Recovering Question Answering Errors via Query Revision

Semih Yavuz and **Izzeddin Gur** and **Yu Su** and **Xifeng Yan**
Department of Computer Science, University of California, Santa Barbara
{syavuz,izzeddingur,ysu,xyan}@cs.ucsb.edu

Abstract

The existing factoid QA systems often lack a post-inspection component that can help models recover from their own mistakes. In this work, we propose to cross-check the corresponding KB relations behind the predicted answers and identify potential inconsistencies. Instead of developing a new model that accepts evidences collected from these relations, we choose to plug them back to the original questions directly and check if the revised question makes sense or not. A bidirectional LSTM is applied to encode revised questions. We develop a scoring mechanism over the revised question encodings to refine the predictions of a base QA system. This approach can improve the F_1 score of STAGG (Yih et al., 2015), one of the leading QA systems, from 52.5% to 53.9% on WEBQUESTIONS data.

1 Introduction

With the recent advances in building large scale knowledge bases (KB) like Freebase (Bollacker et al., 2008), DBpedia (Auer et al., 2007), and YAGO (Suchanek et al., 2007) that contain the world's factual information, KB-based question answering receives attention of research efforts in this area. Traditional semantic parsing is one of the most promising approaches that tackles this problem by mapping questions onto logical forms using logical languages CCG (Kwiatkowski et al., 2013; Reddy et al., 2014; Choi et al., 2015; Reddy et al., 2016), DCS (Berant et al., 2013; Berant and Liang, 2014, 2015), or directly query graphs (Yih et al., 2015) with predicates closely related to KB schema. Recently, neural network based models have been applied to question answering (Bordes et al., 2015; Yih et al., 2015; Xu et al., 2016a,b).

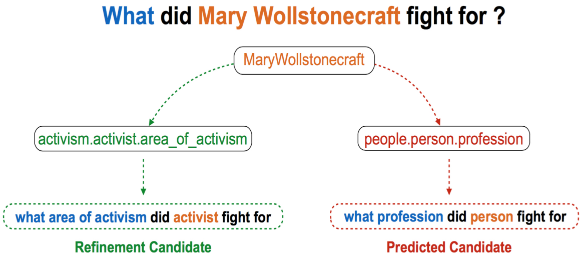

What did Mary Wollstonecraft fight for ?
MaryWollstonecraft
activism.activist.area_of_activism
people.person.profession
what area of activism did activist fight for
what profession did person fight for
Refinement Candidate
Predicted Candidate

Figure 1: Sketch of our approach. Elements in solid round rectangles are KB relation labels. Relation on the left is correct, but the base QA system predicts the one on the right. Dotted rectangles represent **revised questions** with relation labels plugged in. The left revised question looks semantically closer to the original question and itself is more consistent. Hence, it shall be ranked higher than the right one.

While these approaches yielded successful results, they often lack a post-inspection component that can help models recover from their own mistakes. Table 1 shows the potential improvement we can achieve if such a component exists. Can we leverage textual evidences related to the predicted answers to recover from a prediction error? In this work, we show it is possible.

Our strategy is to cross-check the corresponding KB relations behind the predicted answers and identify potential inconsistencies. As an intermediate step, we define **question revision** as a tailored transformation of the original question using textual evidences collected from these relations in a knowledge base, and check if the revised questions make sense or not. Figure 1 illustrates the idea over an example question "*what did Mary Wollstonecraft fight for ?*" Obviously, "*what [area of activism] did [activist] fight for ?*" looks more consistent over "*what [profession] did [person] fight for ?*" We shall build a model that prefers the former one. This model shall be specialized for comparing the revised questions and checking which one makes better sense, not for answering the revised questions. This strategy differentiates

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 912–918
Copenhagen, Denmark, September 7–11, 2017. ©2017 Association for Computational Linguistics

Refinement	F_1	# Refined Qs
STAGG	52.5	-
w/ Best Alternative	58.9	639

Table 1: What if we know the questions on which the system makes mistakes? Best alternative is computed by replacing the predictions of **incorrectly answered questions** by STAGG with its second top-ranked candidate.

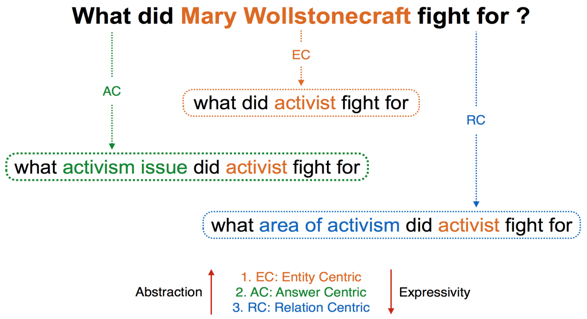
our work from many existing QA studies.

Given a question, we first create its revisions with respect to candidate KB relations. We encode question revisions using a bidirectional LSTM. A scoring mechanism over these encodings is jointly trained with LSTM parameters with the objective that the question revised by a correct KB relation has higher score than that of other candidate KB relations by a certain confidence margin. We evaluate our method using STAGG (Yih et al., 2015) as the base question answering system. Our approach is able to improve the F_1 performance of STAGG (Yih et al., 2015) from 52.5% to 53.9% on a benchmark dataset WEBQUESTIONS (Berant et al., 2013). Certainly, one can develop specialized LSTMs that directly accommodate text evidences without revising questions. We have modified QA-LSTM and ATTENTIVE-LSTM (Tan et al., 2016) accordingly (See Section 4). However, so far the performance is not as good as the question revision approach.

2 Question Revisions

We formalize three kinds of question revisions, namely entity-centric, answer-centric, and relation-centric that revise the question with respect to evidences from topic entity type, answer type, and relation description. As illustrated in Figure 2, we design revisions to capture generalizations at different granularities while preserving the question structure.

Let s_r (e.g., `Activist`) and o_r (e.g., `ActivismIssue`) denote the subject and object types of a KB relation r (e.g., `AreaOfActivism`), respectively. Let α (`type.object.name`) denote a function returning the textual description of a KB element (e.g., relation, entity, or type). Assuming that a candidate answer set is retrieved by executing a KB relation r from a topic entity in question, we can uniquely identify the types of topic entity and answer for the hypothesis by s_r and o_r, respectively. It is also possible that a chain of relations $r = r_1 r_2 \dots r_k$ is used to retrieve an answer set from a topic entity. When $k = 2$, by abuse of notation, we define $s_{r_1 r_2} = s_{r_1}$, $o_{r_1 r_2} = o_{r_2}$, and $\alpha(r_1 r_2) = concat(\alpha(r_1), \alpha(r_2))$.



What did Mary Wollstonecraft fight for ?
EC
AC
RC
what did activist fight for
what activism issue did activist fight for
what area of activism did activist fight for
Abstraction
1. EC: Entity Centric
2. AC: Answer Centric
3. RC: Relation Centric
Expressivity

Figure 2: Illustration of different question revision strategies on the running example w.r.t KB relation `activism.activist.area_of_activism`.

Let $m : (q, r) \mapsto q'$ denote a mapping from a given question $q = [w_1, w_2, \dots, w_L]$ and a KB relation r to revised question q'. We denote the index span of wh-words (e.g., “what”) and topic entity (e.g., “Mary Wollstonecraft”) in question q by $[i_s, i_e]$ and $[j_s, j_e]$, respectively.

Entity-Centric (EC). Entity-centric question revision aims a generalization at the entity level. We construct it by replacing topic entity tokens with its type. For the running example, it becomes “*what did [activist] fight for*”. Formally, $m_{EC}(q, r) = [w_{[1:j_s-1]}; \alpha(s_r); w_{[j_e+1:L]}]$.

Answer-Centric (AC). It is constructed by augmenting the wh-words of entity-centric question revision with the answer type. The running example is revised to “*[what activism issue] did [activist] fight for*”. We formally define it as $m_{AC}(q, r) = [w'_{[1:i_e]}; \alpha(o_r); w'_{[i_e+1:L']}]$, where w'_i's are the tokens of entity-centric question revision $m_{EC}(q, r)$ of length L' with $[i_s, i_e]$ still denoting the index span of wh-words in w'.

Relation-Centric (RC). Here we augment the wh-words with the relation description instead of answer type. This form of question revision has the most expressive power in distinguishing between the KB relations in question context, but it can suffer more from the training data sparsity. For the running example, it maps to “*[what area of activism] did [activist] fight for*”. Formally, it is defined as $m_{RC}(q, r) = [w'_{[1:i_e]}; \alpha(r); w'_{[i_e+1:L']}]$.

3 Model

3.1 Task Formulation

Given a question q, we first run an existing QA system to answer q. Suppose it returns r as the top predicted relation and r' is a candidate relation that is ranked lower. Our objective is to decide if there is a need to replace r with r'. We formulate this task as finding a scoring function $s : (q, r) \rightarrow \mathbb{R}$ and a confidence margin threshold $t \in \mathbb{R}_{>0}$ such that the function

$$replace(r, r', q) = \begin{cases} 1, \text{if } s(q, r') - s(q, r) \geq t \\ 0, \text{otherwise} \end{cases} \quad (1)$$

makes the replacement decision.

3.2 Encoding Question Revisions

Let $q' = (w'_1, w'_2, \ldots, w'_l)$ denote a question revision. We first encode all the words into a d-dimensional vector space using an embedding matrix. Let $\mathbf{e}_i$ denote the embedding of word w'_i. To obtain the contextual embeddings for words, we use bi-directional LSTM

$$\overrightarrow{\mathbf{h}}_i = LSTM_{fwd}(\overrightarrow{\mathbf{h}}_{i-1}, \mathbf{e}_i) \quad (2)$$

$$\overleftarrow{\mathbf{h}}_i = LSTM_{bwd}(\overleftarrow{\mathbf{h}}_{i+1}, \mathbf{e}_i) \quad (3)$$

with $\overrightarrow{\mathbf{h}}_0 = \mathbf{0}$ and $\overleftarrow{\mathbf{h}}_{l+1} = \mathbf{0}$. We combine forward and backward contextual embeddings by $\mathbf{h}_i = \mathbf{concat}(\overrightarrow{\mathbf{h}}_i, \overleftarrow{\mathbf{h}}_i)$. We then generate the final encoding of revised question q' by $\mathbf{enc}(q') = \mathbf{concat}(\mathbf{h}_1, \mathbf{h}_l)$.

3.3 Training Objective

Score Function. Given a question revision mapping m, a question q, and a relation r, our scoring function is defined as $s(q, r) = \boldsymbol{w}^T \mathbf{enc}(m(q, r))$ where $\boldsymbol{w}$ is a model parameter that is jointly learnt with the LSTM parameters.

Loss Function. Let $\mathcal{T} = \{(q, a_q)\}$ denote a set of training questions paired with their true answer set. Let $U(q)$ denote the set of all candidate KB relations for question q. Let $f(q, r)$ denote the F_1 value of an answer set obtained by relation r when compared to a_q. For each candidate relation $r \in U(q)$ with a positive F_1 value, we define

$$N(q, r) = \{r' \in U(q) : f(q, r) > f(q, r')\} \quad (4)$$

as the set of its negative relations for question q. Similar to a hinge-loss in (Bordes et al., 2014), we define the objective function $J(\boldsymbol{\theta}, \boldsymbol{w}, \boldsymbol{E})$ as

$$\sum_{(q,r,r')} max(0, \delta_\lambda(q, r, r') - s(q, r) + s(q, r')) \quad (5)$$

where the sum is taken over all valid $\{(q, r, r')\}$ triplets and the penalty margin is defined as $\delta_\lambda(q, r, r') = \lambda(f(q, r) - f(q, r'))$.

We use this loss function because: i) it allows us to exploit partially correct answers via F_1 scores, and ii) training with it updates the model parameters towards putting a large margin between the scores of correct (r) and incorrect (r') relations, which is naturally aligned with our prediction refinement objective defined in Equation 1.

4 Alternative Solutions

Our approach directly integrates additional textual evidences with the question itself, which can be processed by any sequence oriented model, and benefit from its future updates without significant modification. However, we could also design models taking these textual evidences into specific consideration, without even appealing to question revision. We have explored this option and tried two methods that closely follow QA-LSTM and ATTENTIVE-LSTM (Tan et al., 2016). The latter model achieves the state-of-the-art for passage-level question answer matching. Unlike our approach, they encode questions and evidences for candidate answers in parallel, and measure the semantic similarity between them using *cosine* distance. The effectiveness of these architectures has been shown in other studies (Neculoiu et al., 2016; Hermann et al., 2015; Chen et al., 2016; Mueller and Thyagarajan, 2016) as well.

We adopt these models in our setting as follows: (1) Textual evidences $\alpha(s_r)$ (equiv. of **EC** revision), $\alpha(o_r)$ (equiv. of **AC** revision) or $\alpha(r)$ (equiv. of **RC** revision) of a candidate KB relation r is used in place of a candidate answer a in the original model, (2) We replace the entity mention with a universal #entity# token as in (Yih et al., 2015) because individual entities are rare and uninformative for semantic similarity, (3) We train the score function $sim(q, r)$ using the objective defined in Eq. 5. Further details of the alternative solutions can be found in Appendix A.

5 Experiments

Datasets. For evaluation, we use the WEBQUESTIONS (Berant et al., 2013), a benchmark dataset for QA on Freebase. It contains 5,810 questions whose answers are annotated from Freebase using Amazon Mechanical Turk. We also use SIMPLEQUESTIONS (Bordes et al., 2015), a collection of 108,442 question/Freebase-fact pairs, for training data augmentation in some of our experiments, which is denoted by **+SimpleQ.** in results.

Method	F_1
(Dong et al., 2015)	40.8
(Yao, 2015)	44.3
(Berant and Liang, 2015)	49.7
STAGG (Yih et al., 2015)	**52.5**
(Reddy et al., 2016)	50.3
(Xu et al., 2016b)	53.3
(Xu et al., 2016a)	**53.8**
QUESREV on STAGG	**53.9**
Ensemble	
STAGG-RANK (Yavuz et al., 2016)	54.0
QUESREV on STAGG-RANK	**54.3**

Table 2: Comparison of our question revision approach (QUESREV) on STAGG with variety of recent KB-QA works.

Training Data Preparation. WEBQUESTIONS only provides question-answer pairs along with annotated topic entities. We generate candidates $U(q)$ for each question q by retrieving 1-hop and 2-hop KB relations r from annotated topic entity e in Freebase. For each relation r, we query $(e, r, ?)$ against Freebase and retrieve the candidate answers r_a. Then, we compute $f(q, r)$ by comparing the answer set r_a with the annotated answers.

5.1 Implementation Details

Word embeddings are initialized with pretrained GloVe (Pennington et al., 2014) vectors[1], and updated during the training. We take the dimension of word embeddings and the size of LSTM hidden layer equal and experiment with values in $\{50, 100, 200, 300\}$. We apply dropout regularization on both input and output of LSTM encoder with probability 0.5. We hand tuned penalty margin scalar λ as 1. The model parameters are optimized using Adam (Kingma and Ba, 2015) with batch size of 32. We implemented our models in *tensorflow* (Abadi et al., 2016).

To refine predictions r of a base QA system, we take its second top ranked prediction as the refinement candidate r', and employ $replace(r, r', q)$ in Eq. 1. Confidence margin threshold t is tuned by grid search on the training data after the score function is trained. QUESREV-**AC** + **RC** model is obtained by a linear combination of QUESREV-**AC** and QUESREV-**RC**, which is formally defined in Appendix B. To evaluate the alternative solutions for prediction refinement, we apply the same decision mechanism in Eq. 1 with the trained $sim(q, r)$ in Section 4 as the score function.

We use a dictionary[2] to identify wh-words in a question. We find topic entity spans using Stanford NER tagger (Manning et al., 2014). If there are multiple matches, we use the first matching span for both.

Refinement Model	WebQ.	+ SimpleQ.
QA-LSTM-(equiv **EC**)	51.9	52.5
QA-LSTM-(equiv **AC**)	52.4	52.9
QA-LSTM-(equiv **RC**)	52.6	53.0
ATTENTIVE-LSTM-(equiv **EC**)	52.2	52.6
ATTENTIVE-LSTM-(equiv **AC**)	52.7	53.0
ATTENTIVE-LSTM-(equiv **RC**)	52.9	53.1
QUESREV-**EC**	52.9	52.8
QUESREV-**AC**	**53.5**	53.6
QUESREV-**RC**	53.2	53.8
QUESREV-**AC** + **RC**	53.3	**53.9**

Table 3: $\mathbf{F}_1$ performance of variants of our model QUESREV and alternative solutions on base QA system STAGG.

5.2 Results

Table 2 presents the main result of our prediction refinement model using STAGG's results. Our approach improves the performance of a strong base QA system by 1.4% and achieves 53.9% in F_1 measure, which is slightly better than the state-of-the-art KB-QA system (Xu et al., 2016a). However, it is important to note here that Xu et al. (2016a) uses DBPedia knowledge base in addition to Freebase and the Wikipedia corpus that we do not utilize. Moreover, applying our approach on the STAGG predictions reranked by (Yavuz et al., 2016), referred as STAGG-RANK in Table 2, leads to a further improvement over a strong ensemble baseline. These suggest that our system captures orthogonal signals to the ones exploited in the base QA models. Improvements of QUESREV over both STAGG and STAGG-RANK are statistically significant.

In Table 3, we present variants of our approach. We observe that **AC** model yields to best refinement results when trained only on WEBQUESTIONS data (e.g., **WebQ.** column). This empirical observation is intuitively expected because it has more generalization power than **RC**, which might make **AC** more robust to the training data sparsity. This intuition is further justified by observing that augmenting the training data with SIMPLEQUESTIONS improves the performance of **RC** model most as it has more expressive power.

Although both QA-LSTM and ATTENTIVE-LSTM lead to successful prediction refinements on STAGG, question revision approach consistently outperforms both of the alternative solutions. This suggests that our way of incorporating the new textual evidences by naturally blending them in

[1] http://nlp.stanford.edu/projects/glove/

[2] what, who, where, which, when, how

Example Predictions and Replacements
1. What position did **vince lombardi** play in college ? **STAGG**: person.education / education.institution (2-hop) - what position did person play in college **QUESREV-EC**: football_player.position_s - what position did american football player play in college
2. What did **mary wollstonecraft** fight for ? **STAGG**: person.profession - what profession did person fight for **QUESREV-AC**: activist.area_of_activism - what activism issue did activist fight for
3. Where was **anne boleyn** executed ? **STAGG**: person.place_of_birth - where place of birth was person executed **QUESREV-RC**: deceased_person.place_of_death - where place of death was deceased person executed
4. Where does the **zambezi river** start ? **STAGG**: river.mouth - where mouth does the river start **QUESREV-RC**: river.origin - where origin does the river start

Table 4: Example predictions of STAGG (Yih et al., 2015) and replacements proposed by variants of QUESREV, followed by their corresponding question revisions. The colors *red* and *blue* indicate wrong and correct, respectively. Domain names of KB relations are dropped for brevity.

the question context leads to a better mechanism for checking the consistency of KB relations with the question. It is possible to argue that part of the improvements of refinement models over STAGG in Table 3 may be due to model ensembling. However, the performance gap between QUESREV and the alternative solutions enables us to isolate this effect for query revision approach.

6 Related Work

One of the promising approaches for KB-QA is semantic parsing, which uses logical language CCG (Kwiatkowski et al., 2013; Reddy et al., 2014; Choi et al., 2015) or DCS (Berant et al., 2013) for finding the right grounding of the natural language on knowledge base. Another major line of work (Bordes et al., 2014; Yih et al., 2015; Xu et al., 2016b) exploit vector space embedding approach to directly measure the semantic similarity between questions and candidate answer subgraphs in KB. In this work, we propose a post-inspection step that can help existing KB-QA systems recover from answer prediction errors.

Our work is conceptually related to traditional query expansion, a well-explored technique (Qiu and Frei, 1993; Mitra et al., 1998; Navigli and Velardi, 2003; Riezler et al., 2007; Fang, 2008; Sordoni et al., 2014; Diaz et al., 2016) in information retrieval area. The intuition behind query expansion is to reformulate the original query to improve retrieval performance. Our approach revises questions using candidate answers already retrieved by a base QA system. Revised questions are then used for reasoning about the corresponding predictions themselves, not for retrieving more candidates. Hence, it is specialized rather as a reasoning component than a retrieval one.

Hypothesis generation steps in (Téllez-Valero et al., 2008) and (Trischler et al., 2016) are related to our question revision process. However, hypotheses in these approaches need to be further compared against supporting paragraphs for reasoning. This limits the applicability of them in KB-QA setting due to lack of supporting texts. Our approach modifies the appropriate parts of the question using different KB evidences behind candidate answers that are more informative and generalizable. This enables us to make reasoning about candidate predictions directly via revised questions without relying on any supporting texts.

7 Conclusion

We present a prediction refinement approach for question answering over knowledge bases. We introduce question revision as a tailored augmentation of the question via various textual evidences from KB relations. We exploit revised questions as a way to reexamine the consistency of candidate KB relations with the question itself. We show that our method improves the quality of answers produced by STAGG on the WEBQUESTIONS dataset.

Acknowledgements

The authors would like to thank the anonymous reviewers for their thoughtful comments. This research was sponsored in part by the Army Research Laboratory under cooperative agreements W911NF09-2-0053, NSF IIS 1528175, and NSF CCF 1548848. The views and conclusions contained herein are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notice herein.

A Implementation details of alternative solutions

Following (Tan et al., 2016), we use the same bidirectional LSTM for both questions and textual evidences. For the attentive model, we apply the attention mechanism on the question side because our objective is to match textual evidences to the question context unlike the original model. We use average pooling for both models and compute the *general* attention via a bilinear term that has been shown effective in (Luong et al., 2015).

For the model and training parameters, we follow the strategy described in Section 5.1 with a difference that λ is tuned to be 0.2 in this setting. This intuitively makes sense because the score $sim(q, r)$ is in $[-1, 1]$.

To clarify the question and answer sides for the alternative models, we provide concrete examples in Table 5 for the running example.

Question Side	Answer Side	Model Name
what did #entity# fight for	activist	ALT.-(equiv **EC**)
what did #entity# fight for	activism issue	ALT.-(equiv **AC**)
what did #entity# fight for	area of activism	ALT.-(equiv **RC**)

Table 5: Question (q) and answer (a) sides used for alternative (e.g., ALT.) solutions QA-LSTM and ATTENTIVE-LSTM.

B Combining multiple question revision strategies

We also performed experiments combining multiple question revisions that may potentially capture complementary signals. To this end, let $s_1, \ldots, s_k$ be the trained scoring functions with question revisions constructed by $m_1, \ldots, m_k$, we define $s(q, r) = \sum_{i=1}^{k} \gamma_i s_i(q, r)$ where $\gamma \in \mathbb{R}^k$ is a weight vector that is trained using the same objective defined in Equation 5. This strategy is used to obtain **AC+RC** model reported in experimental results by combining **AC** and **RC** for $k = 2$.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. DBpedia: A nucleus for a web of open data. In *International Semantic Web Conference (ISWC)*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Empirical Methods on Natural Language Processing (EMNLP)*.

Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing. *Annual Meeting of the Association for Computational Linguistics (ACL)* .

Jonathan Berant and Percy Liang. 2015. Imitation learning of agenda-based semantic parsers. *Transactions of the Association for Computational Linguistics (TACL)* .

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *ACM SIGMOD International Conference on Management of Data*.

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

Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale simple question answering with memory networks. *ArXiv* .

Danqi Chen, Jason Bolton, and Christopher D. Manning. 2016. A thorough examination of the cnn/daily mail reading comprehension task. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Eunsol Choi, Tom Kwiatkowski, and Luke Zettlemoyer. 2015. Scalable semantic parsing with partial ontologies. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Fernando Diaz, Bhaskar Mitra, and Nick Craswell. 2016. Query expansion with locally-trained word embeddings. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Li Dong, Furu Wei, Ming Zhou, and Ke Xu. 2015. Question answering over freebase with multicolumn convolutional neural networks. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Hui Fang. 2008. A re-examination of query expansion using lexical resources. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

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 (NIPS)*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Tom Kwiatkowski, Eunsol Choi, Yoav Artzi, and Luke Zettlemoyer. 2013. Scaling semantic parsers with on-the-fly ontology matching. In *Empirical Methods on Natural Language Processing (EMNLP)*.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Empirical Methods on Natural Language Processing (EMNLP)*.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Mandar Mitra, Amit Singhal, and Chris Buckley. 1998. Improving automatic query expansion. In *International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.

Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Roberto Navigli and Paola Velardi. 2003. An analysis of ontology-based query expansion strategies. In *European Conference on Machine Learning (ECML)*.

Paul Neculoiu, Maarten Versteegh, and Mihai Rotaru. 2016. Learning text similarity with siamese recurrent networks. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods on Natural Language Processing (EMNLP)*.

Yonggang Qiu and H.P Frei. 1993. Concept based query expansion. In *International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.

Siva Reddy, Mirella Lapata, and Mark Steedman. 2014. Large scale semantic parsing without question-answer pairs. *Transactions of the Association for Computational Linguistics (TACL)* .

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 (TACL)* .

Stefan Riezler, Alexander Vasserman, Ioannis Tsochantaridis, Vibhu Mittal, and Yi Liu. 2007. Statistical machine translation for query expansion in answer retrieval. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Alessandro Sordoni, Yoshua Bengio, and Jian-Yun Nie. 2014. Learning concept embeddings for query expansion by quantum entropy minimization. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A core of semantic knowledge. In *World Wide Web (WWW)*.

Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2016. Improved representation learning for question answer matching. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Alberto Téllez-Valero, Manuel Montes-y Gómez, Luis Villaseñor-Pineda, and Anselmo Peñas. 2008. Improving question answering by combining multiple systems via answer validation. In *International Conference on Computational Linguistics and Intelligent Text Processing (CICLING)*.

Adam Trischler, Zheng Ye, Xingdi Yuan, and Kaheer Suleman. 2016. Natural language comprehension with epireader. In *Empirical Methods on Natural Language Processing (EMNLP)*.

Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2016a. Hybrid question answering over knowledge base and free text. In *International Conference on Computational Linguistics (COLING)*.

Kun Xu, Siva Reddy, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2016b. Question answering on freebase via relation extraction and textual evidence. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Xuchen Yao. 2015. Lean question answering over freebase from scratch. In *The North American Chapter of the Association for Computational Linguistics (NAACL)*.

Semih Yavuz, Izzeddin Gur, Yu Su, Mudhakar Srivatsa, and Xifeng Yan. 2016. Improving semantic parsing via answer type inference. In *Empirical Methods on Natural Language Processing (EMNLP)*.

Wen-tau Yih, MingWei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.