EviNets: Neural Networks for Combining Evidence Signals for Factoid Question Answering

Denis Savenkov
Emory University
denis.savenkov@emory.edu

Eugene Agichtein
Emory University
eugene.agichtein@emory.edu

Abstract

A critical task for question answering is the final answer selection stage, which has to combine multiple signals available about each answer candidate. This paper proposes *EviNets*: a novel neural network architecture for factoid question answering. *EviNets* scores candidate answer entities by combining the available supporting evidence, *e.g.,* structured knowledge bases and unstructured text documents. *EviNets* represents each piece of evidence with a dense embeddings vector, scores their relevance to the question, and aggregates the support for each candidate to predict their final scores. Each of the components is generic and allows plugging in a variety of models for semantic similarity scoring and information aggregation. We demonstrate the effectiveness of *EviNets* in experiments on the existing TREC QA and WikiMovies benchmarks, and on the new Yahoo! Answers dataset introduced in this paper. *EviNets* can be extended to other information types and could facilitate future work on combining evidence signals for joint reasoning in question answering.

1 Introduction

Most of the recent works in Question Answering (QA) have focused on the problem of semantic matching between a question and candidate answer sentences (He and Lin, 2016; Rao et al., 2016; Yang et al., 2016). The datasets used in these works, such as Answer Sentence Selection Dataset (Wang et al., 2007) and WikiQA (Yang et al., 2015), typically contain a relatively small set of sentences, and the task is to select those that state the answer to the question. However, for many questions, a single sentence does not provide sufficient information, and it may not be reliable in isolation. At the same time, the redundancy of information in large corpora, such as the Web, has been shown useful to improve information retrieval approaches to QA (Clarke et al., 2001).

This work focuses on factoid questions, which can be answered with an entity, *i.e.,* an object in a Knowledge Base (KB) such as Freebase. Knowledge Base Question Answering (KBQA) techniques, such as Berant et al. (2013); Yih et al. (2015); Bast and Haussmann (2015), can be used to answer some of the user questions directly from a KB. However, KBs are inherently incomplete (Dong et al., 2014), and do not have sufficient information to answer many other questions (Fader et al., 2014).

Previous, feature-engineering, approaches for combining different data sources to improve answer retrieval were shown to be quite effective for QA (Sun et al., 2015; Xu et al., 2016; Savenkov and Agichtein, 2016). Alternatively, Memory Networks (Sukhbaatar et al., 2015) and their extensions (Miller et al., 2016) use embeddings to represent relevant data as memories, and summarize them into a single vector, therefore losing information about answers provenances.

In this paper, we introduce *EviNets*, a novel neural network architecture for factoid question answering, which provides a unified framework for aggregating evidence, supporting answer candidates. Given a question, *EviNets* retrieves a set of relevant pieces of information, *e.g.,* sentences from a corpora or knowledge base triples, and extracts mentioned entities as candidate answers. All the evidence signals are then embedded into the same vector space, scored and aggregated using multiple strategies for each answer candidate. Experiments on the TREC QA, WikiMovies and new Yahoo! Answers datasets demonstrate the effectiveness of *EviNets*, and its ability to handle both unstructured text and structured KB triples.

Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 299–304
Vancouver, Canada, July 30 - August 4, 2017. ©2017 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17-2047

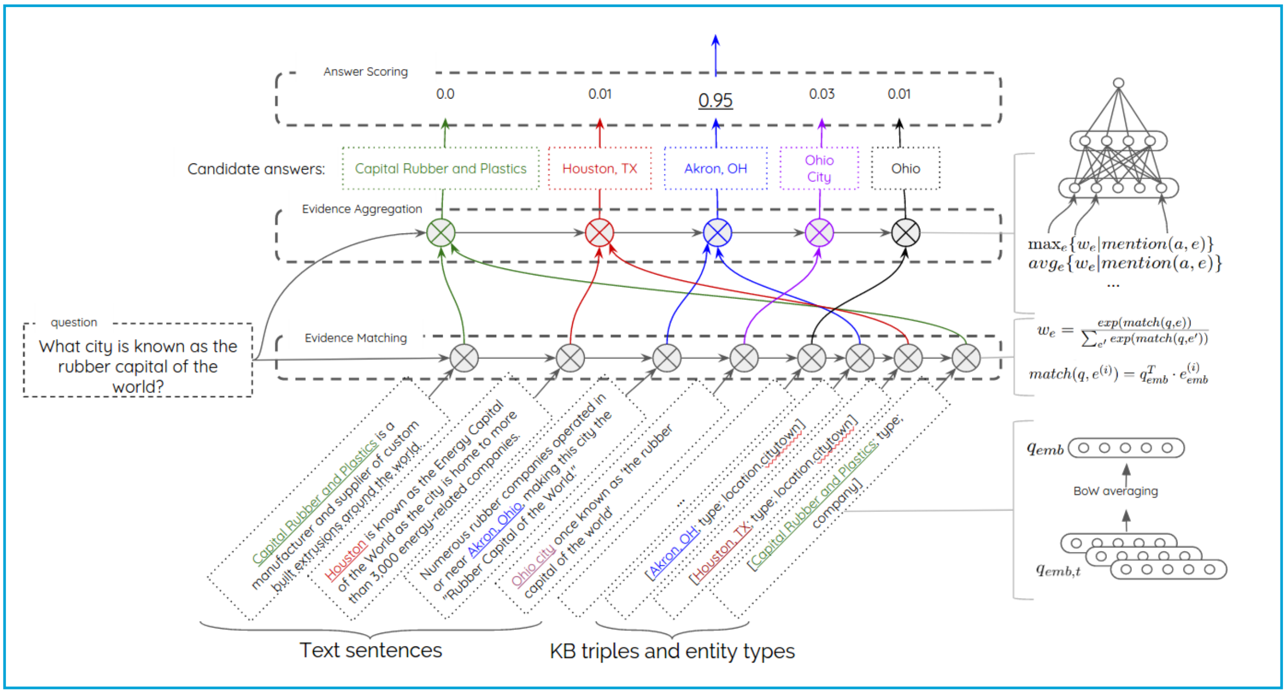
Answer Scoring
0.0
0.01
0.95
0.03
0.01
Candidate answers:
Capital Rubber and Plastics
Houston, TX
Akron, OH
Ohio City
Ohio
Evidence Aggregation
question
What city is known as the rubber capital of the world?
Evidence Matching
Capital Rubber and Plastics is a manufacturer and supplier of custom built extrusions around the world.
Houston is known as the Energy Capital of the World as the city is home to more than 3,000 energy-related companies.
Numerous rubber companies operated in or near Akron, Ohio, making this city the "Rubber Capital of the World."
Ohio city once known as 'the rubber capital of the world'
...
[Akron, OH; type; location.citytown]
[Houston, TX; type; location.citytown]
[Capital Rubber and Plastics; type; company]
Text sentences
KB triples and entity types
$\max_e\{w_e|mention(a,e)\}$
$avg_e\{w_e|mention(a,e)\}$
...
$w_e = \frac{exp(match(q,e))}{\sum_{e'} exp(match(q,e'))}$
$match(q,e^{(i)}) = q_{emb}^T \cdot e_{emb}^{(i)}$
q_{emb}
BoW averaging
$q_{emb,t}$

Figure 1: The EviNets neural network architecture for combining evidence in factoid question answering.

2 *EviNets* Question Answering Model

The high level architecture of *EviNets* is illustrated in Figure 1. For a given question, we extract potentially relevant information, *e.g.,* sentences from documents retrieved from text corpora using a search system. Next, we can use an entity linking system, such as TagMe (Ferragina and Scaiella, 2010), to identify entities mentioned in the extracted information, which become candidate answers. *EviNets* can further incorporate additional supporting evidence, *e.g.,* textual description of candidate answer entities, and potentially useful KB triples, such as types (Sun et al., 2015). Finally, question, answer candidates and supporting evidence are given as input to the *EviNets* neural network.

Let us denote a question by q, and $\{q_t \in R^{|V|}\}$, as a one-hot encoding of its tokens from a fixed vocabulary V. a_i is a candidate answer from the set A, and we will assume, that each answer is represented as a single entity. For each question, we have a fixed set $E = E_{text} \cup E_{KB}$ of evidence statements $e^{(i)}, i = 1..M$, and their tokens $e_t^{(i)}$. A boolean function $mention: A \times E \rightarrow \{0, 1\}$ provides the information about which answer candidates are mentioned in which evidences. Individual tokens $q_t, a_i, e_t^{(i)}$ are translated into the embedding space using a matrix $W_{D\times\,|V|}$, where D is the dimension of the embeddings, *i.e.,* $q_{emb,t} = Wq_t$, $a_{emb,i} = Wa_t$ and $e_{emb,t}^{(i)} = We_t^{(i)}$. In our experiments, we use the same matrix for questions, evidence, and answers. KB entities are considered to be individual tokens, while predicates and type names are tokenized into constituent words.

2.1 Memory Matching Module

Evidence matching is responsible for estimating the relevance of each of the pieces of evidence to the question, *i.e.,* $w_e = softmax(match(q, e))$. The function *match(q, e)* can be implemented using any of the recently proposed semantic similarity estimation architectures[1]. One of the simplest approaches is to average question and each evidence token embeddings and score the similarity using the dot product: $q_{emb} = \frac{1}{L_q}\sum_t q_{emb,t}$ and $e_{emb}^{(i)} = \frac{1}{L_e}\sum_t e_{emb,t}^{(i)}$ and $match(q, e^{(i)}) = q_{emb}^T \cdot e_{emb}^{(i)}$.

2.2 Evidence Aggregation Module

After all the evidence signals have been scored, *EviNets* aggregates the support for each answer candidate. Table 1 summarizes the evidence signals used. With these features, *EviNets* captures different aspects, *i.e.,* how well individual sentences match the question, how frequently the candidate is mentioned and how well a set of answer

[1] https://goo.gl/6gWrgA

Evidence Feature	Description
Maximum evidence score mentioning the answer	$\max_e\{w_e \mid mention(a,e)\}, e \in E, E_{text}\ or\ E_{KB}$
Average evidence score mentioning the answer	$avg_e\{w_e \mid mention(a,e)\}, e \in E, E_{text}\ or\ E_{KB}$
Sum of evidence scores mentioning the answer	$\sum_e\{w_e \mid mention(a,e)\}, e \in E, E_{text}\ or\ E_{KB}$
Number of mentions	$\sum_e\{1 \mid mention(a,e)\}, e \in E_{text}$
Weighted memory similarity to the question	$(\frac{1}{M}\sum_i w_e e_{emb}^{(i)}) \cdot q_{emb}$
Weighted memory similarity to the answer (Sukhbaatar et al., 2015)	$(\frac{1}{M}\sum_i w_e e_{emb}^{(i)}) \cdot a_{emb}$ or $R^T(\frac{1}{M}\sum_i w_e e_{emb}^{(i)} + q_{emb}) \cdot a_{emb}$, where $R_{D\times\ D}$ is a rotation matrix
Weighted memory answer mentions similarity to the answer (Miller et al., 2016)	$(\frac{1}{M}\sum_e w_e[\sum_a\ a_{emb} \mid mention(e,a)]) \cdot a_{emb}$

Table 1: Signals we used to aggregate evidence in support for each of the answer candidates a.

Dataset	Example Questions
TREC QA	Where is the highest point in Japan?
1236 train	What is the coldest place on earth?
202 test	Who was the first U.S. president to appear on TV?
WikiMovies	what films did Ira Sachs write?
96185 train	what films does Claude Akins appear in?
10000 dev	the movie Victim starred who?
9952 test	what type of film is Midnight Run?
Y! Answers	What is Elvis's hairstyle called?
1898 train	Who is this kid in Mars Attacks?
271 dev	who invented denim jeans?
542 test	who's the woman on the progressive.com commercials?

Table 2: Description of TREC QA, WikiMovies and Yahoo! Answers factoid QA datasets.

evidences covers the information requested in the question.

2.3 Answer Scoring Module

Finally, *EviNets* uses the aggregated signals to predict the answer scores, to rank them, and to return the best candidate as the final answer to the question. For this purpose, we use two fully-connected neural network layers with the ReLU activation function, with 32 and 8 hidden units respectively. The model was trained end-to-end by optimizing the cross entropy loss function using the Adam algorithm (Kingma and Ba, 2014).

3 Experimental Evaluation

To test our framework we used TREC QA (Sun et al., 2015), WikiMovies (Miller et al., 2016) benchmarks and the new Yahoo! Answers dataset[2] derived from factoid questions posted on the CQA website (Table 2). In all experiments, embeddings were initialized with 300-dimensional vectors pre-trained with Glove (Pennington et al., 2014). Embeddings for multi-word entity names were obtained by averaging the word vectors of constituent words.

3.1 Baselines

As baselines for different experiments depending on availability and specifics of a dataset we considered the following methods:

- IR-based QA systems: *AskMSR* (Brill et al., 2002) and *AskMSR+* (Tsai et al., 2015), which select the best answer based on the frequency of entity mentions in retrieved text snippets.
- KBQA systems: *SemPre* (Berant et al., 2013) and *Aqqu* (Bast and Haussmann, 2015), which identify possible topic entities of the question, and select the answer from the candidates in the neighborhood of these entities in a KB.
- Hybrid system *QuASE* (Sun et al., 2015) detects mentions of knowledge base entities in text passages, and uses the types and description information from the KB to support answer selection.
- Hybrid system *Text2KB* (Savenkov and Agichtein, 2016), which uses textual resources to improve different stages of the KBQA pipeline.
- Memory Networks: *MemN2N* (Sukhbaatar et al., 2015) and *KV MemN2N* (Miller et al., 2016) represent relevant information with embeddings, and summarize the memories into a single vector using the soft attention mechanism. Additionally, KV MemN2N splits memories into key-value pairs, where keys are used for matching against the question, and values are used to summarize the memories.

[2] available for research purposes at http://ir.mathcs.emory.edu/software-data/

Method	P	R	F1
SemPre	0.157	0.104	0.125
Text2KB	0.287	0.287	0.288
AskMSR+	0.493	0.490	0.491
QuASE (text)	0.550	0.550	0.550
QuASE (text+kb)	0.579	**0.579**	**0.579**
MemN2N	0.333	0.328	0.330
KV MemN2N	0.517	0.500	0.508
EviNets (text)	0.580	0.560	0.569
EviNets (text+kb)	**0.585**	0.564	0.574

Table 3: Precision, Recall and F1 of different methods on TREC QA dataset. Improvements over KV MemN2N are statistically significant.

3.2 TREC QA dataset

The TREC QA dataset is composed of factoid questions, which can be answered with an entity, and were used in TREC 8-12 question answering tracks. Similarly to Sun et al. (2015) we used web search (using the Microsoft Bing Web Search API) to retrieve top 50 documents, parsed them, extracted sentences and ranked them using tf-idf similarity to the question. To compare our results with the existing state-of-the-art, we used the same set of candidate entities as used by the QuASE model. We note that the extracted evidence differs between the models, and we were unable to match some of the candidates to our sentences. For text+kb experiment, just as QuASE, we used entity descriptions and types from Freebase knowledge base. Table 3 summarizes the results. *EviNets* achieves competitive results on the dataset, beating KV MemN2N by 13% in F1 score, and, unlike QuASE, does not rely on expensive feature engineering and does not require any external resources to train.

3.3 WikiMovies dataset

The WikiMovies dataset contains questions in the movies domain along with relevant Wikipedia passages and OMDb knowledge base. Since KVMemN2N already achieves an almost perfect result answering the questions using the KB, we focus on using the provided movie articles from Wikipedia. We followed the preprocessing procedures described in Miller et al. (2016). Unlike TREC QA, where there are often multiple relevant supporting pieces of evidence, answers in the WikiMovies dataset usually have a single relevant sentence, which, however, mentions multiple entities. To help the model distinguish the correct answer, and explore its abilities to encode structured and unstructured data, we generated additional *entity type* triples. For example, if an entity E appears as an object of the predicate `directed_by` in OMDb, we added the `[E, type, director]` triple. As baselines, we used MemN2N and KV MemN2N models, and the results are presented in Table 4. As we can see, with the same setup using individual sentences as evidence/memories *EviNets* significantly outperforms the KV MemN2N model by 27%. It is important to emphasize that the best-reported results of memory networks were obtained using *entity-centered windows* as memories, which requires special pre-processing and increases the number of memories. Additionally, these models used *all* of the KB entities as candidate answers, whereas *EviNets* relies only on the mentioned ones, which is a more scalable scenario for open-domain question answering, where it is not realistic to score millions of candidate answers in real-time.

Method	Accuracy
MemN2N (wiki windows)	0.699*
KV MemN2N (wiki windows)	0.762*
AskMSR (entities)	0.314
KV MemN2N (wiki sentences)	0.524
EviNets (wiki)	0.616
EviNets (wiki + entity types)	**0.667**

Table 4: Accuracy of EviNets and baseline models on the WikiMovies dataset. The results marked * are obtained using a different setup, *i.e.,* they use pre-processed entity window memories, and the whole set of entities as candidates.

3.4 Yahoo! Answers dataset

Yahoo! recently released a dataset with search queries, which lead to clicks on factoid Yahoo! Answers questions, identified as questions with the best answer containing less than 3 words and a Wikipedia page as the specified source of information[3]. This dataset contains 15K queries, which correspond to 4725 unique Yahoo! Answers questions (Table 2). We took these questions, and mapped answers to KB entities using the TagMe entity linking library (Ferragina and Scaiella, 2010). We filtered out questions, for

[3] L27 dataset https://webscope.sandbox.yahoo.com

Method	P	R	F1
Aqqu	0.116	0.117	0.116
Text2KB	0.170	0.170	0.170
AskMSR (entities)	0.175	0.319	0.226
MemN2N	0.072	0.131	0.092
KV MemN2N	0.126	0.228	0.162
EviNets (text)	0.210	0.383	0.271
EviNets (text+kb)	**0.226**	**0.409**	**0.291**
Oracle	0.622	1.0	0.767

Table 5: Precision, Recall and F1 of different methods on Yahoo! Answers factoid QA dataset. The Oracle assumes candidate answers are ranked perfectly and its performance is limited by the initial retrieval step.

which no answer entities with a good confidence[4] were identified, *e.g.,* date answers, and randomly split the rest into training, development and test sets, with 2711 questions in total. Similarly to the TREC QA experiments, we extracted textual evidence using Bing Web Search API, by retrieving top 50 relevant documents, extracting the main content blocks, and splitting them into sentences. We applied the TagMe entity linker to the extracted sentences, and considered all entities of mentions with the confidence score above the 0.2 threshold as candidate answers. For candidate entities we also retrieved relevant KB triples, such as entity types and descriptions, which extended the original pool of evidences.

Table 5 summarizes the results of *EviNets* and some baseline methods on the created Yahoo! Answers dataset. As we can see, knowledge base data is not enough to answer most of these questions, and a state-of-the-art KBQA system Aqqu gets only 0.116 precision. Adding textual data helps significantly, and Text2KB improves the precision to 0.17, which roughly matches the results of the AskMSR system, that ranks candidate entities by their popularity in the retrieved documents. Using text along with KB evidence gave higher performance metrics, boosting F1 from 0.271 to 0.291. EviNets significantly improves over the baseline approaches, beating AskMSR by 28% and KV MemN2N by almost 80% in F1 score.

4 Related Work

The success of deep neural network architectures in computer vision and NLP applications motivated researchers to investigate applying these techniques for answer sentence selection, evaluated on TREC QA (Wang et al., 2007), WikiQA (Yang et al., 2015) and other datasets. A number of models proposed in recent years explore different ways of matching questions and answer sentences (He and Lin, 2016; Yang et al., 2016; Rao et al., 2016). Our *EviNets* architecture allows to easily plug these sentence matching networks into the evidence matching module, and provides the aggregation layer, which helps to make a decision based on all available information.

Our evidence representation module is based on the ideas of memory networks (Sukhbaatar et al., 2015; Kumar et al., 2015; Miller et al., 2016), which also embed relevant information into a vector space. However, they use soft attention mechanism to retrieve the memories, and do not use links from memories to the corresponding answer candidates, which means that all relevant information is squeezed into a fixed dimensional vector. This limitation has been partially addressed in Wang et al. (2016) and Henaff et al. (2016), which accumulate evidence for each answer separately using a recurrent neural network. In contrast, the evidence aggregation in our *EviNets* model uses multiple different features, which is more flexible and can be extended with other signals.

5 Conclusions

We presented *EviNets*, a neural network for question answering, which encodes and aggregates multiple evidence signals to select answers. Experiments on TREC QA, WikiMovies and Yahoo! Answers datasets demonstrate that *EviNets* can be trained end-to-end to use both the available textual and knowledge base information. EviNets improves over the baselines, both in cases when there are many or just a few relevant pieces of evidence, by helping build an aggregate picture and distinguish between candidates, mentioned together in a relevant memory, as is the case for WikiMovies dataset. The results of our experiments also demonstrate that EviNets can incorporate signals from different data sources, *e.g.,* adding KB triples helps to improve the performance over text-only setup. As a limitation of this work and a direction for future research, *EviNets* could be extended to support dynamic evidence retrieval, which would allow retrieving additional answer candidates and evidence as needed.

[4] A minimum ρ score of 0.2 from TagMe was required.

References

Hannah Bast and Elmar Haussmann. 2015. More accurate question answering on freebase. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*.

Eric Brill, Susan Dumais, and Michele Banko. 2002. An analysis of the askmsr question-answering system. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*.

Charles LA Clarke, Gordon V Cormack, and Thomas R Lynam. 2001. Exploiting redundancy in question answering. In *Proceedings of the 24th annual international ACM SIGIR conference*.

Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD*.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2014. Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*.

Paolo Ferragina and Ugo Scaiella. 2010. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM ICKM*.

Hua He and Jimmy Lin. 2016. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *Proceedings of NAACL-HLT*.

Mikael Henaff, Jason Weston, Arthur Szlam, Antoine Bordes, and Yann LeCun. 2016. Tracking the world state with recurrent entity networks. *arXiv preprint arXiv:1612.03969* .

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980. http://arxiv.org/abs/1412.6980.

Ankit Kumar, Ozan Irsoy, Jonathan Su, James Bradbury, Robert English, Brian Pierce, Peter Ondruska, Ishaan Gulrajani, and Richard Socher. 2015. Ask me anything: Dynamic memory networks for natural language processing. *CoRR, abs/1506.07285* .

Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. *arXiv preprint arXiv:1606.03126* .

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. http://www.aclweb.org/anthology/D14-1162.

Jinfeng Rao, Hua He, and Jimmy Lin. 2016. Noise-contrastive estimation for answer selection with deep neural networks. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*.

Denis Savenkov and Eugene Agichtein. 2016. When a knowledge base is not enough: Question answering over knowledge bases with external text data. In *Proceedings of the 39th ACM SIGIR conference*.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*. pages 2440–2448.

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 *Proceedings of the 24th International Conference on World Wide Web*.

C Tsai, Wen-tau Yih, and C Burges. 2015. Web-based question answering: Revisiting askmsr. Technical report, Technical Report MSR-TR-2015-20, Microsoft Research.

Mengqiu Wang, Noah A Smith, and Teruko Mitamura. 2007. What is the jeopardy model? a quasi-synchronous grammar for qa. In *EMNLP-CoNLL*. volume 7, pages 22–32.

Xun Wang, Katsuhito Sudoh, Masaaki Nagata, Tomohide Shibata, Kawahara Daisuke, and Kurohashi Sadao. 2016. Reading comprehension using entity-based memory network. *arXiv preprint arXiv:1612.03551* .

Kun Xu, Siva Reddy, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2016. Question answering on freebase via relation extraction and textual evidence. *arXiv preprint arXiv:1603.00957* .

Liu Yang, Qingyao Ai, Jiafeng Guo, and W Bruce Croft. 2016. anmm: Ranking short answer texts with attention-based neural matching model. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*.

Yi Yang, Scott Wen-tau Yih, and Chris Meek. 2015. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Scott 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 *Proceedings of the 53rd Annual Meeting of the ACL*.