developing a shallow semantic parser based on relation extraction in order to better understand the meaning of the questions.

Acknowledgments

We would like to thank Weiwei Sun, Liwei Chen, and the anonymous reviewers for their helpful feedback. This work is supported by National High Technology R&D Program of China (Grant No. 2015AA015403, 2014AA015102), Natural Science Foundation of China (Grant No. 61202233, 61272344, 61370055) and the joint project with IBM Research. For any correspondence, please contact Yansong Feng.

References

Gabor Angeli, Melvin Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Association for Computational Linguistics (ACL)*.

Sren Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. 2007. Dbpedia: A nucleus for a web of open data. In *ISWC/ASWC*.

Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction for the web. In *IJCAI*.

Junwei Bao, Nan Duan, Ming Zhou, and Tiejun Zhao. 2014. Knowledge-based question answering as machine translation. In *ACL*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *EMNLP*.

Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*.

Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. Question answering with subgraph embeddings. In *EMNLP*.

Qingqing Cai and Alexander Yates. 2013. Large-scale semantic parsing via schema matching and lexicon extension. In *ACL*.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka Jr, and Tom M Mitchell. 2010. Toward an architecture for never-ending language learning. In *AAAI*.

Liwei Chen, Yansong Feng, Songfang Huang, Yong Qin, and Dongyan Zhao. 2014. Encoding relation requirements for relation extraction via joint inference. In *ACL*, pages 818–827.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Li Dong, Furu Wei, Ming Zhou, and Ke Xu. 2015. Question answering over freebase with multi-column convolutional neural networks. In *ACL-IJCNLP*.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *EMNLP*.

Anthony Fader, Luke S. Zettlemoyer, and Oren Etzioni. 2013. Paraphrase-driven learning for open question answering. In *ACL*.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2014. Open question answering over curated and extracted knowledge bases. In *SIGKDD*.

Matthew R. Gormley, Mo Yu, and Mark Dredze. 2015. Improved relation extraction with feature-rich compositional embedding models. In *MENLP*, pages 1774–1784, Lisbon, Portugal, September. Association for Computational Linguistics.

Michael Heilman and Noah A Smith. 2010. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *NAACL*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*.

Mandar Joshi, Uma Sawant, and Soumen Chakrabarti. 2014. Knowledge graph and corpus driven segmentation and answer inference for telegraphic entity-seeking queries. In *EMNLP*.

Jayant Krishnamurthy and Tom M Mitchell. 2012. Weakly supervised training of semantic parsers. In *EMNLP-CoNLL*.

Tom Kwiatkowski, Eunsol Choi, Yoav Artzi, and Luke S. Zettlemoyer. 2013. Scaling semantic parsers with on-the-fly ontology matching. In *EMNLP*.

Ndapandula Nakashole, Gerhard Weikum, and Fabian M. Suchanek. 2012. PATTY: A taxonomy of relational patterns with semantic types. In *EMNLP*, pages 1135–1145.

Ankur P. Parikh, Hoifung Poon, and Kristina Toutanova. 2015. Grounded semantic parsing for complex knowledge extraction. In *NAACL*.

Siva Reddy, Oscar Täckström, Michael Collins, Tom Kwiatkowski, Dipanjan Das, Mark Steedman, and Mirella Lapata. 2016. Transforming Dependency Structures to Logical Forms for Semantic Parsing. *Transactions of the Association for Computational Linguistics*, 4.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *NAACL*.

Richard Socher, Eric H. Huang, Jeffrey Pennington, Andrew Y. Ng, and Christopher D. Manning. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *NIPS.*, pages 801–809.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *WWW*.

Huan Sun, Hao Ma, Wen-tau Yih, Chen-Tse Tsai, Jingjing Liu, and Ming-Wei Chang. 2015. Open domain question answering via semantic enrichment. In *WWW*.

Joseph P. Turian, Lev-Arie Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *ACL*, pages 384–394.

Christina Unger, Lorenz Bühmann, Jens Lehmann, Axel-Cyrille Ngonga Ngomo, Daniel Gerber, and Philipp Cimiano. 2012. Template-based question answering over rdf data. In *WWW*.

Ellen M Voorhees and Dawn M. Tice. 1999. The trec-8 question answering track report. In *TREC*.

Mengqiu Wang, Noah A Smith, and Teruko Mitamura. 2007. What is the jeopardy model? a quasi-synchronous grammar for qa. In *EMNLP-CoNLL*.

Kun Xu, Siva Reddy, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2016. Question Answering on Freebase via Relation Extraction and Textual Evidence. In *Proceedings of the Association for Computational Linguistics (ACL 2016)*, Berlin, Germany, August. Association for Computational Linguistics.

Mohamed Yahya, Klaus Berberich, Shady Elbassuoni, Maya Ramanath, Volker Tresp, and Gerhard Weikum. 2012. Natural language questions for the web of data. In *EMNLP*.

Yi Yang and Ming-Wei Chang. 2015. S-mart: Novel tree-based structured learning algorithms applied to tweet entity linking. In *ACL-IJNLP*.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. Wikiqa: A challenge dataset for open-domain question answering. In *EMNLP*.

Xuchen Yao, Benjamin Van Durme, and Peter Clark. 2013. Answer extraction as sequence tagging with tree edit distance. In *NAACL*.

Xuchen Yao. 2015. Lean question answering over freebase from scratch. In *NAACL*.

Wen-tau Yih, Ming-Wei Chang, Christopher Meek, and Andrzej Pastusiak. 2013. Question answering using enhanced lexical semantic models. In *ACL*.

Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *ACL-IJCNLP*.

Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. 2014. Deep learning for answer sentence selection. *arXiv preprint arXiv:1412.1632*.

John M Zelle and Raymond J Mooney. 1996. Learning to parse database queries using inductive logic programming. In *AAAI*.

Yuanzhe Zhang, Shizhu He, Kang Liu, and Jun Zhao. 2016. A joint model for question answering over multiple knowledge bases. In *AAAI*.