Question Answering Using Structured and Semi-Structured User Generated Content

Doctoral thesis proposal

Denis Savenkov

Dept. of Math & Computer Science

Emory University

denis.savenkov@emory.edu

March, 2016

Abstract

Question answering (QA) research has come a long way from closed domain small systems to IBM Watson, who defeated best human competitors on the Jeopardy! TV show. However, an ultimate human assistant, who can automatically answer all kinds of questions one have is still just a dream. Over the year of research most efforts were put on factoid questions, which can be answered with a short phrase, *e.g.* an entity name, date, number, etc. Modern QA systems employ a variety of different unstructured (text-corpora), semi-structured (tables, Wikipedia infoboxes, question-answer pairs) and structured (databases, knowledge bases) data sources to generate candidate answers. Each of the data sources has its own advantages and limitations, in particular a text fragment encodes very limited amount of information about the entities involved in the statements, which complicates the reasoning about the answer correctness. For example, most factoid QA systems tries to substitute missing information with a prediction, *i.e.* predict an expected lexical answer type (LAT) from the question and match it against the also predicted answer entity type. On the other side of the spectrum knowledge bases (KB) aggregate all available information about entities and support effective querying with a structured query language, such as SPARQL. The problem comes when we need to translate natural language information need to a structured query. Modern knowledge base question answering (KBQA) systems use question-answer pairs (QnA), question paraphrases and other resources to learn a lexicon to map from natural language phrases to knowledge base objects, which is still limited and works well for relatively popular simple questions. In addition knowledge bases are inherently incomplete and many entities, predicates and facts are simply missing. Therefore, it make sense to combine different data sources for question answering, and this approach was already shown to be successful by systems such as IBM Watson, but they treat different data sources mostly independently and use them to produce as a set of candidates, which are then ranked and the best answer is selected. In my dissertation I propose to consider unstructured textual and structured knowledge base resources, connected via entity linking, together for joint reasoning on the candidate generation stage. Existing datasets for question answering are either relatively small (QALD tasks), focused on text (TREC QA) or on knowledge bases only (*e.g.* WebQuestions). To evaluate the approach I'm going to build a new realistic dataset extracted from Yahoo! Answers question-answer pairs.

Beyond factoid questions we have a plethora of different information needs, that require more than a simple fact to answer. Such questions are usually called non-factoid and more and more research effort is devoted to answering such questions. In 2015 Text REtrieval Conference (TREC) pioneered LiveQA shared task track, which targets automatic question answering of various types of questions user post on Yahoo! Answers Community Question Answering (CQA) website. Existing research has demonstrated the effectiveness of reusing answers to similar previously posted questions, but in many cases such questions are not available or challenging to find. Alternatively, existing systems rank passages extracted from regular web pages. However, ranking is complicated due to the lexical gap between question and answer text. Knowledge about what question does a paragraph of text answers would be very useful signal for ranking, which is supported by the results of the winning TREC LiveQA approach. In my thesis I propose to make a step further and automatically extract candidates text passages along with questions which they answers. This can be done by automatically detecting question-answer pairs from certain web pages (*e.g.* forums, FAQ, *etc.*). In addition, we can build upon the recent success with automatic text generation by recurrent neural networks and train a model to predict a question for a given text fragment.

In summary, this dissertation aims to improve the performance of automatic question answering systems for both factoid and non-factoid question answering.

Contents

1 Introduction

1.1 Motivation

The ability to answer user questions with precise and concise information is a hard problem with a long history of research. Various data sources are available for candidate answer generation, two major ones are unstructured text corpora, and structured knowledge bases (e.g. dpPedia [3] and Freebase [8]). A hybrid approach to question answering [5, 16] generates candidates from multiple sources, however each of them is typically processed separately and results are merged on the scoring and ranking stage when some information is already lost. Efficient combination of different information sources has potential to improve both text and knowledge base question answering systems. I propose to combine all the available sources together and do joint reasoning to generate better answer candidates and improve the overall question answering performance.

Question answering from text corpora typically starts by retrieving a set of potentially relevant documents using the question (or some transformation of the question [1]) as the query, and then extracting entities, phrases, sentences or paragraphs believed to be the answer to the question. However, the information available in the retrieved pieces of text is very limited and often not enough to decide whether it can be the answer to the given question. For example, below is one of the questions from TREC QA 2007 dataset:
"What republican senators supported the nomination of Harriet Miers to the Supreme Court?"
A candidate answer sentence *"Minority Leader Harry Reid had already offered his open support for Miers."* mentions a senator "Harry Reid" and clearly says about his support of the nomination. However, "Harry Reid" is not a correct answer to the question because he is a member of the Democratic party. This information is not available in the answer candidate sentence, but it is present as one of the properties in Freebase: [Harry Reid, political_party, Democratic party][1]. Therefore, by looking into the knowledge available about the mentioned entities a QA system can make a better judgment about the candidate answer.

Question answering over linked data (knowledge bases) converts a natural language question into a structured query, such as SPARQL. The main challenge for such systems is to map words and phrases from the question to the corresponding entities and predicates from a KB. Usually, such lexicon is built during training using ground truth question-query pairs [12] or question-answer pairs [6]. Improvements were made by extending the lexicon using Wikipedia and patterns expressing certain predicates obtained via distant supervision [4, 9, 25, 28, 31]. But still, the amount of available labeled or weakly labeled training data is much smaller than the amount of unstructured data. This unstructured data will complement the learned lexicon, e.g. even if a question about a certain predicate wasn't seen during training, a set of text paragraphs mentioning both of the related entities can provide a QA system with enough evidence to make the correct decision.

[1] Actually, in Freebase the entities are connected by a path of length 2 through a mediator node. The predicates on the path are: /government/politician/party and /government/political_party_tenure/party

1.2 Research Questions

1.3 Research Plan

1.3.1 Step 1 (Chapter 3)

1.3.2 Step 2 (Chapter 4)

1.3.3 Step 3 (Chapter 5)

1.3.4 Research Timeline

1.4 Contributions and Implications

The key contributions of the proposed research are: 1. New hybrid KB-text question answering algorithm, that is based on graph search, which includes both KB links as well as text search edges to follow. 2. New labelled dataset for question answering (???) 3. New features for ranking answer candidates ???

2 Related Work

Some useful materials for the related work section:

- https://web.stanford.edu/∼jurafsky/slp3/28.pdf
- PhD thesis "FEATURE-DRIVEN QUESTION ANSWERING WITH NATURAL LANGUAGE ALIGNMENT"

The field of question answering has a long history of research and dates back to 60s (see [19] for a survey of different approaches). The modern era of question answering research started with the rise of the Internet and exponential growth of information available in the World Wide Web. Since 1999 the annual TREC organized a number of open domain question answering shared tasks [13]. Approaches proposed over the years can be largely classified by the type of the information used to find the answers into knowledge base and text-based systems.

2.1 Text-based QA

A traditional approach to factoid question answering over document collections popularized by the TREC QA track is to retrieve a set of potentially relevant documents, extract and rank mentioned entities as candidate answers. One of the main challenges of such an approach is limited amount of information present in the extracted pieces of text. Systems test answer for incorrectness by matching the expected answer type with the type of candidate entity often predicted by an named entity tagger. These systems rely heavily on special complicated ontologies that encode the relationships between different question and answer types, e.g. [18, 20, 24]. Alternatively, the AskMSR system [10] (recently reviewed in [29]) used the redundancy of large text collections such as the web to extract n-grams that occur frequently in a retrieved set of documents. Their counting-based approach performed unexpectedly well on TREC 2001 and sparkled an interest in exploring the web for question answering purposes [21]. However, in many cases the information from the extracted text fragments is not enough to make a judgment on an answer candidate. To solve this problem researchers experimented with using external resources, both unstructured (e.g. Wikipedia articles [2, 11]) and structured (e.g. Wordnet [23]), and demonstrated improved question answering performance. Recently [27] proposed to link entities from candidate answers to Freebase and use its type system and textual entity description for candidate scoring. However, most of the information in a KB is stored as relations between entities, therefore there is a big potential in using all available KB data to improve question answering.

2.2 Knowledge base QA

Recent development of large scale knowledge bases (e.g. dbPedia [3]) and Freebase [8]) motivated research in open domain question answering over linked data. Developed models can be compared on the annual QALD shared task [1] and on a number of available benchmark datasets, e.g. WebQuestions [6]. The main challenge of such systems is to map natural language questions into the

[1] http://greententacle.techfak.uni-bielefeld.de/∼cunger/qald/

structured query representation. Such a lexicon can be learned from a labeled training set [6], ClueWeb collection aligned to Freebase [25, 31], question paraphrases clusters from WikiAnswers [7], Freebase triples rephrased as questions [9], and can be based on the embeddings of questions and knowledge base entities and predicates [9, 28]. However, most of the models are still biased towards the types of questions present in the training set and would benefit from more training data. In this work I propose to extend the training set with question-answer pairs available on CQA websites, which were shown to be useful for relation extraction [26]. In addition, I propose to use unlabeled text resources for candidate query ranking, which can help to generalize to unseen types of questions and questions about predicates never mentioned in the training set.

2.3 Hybrid techniques

Hybrid question answering systems combine multiple available information sources, in particular text document and knowledge bases. Examples of such systems include IBM Watson [16], OpenQA [15], YodaQA [5]. The main difference between such systems and the proposed research is that hybrid systems typically use separate pipelines to extract candidates from different sources and only merge the candidate set while ranking. I propose to extend the representation of each of the data sources for better candidate generation from the beginning.

3 Structured and Unstructured Data for Factoid Question Answering

In this work I propose to enrich the input data representation for QA systems by combining available unstructured, semi-structured and structured data sources for joint reasoning, which can improve the performance of question answering over both text collections and knowledge bases.

3.1 Improving Knowledge Base Question Answering using Unstructured Text Data

3.1.1 Relation extraction for Knowledge Base completion

Relation extraction from Question-Answer pairs

This is my work from NAACL student research workshop.

Question-guided relation extraction

The idea is that we can aggregate related questions and relation extraction patterns. When a person asks a question, we retrieve passages and sentences to extract the answer from. Imaging a question is asking a certain property of an entity. If we can retrieve a sentence, that mentions this entity along with a candidate answer, we can build a pattern for relation extraction. This pattern will be connected to the question "template". Likewise, if we already have relation extraction patterns we can boost those that are retrieve in response to the question and save this connection.

Hypothesis:

1. Patterns retrieved in response to the question are better in quality, we can boost them. We can try to verify this on some relation extraction dataset and questions from some query log. We can also try to use some KBQA dataset.
2. Patterns mined for questions should help question answering. This is essentially weak supervision for training knowledge base question answering using text based question answering.

Problems:

- How to extract new predicates? If we have a question, and a sentence is mentioning a pair of non-related entities, how can we make a new one?
- How to deal with more complex questions, that are not simple relations

Useful dataset: MSN query log, SimpleQuestions from Facebook, WebQuestions, NYT relation extraction dataset.

3.1.2 Question Answering over Knowledge Bases with External Text Data

This is Text2KB SIGIR submission.

3.1.3 Answer ranking approach for Knowledge Base Question Answering

1. Use entity search to retrieve a set of candidate answer entities
2. Predict which entity should be returned as the answer
    - independently for each entity
    - jointly - select a set of entities as the answer. This should be better, can account for the size and homogeneity of the answers

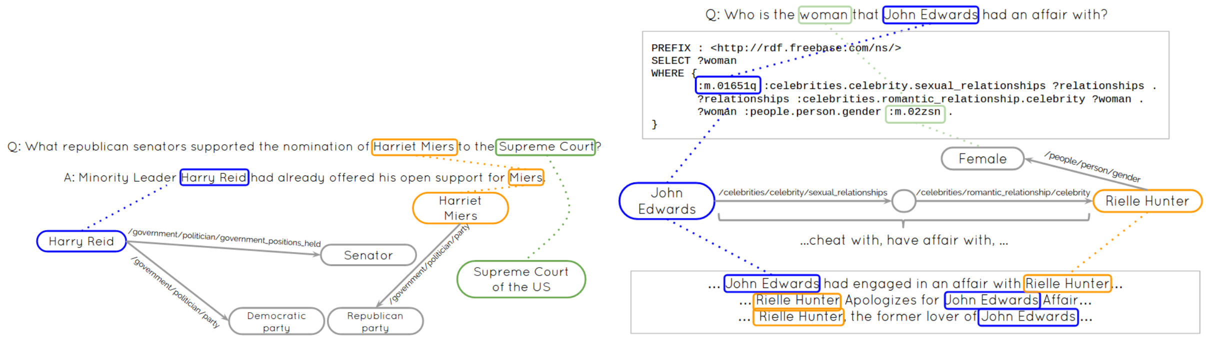
We can index textual data that mention entities to improve entity ranking? Can we just rank Wikipedia pages?

3.2 Semantic Text Annotations for Hybrid Question Answering

In my thesis I propose to annotate text document collection with links to mentioned knowledge base entities. Such semantic annotations open up many opportunities for QA reasoning, because it allows one to go from the information stored in text to structured data and vice versa.

More specifically, I propose the following factoid QA system architecture:

- **Pre-processing**: identify mentions of KB entities in text document collection and index the documents text and mentions in separate fields
- **Topical entity identification**: search the text collection using question (or reformulated question [1]) as a query and use an approach similar to [**?**] to detect question topical entities
- **Candidate generation from text**: extract candidate answer (or intermediate answer) entities with evidence from the retrieved text documents using existing techniques, e.g. [29].
- **Candidate generation from KB**: explore the KB neighborhood of question topical entities and entities extracted from text documents on the previous step
- **Candidate generation from KB & Text**: use entity and text index to find entities mentioned near question topical entity and question terms in the document collection
- **KB evidence extraction**: match neighbourhood of answer entities (entity type and other entities) against the question to get additional evidence
- **Text evidence extraction**: estimate the similarity between the collection text fragments mentionining question and answer entities and the question text
- **Rank candidate**: rank candidate answers using evidence extracted from the KB as well as from text



(a) Annotation of natural language text with mentioned entities and their subgraphs in a knowledge base

(b) Annotation of KB graph nodes and edges with unstructured text data

Figure 3.1: Unstructured text and structured Knowledge Base connected via entity links for question answering

For example, for the question mentioned in the introduction *"What republican senators supported the nomination of Harriet Miers to the Supreme Court?"* and a candidate answer sentence *"Minority Leader Harry Reid had already offered his open support for Miers."*, such joint text-KB representation can look like Figure 3.1a. A QA system can discover that "Harry Reid" political affiliation is with the Democratic Party, and he cannot be referred to as "republican senator". In other cases using a KB as an additional source of information may reveal specific connections between entities in the question and in the answer candidates. For example, for another TREC QA 2007 question *"For which newspaper does Krugman write?"* and retrieved candidate answer *New York Times* a path between "Paul Krugman" and "New York Times" in the knowledge graph gives an evidence in support of the candidate.

Knowledge base question answering (KBQA) produce answers by constructing a structured query, that retrieves answer entities from the KB. The main challenge in KBQA is mapping between natural language phrases in the question and knowledge base entities and predicates. Such systems typically rely on the lexicon learned from the training data [4, 6, 7, 28**?**]. Such lexicons are often limited and needs to be retrained to include additional data. The proposed approach allows a system to dig into the text resources that mention question and candidate answer pairs and use this information for scoring. Figure 3.1b shows a sample of data available for KBQA system to answer the *" Who is the woman that John Edwards had an affair with?"* question from a popular WebQuestions dataset [6].

3.3 Evaluation

3.4 Experiments

3.4.1 Text-based QA

TREC QA datasets served as a benchmark for various question answering systems. Therefore, to evaluate the proposed approach for question answering over text enriched with the structured data I propose to test it on dataset derived from TREC QA and compare against existing strong baselines, including the most related approaches [15, 27]. The proposed system can use the web as the corpus and query it using Bing Search API[1]. Freebase and Reverb extractions [14] are examples of schema-based and open knowledge bases that can be used for the experiments. The metrics used for evaluation typically include accuracy and mean reciprocal rank (MRR).

For non-factoid question answering this year TREC pioneered a new question answering track - TREC LiveQA[2], which targets questions asked by real users of Yahoo! Answers website. This year the deadline for system submission was on August 31 and my system trained on CQA QnA pairs participated in the challenge. The results will be available on the TREC Conference in November 2015. Organizers plan to continue with another TREC LiveQA task next year and this is going to be a good estimation of the effectiveness of the proposed techniques on hard real user questions.

3.4.2 Knowledge base QA

Most of the recent work on knowledge base question answering and semantic parsing have been evaluated on the WebQuestions dataset [6], which contains a collection of question text and correct answer entities. The questions were collected using Google Suggest API and answers crowdsourced using Amazon Mechanical Turk[3] The proposed approach will be compared against the previous results[4] on this dataset. Again, web can be used as a text collection which can be queried using Bing Search API. Relation extraction patterns can be mined using distant supervision from ClueWeb collection using publicly available dataset of Freebase annotations [17].

However, WebQuestions dataset has certain limitations, e.g. questions mined using Google Suggest API have very similar structure and lexicon, and to find the answer to the mined questions users were asked to use the question entity Freebase profile page, which only include entities connected directly with a predicate or through a mediator node. Therefore most of the state-of-the-art results on the dataset use a small number of predefined logical form patterns. On the other hand CQA websites have a fraction of factoid questions with provided text answers. Here I propose to use to construct a new dataset for question answering over Freebase by selecting a subset of QnA pairs with at least one entity in question and answer and some reasonable filtering heuristics and manual validation using crowdsourcing (e.g. through Amazon Mechanical Turk). Existing systems need to be retrained and tested on the new dataset to compare against the proposed model.

Datasets:

- TREC QA
- WebQuestions
- QALD
- New dataset for factoid quetion answering derived from Yahoo! Answers data.

[1]https://datamarket.azure.com/dataset/bing/searchweb
[2]http://trec-liveqa.org/
[3]http://mturk.com/
[4]http://goo.gl/sePBja

3.5 Summary

4 Non-factoid Question Answering

4.1 Utilizing the Structure of Web Pages

4.2 Evaluation

LiveQA

4.3 Summary

5 Discussion and Implication

The proposed approach targets the problem of improving the performance of question answering systems using joint reasoning over unstructured, semi-structured and structured data sources. By linking entity mentions to their knowledge base objects a text-based QA system will be able to use not only lexical information present in extracted text fragments, but also all the factual information about the entities, which should improve its performance. On the other hand, knowledge base question answering should benefit from textual data about predicates and entities mentioned in a questions and a candidate answer. Additional unstructured data will serve as a bridge between a natural language question and the corresponding knowledge base query, which should boost the recall of question answering systems.

However, there are certain questions and limitations, that I would like to discuss. As we know, knowledge bases are inherently incomplete: not only many facts are missing, but also a set of predicates is far from being complete. Therefore, for many questions there are no corresponding predicates in a knowledge base. Given the fact that at the moment text-based QA systems outperform knowledge base systems on factoid questions from the TREC QA dataset, it is unclear how much additional information a KB can add and how big is an advantage over hybrid approaches that simply combine the candidates obtained from various data sources. An alternative approach to get more knowledge about candidate answers is to retrieve more unstructured data, e.g. previous research found Wikipedia articles to be useful. Another question is related to the usefulness of the information stored in a KB for complex and non-factoid questions. The main challenge is to "understand" the text of the answer and predict whether it replies to the question. Facts stored in Freebase or similar KB might not reveal much about the meaning of the answer and we would need a different source of knowledge.

Bibliography

[1] E. Agichtein, S. Lawrence, and L. Gravano. Learning search engine specific query transformations for question answering. In *Proceedings of the Tenth International World Wide Web Conference, WWW 10*, pages 169–178, 2001.

[2] D. Ahn, V. Jijkoun, G. Mishne, K. Müller, M. de Rijke, and K. Schlobach. Using wikipedia at the trec qa track. In *The Thirteenth Text Retrieval Conference (TREC 2004)*, 2005.

[3] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. *Dbpedia: A nucleus for a web of open data*. Springer, 2007.

[4] H. Bast and E. Haussmann. More accurate question answering on freebase. In *CIKM*, 2015.

[5] P. Baudiš and J. Šedivỳ. Modeling of the question answering task in the yodaqa system. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 222–228. Springer, 2015.

[6] J. Berant, A. Chou, R. Frostig, and P. Liang. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, 2013.

[7] J. Berant and P. Liang. Semantic parsing via paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014*, pages 1415–1425, 2014.

[8] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM International Conference on Management of Data*, SIGMOD '08, pages 1247–1250, New York, NY, USA, 2008. ACM.

[9] A. Bordes, S. Chopra, and J. Weston. Question answering with subgraph embeddings. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*, pages 615–620, 2014.

[10] E. Brill, J. Lin, M. Banko, S. Dumais, and A. Ng. Data-intensive question answering. In *Proceedings of TREC 2001*, January 2001.

[11] D. Buscaldi and P. Rosso. Mining knowledge from wikipedia for the question answering task. In *Proceedings of the International Conference on Language Resources and Evaluation*, pages 727–730, 2006.

[12] Q. Cai and A. Yates. Large-scale semantic parsing via schema matching and lexicon extension. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013*, pages 423–433, 2013.

[13] H. T. Dang, D. Kelly, and J. J. Lin. Overview of the trec 2007 question answering track. In *TREC*, volume 7, page 63. Citeseer, 2007.

[14] A. Fader, S. Soderland, and O. Etzioni. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011*, pages 1535–1545, 2011.

[15] A. Fader, L. Zettlemoyer, and O. Etzioni. Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 1156–1165, New York, NY, USA, 2014. ACM.

[16] D. A. Ferrucci, E. W. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. M. Prager, N. Schlaefer, and C. A. Welty. Building watson: An overview of the deepqa project. *AI Magazine*, 31(3):59–79, 2010.

[17] E. Gabrilovich, M. Ringgaard, and A. Subramanya. Facc1: Freebase annotation of clueweb corpora, 2013.

[18] E. H. Hovy, L. Gerber, U. Hermjakob, M. Junk, and C.-Y. Lin. Question answering in webclopedia. In *TREC*, volume 52, pages 53–56, 2000.

[19] O. Kolomiyets and M.-F. Moens. A survey on question answering technology from an information retrieval perspective. *Information Sciences*, 181(24):5412–5434, Dec. 2011.

[20] X. Li and D. Roth. Learning question classifiers. In *19th International Conference on Computational Linguistics, COLING 2002*, 2002.

[21] J. J. Lin and B. Katz. Question answering from the web using knowledge annotation and knowledge mining techniques. In *Proceedings of the 2003 ACM CIKM International Conference on Information and Knowledge Management*, pages 116–123, 2003.

[22] M. Mintz, S. Bills, R. Snow, and D. Jurafsky. Distant supervision for relation extraction without labeled data. In *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, 2009.

[23] M. Pasca and S. Harabagiu. The informative role of wordnet in open-domain question answering. In *Proceedings of NAACL-01 Workshop on WordNet and Other Lexical Resources*, pages 138–143, 2001.

[24] J. Prager, J. Chu-Carroll, E. W. Brown, and K. Czuba. Question answering by predictive annotation. In *Advances in Open Domain Question Answering*, pages 307–347. Springer, 2006.

[25] S. Reddy, M. Lapata, and M. Steedman. Large-scale semantic parsing without question-answer pairs. *TACL*, 2:377–392, 2014.

[26] D. Savenkov, W. Lu, J. Dalton, and E. Agichtein. Relation extraction from community generated question-answer pairs. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 96–102, 2015.

[27] H. Sun, H. Ma, W.-t. Yih, C.-T. Tsai, J. Liu, and M.-W. Chang. Open domain question answering via semantic enrichment. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, pages 1045–1055, Republic and Canton of Geneva, Switzerland, 2015. International World Wide Web Conferences Steering Committee.

[28] W. tau Yih, M.-W. Chang, X. He, and J. Gao. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the Joint Conference of the 53rd Annual Meeting of the ACL and the 7th International Joint Conference on Natural Language Processing of the AFNLP*. ACL Association for Computational Linguistics, July 2015.

[29] C. Tsai, W.-t. Yih, and C. Burges. Web-based question answering: Revisiting askmsr. Technical report, Technical Report MSR-TR-2015-20, Microsoft Research, 2015.

[30] D. Wang and E. Nyberg. A long short-term memory model for answer sentence selection in question answering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015*, pages 707–712, 2015.

[31] X. Yao and B. V. Durme. Information extraction over structured data: Question answering with freebase. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014*, pages 956–966, 2014.