Question Answering using Structured and Semi-structured User Generated Content

Denis Savenkov
Emory University
Atlanta, USA
denis.savenkov@emory.edu

ABSTRACT

The proposed thesis research is focusing on the effective utilization of different unstructured, semi-structured and structured data sources for factoid and non-factoid question answering. Most of the existing factoid QA approaches incorporate text document collections and knowledge bases (KB). However, these data sources are used rather independently for answer candidates generation, which are then merged and ranked together. Text-based approaches find candidates solely in the retrieved text fragments, whereas knowledge base approaches only explore a neighborhood of identified question topic entities. Such separation of data sources is limiting the ability of a QA system to find good candidates in the first place and rank them effectively using all the information available in the data sources. In this thesis I propose to connect text documents and KB via entity linking and use this extended representation to improve both answer generation and ranking. In non-factoid question answering one of the most effective strategies is retrieving answers to similar questions previously posted to community question answering (CQA) websites. Unfortunately, in many cases such questions are not available or challenging to find. Matching between a question and a regular passage from a text document is complicated because of the lexical gap and necessity for deep semantic understanding of answer utterances. In my thesis I propose to tackle these challenges with better web document structure understanding, which can be used to extract question-answer pairs from web forums, FAQ-pages and other documents. The proposed research directions will help to improve the performance of today's QA systems, and can guide future efforts on combining knowledge representations for AI tasks.

Categories and Subject Descriptors

H.3.3 [**H.3.3 Information Search and Retrieval**]: Retrieval models; I.2.3 [**Artificial Intelligence**]: Answer/reason extraction

Keywords

Question Answering; Knowledge Bases; Web Search

1. INTRODUCTION

The ability to answer user questions with precise and concise information is a hard problem with a long history of research. The knowledge necessary to answer these questions is scattered across a variety of resources of various types. Over the years of research in automatic question answering people have studied different unstructured (natural language text), semi-structured (tables, question and answer pairs) and strucutred (knowledge bases) resources for answer generation. However, different types of data sources have their own advantages and limitations. For example, a lot of world knowledge is encoded in raw text format, however, it's hard to reason beyond what is stated in a fragment of text. On the contrary, modern large scale knowledge bases (KB), such as Freebase[1], dbPedia[2], YAGO[3], etc., aggregate all the information about a particular entity and make it quite easy to query using special query languages, such as SPARQL. Unfortunately, regular users would prefer to use natural text for their questions, which leads to a problem of mapping between text phrases and knowledge base entities and predicates. In addition, knowledge bases are inherently incomplete[14] and miss a lot of entities, facts and predicates. In my thesis, I propose to alleviate these disadvantages by combining different data sources together. In particular, by finding mentions of KB entities in text documents we make it possible to get all the information about entities mentioned in a text fragment, as well as get all text fragments mentionining an entity or a pair of entities. 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]`. Therefore, by looking into the knowledge available

[1]http://www.freebase.com/
[2]http://wiki.dbpedia.org/
[3]https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/

about the mentioned entities a QA system can make a better judgment about the candidate answer.

Another big chunk of user questions cannot be answered with an entity, number or date. These questions, usually referred to as non-factoid, include various types of information needs, including defitions, opinions, recommendations, procedures, etc. A series of TREC LiveQA evaluation campaigns, started in 2015, targets non-factoid question answering and asks participants to develop a system to answer real user questions, posted to Yahoo! Answers[4] CQA website, in real-time. It was previously demonstrated [29] that user needs repeat and a strategy of retrieving answer to similar past questions can be quite effective for answering new questions. However, in many cases such questions do not exist or challenging to retrieve. Alternative strategies include ranking text passages extracted from retrieved web documents, which is a challenging problem due to the lexical gap between question and answer texts. Therefore, one would benefit from knowing what kind of questions does a paragraph of text answer. This information can often be inferred from the structured of the web page, e.g. forums, FAQ pages, and titles and subtitles of web pages. In my thesis I propose to make a better use of the structure of web documents where possible, e.g. by detecting the type of a web page and extracting its key components. For other generic web documents and their passages I propose to predict how likely a passage can answer the given question using the advances in deep learning method for text processing, including sequence to sequence models [31].

2. RESEARCH PROPOSAL

In this section I describe the proposed research for both factoid and non-factoid question answering in more details.

2.1 Factoid 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 [12] 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. [33].
- **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

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 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.

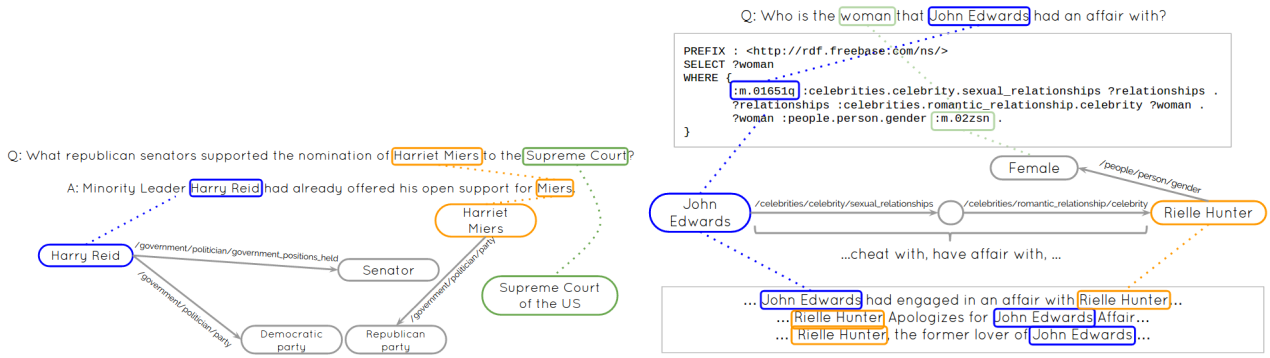
Knowledge base question answering (KBQA) produce answers by constructing a structured query, that retrieves answer entities from the KB. The main challenge in KBQA is mapping between natural language phrases in the question and knowledge base entities and predicates. Such systems typically rely on the lexicon learned from the training data [4, 6, 7, 32, 34]. 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 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].

2.2 Non-factoid Question Answering

Non-factoid questions are typically answered with a relatively long paragraph of text[5]. This fact and the nature of questions limits the utility of structured KB resources. One of the main challenges for non-factoid question answering is matching between the question needs and the information expressed in text fragment. Analysis of TREC LiveQA 2015 participants [27] revealed that the quality of answers extracted from previously posted similar questions is typically higher than from regular web passages. Therefore, non-factoid QA system would benefit from the information on which questions does a paragraph of text answer. This information can often be extracted from the structure of a web document, e.g. forum threads, FAQ pages or various CQA websites. Alternatively, we can train a model to predict whether a paragraph answers a given question using titles, subtitles and surrounding text of a web page.

[4]http://answers.yahoo.com/

[5]TREC LiveQA'15 challenge limits the answer to 1000 characters



(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 1: Unstructured text and structured Knowledge Base connected via entity links for question answering

My proposal for non-factoid question answering can be summarized as follows:

- **CQA candidate generation**: retrieve a set of question-answer pairs by search a CQA archive[6]
- **Web document retrieval**: retrieve a set of documents by querying web search with the question (and queries generated from it)
- **Web candidate answer generation**: classify web page into one of the following types: article, forum thread, FAQ page, CQA page, other. Extract key elements using type-specific extractors (QnA pairs FAQ and CQA pages, forum question and posts and article passages with the corresponding titles, subtitles and surrounding text).
- **Ranking**: Rank the generated candidate answers

==========

More specifically, to do this kind of inference I propose:

- use existing approaches for document retrieval (e.g. web search using question as a query [33]) and candidate answer extraction.
- perform entity linking to mentions of KB entities in questions and corresponding candidate answers.
- for each mentioned entity extract a subgraph containing its neighborhood up to certain number of edges away and paths to other mentioned entities.
- follow machine learning approach for candidate answer ranking and extend the feature representation with features derived from subgraph analysis. Examples of features are:
  - features describing discovered connections between entities mentioned in a question and a candidate answer, such as indicators of the relations, combination of relations with words and n-grams from the questions, similarity between the relations and the question text (using tf-idf or embeddings representation), etc. Textual representations of the predicates in structured knowledge bases can be obtained either from its description or using patterns learned from a large collection using distant supervision [23].
  - features describing the entities mentioned in the answer, i.e. similarities between entity properties and question words, n-grams and phrases, etc.

[6]https://answers.yahoo.com/

3. EXPERIMENTS

3.1 Factoid 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 [16, 30]. The proposed system can use the web as the corpus and query it using Bing Search API[7]. Freebase and Reverb extractions [15] 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).

On the other hand, 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[8] The proposed approach will be compared against the previous results[9] on this dataset. Again, web can be used as a text collection which can be queried using Bing Search API. To allow entity-based search of text-documents we can use ClueWeb12 collection and the corresponding entity mentions annotations [18].

However, both of these benchmarks have certain limitations. TREC QA dataset is relatively small, and WebQuestions dataset is biased towards questions that can be relatively easily answered from Freebase (because answers were labelled using Freebase entity profile pages). In my disser-

[7]https://datamarket.azure.com/dataset/bing/searchweb
[8]http://mturk.com/
[9]http://goo.gl/sePBja

tation I propose to build a new dataset for factoid question answering, which is going to be based on questions posted to a CQA website. Using simple heuristics it's possible to pre-filter a set of questions that can be further post-filtered and labelled using crowdsourcing. More specifically, I propose to use Yahoo! Answers WebScope dataset of question-answer pairs, filter them to remove non-factoid and opinion questions using some simple heuristics, such as: keep single sentence questions only, that start with the question word (except why), do not contain personal pronouns, comparative and superlative adjectives. Preliminary analysis revealed, that after the pre-filtering of 4.4M questions from the WebScope dataset, we have 70K questions left, from which ∼30% are factoid. I will further filter this dataset using Mechanical Turk workers, who will also annotate the correct answers by selecting one or more entities, mentioned in the answer text, provided on Yahoo! Answers. This dataset will represent a sample of real user information needs, will be more general than WebQuestions dataset and larger than TREC QA. In addition, since KB entities will be used to annotate answers, there will be no need in designing answer checking regular expressions, etc. I'm going to make this dataset public and compare existing text-based and knowledge base question answering system along with the system I develop in my thesis.

3.2 Non-factoid QA

To evaluate the performance of the developed non-factoid QA system I'm going to participate in TREC LiveQA 2016 shared task[10]. I participated in the task in 2015 [27], and the second attempt will allow me not only compare the developed system against other teams, but also check the progress of my system against the last year baseline.

4. RELATED WORK

The field of question answering has a long history of research and dates back to 60s (see [20] 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.

4.1 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. [19, 21, 25]. Alternatively, the AskMSR system [10] (recently reviewed in [33]) 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 [22]. 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 [24]), and demonstrated improved question answering performance. Recently [30] 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.

4.2 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 [11] 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 structured query representation. Such a lexicon can be learned from a labeled training set [6], ClueWeb collection aligned to Freebase [26, 35], 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, 32]. 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 [28]. 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.

4.3 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 [17], OpenQA [16], 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.

5. DISCUSSION

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,

[10] https://sites.google.com/site/trecliveqa2016/call-for-participation

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

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.

6. REFERENCES

[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] M. Cornolti, P. Ferragina, M. Ciaramita, H. Schütze, and S. Rüd. The smaph system for query entity recognition and disambiguation. In *Proceedings of the first international workshop on Entity recognition & disambiguation*, pages 25–30. ACM, 2014.

[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] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 601–610, New York, NY, USA, 2014. ACM.

[15] 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.

[16] 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.

[17] 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.

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

[19] 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.

[20] 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.

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

[22] 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.

[23] 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.

[24] 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.

[25] 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.

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

[27] D. Savenkov. Ranking answers and web passages for non-factoid question answering: Emory ir lab @ trec 2015 liveqa. In *TREC*, 2015.

[28] 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.

[29] A. Shtok, G. Dror, Y. Maarek, and I. Szpektor. Learning from the past: answering new questions with past answers. In *Proceedings of the 21st international conference on World Wide Web*, pages 759–768. ACM, 2012.

[30] 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.

[31] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[32] 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 âĂŞ Association for Computational Linguistics, July 2015.

[33] 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.

[34] X. Yao. Lean question answering over freebase from scratch. In *Proceedings of NAACL Demo*, 2015.

[35] 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.

APPENDIX

A. STUDENT STATEMENT

I'm a 5th year PhD student at Emory University and during the course of my studies I've worked on a number of different topics, starting from click log analysis, study of mobile users behavior, search user assistance for difficult search tasks, information extraction and now question answering. In addition I've spent 4 wonderful summers in Microsoft and Google working on search results diversity, entity linking, information extraction and coreference resolution. From a wide range of topics I've worked on, I've decided to focus my thesis on question answering. In several month I'm going to do a thesis proposal and WSDM Doctoral Consortium presents a great chance to discuss the proposed direction with experienced researchers with various background. I hope to make a valuable contribution to the area and discussion with the experts will help me polish the plan for the future thesis. The proposed topic involves several related areas, such as information retrieval, natural language processing, web search and mining, knowledge representation, knowledge base construction and semantic parsing. Most of my work was in the information retrieval area, and I have less experience with knowledge representation and natural language processing. Therefore, I would greatly appreciate a chance to attend the consortium and discuss the potential challenges and ways to improve the work with researches with complementary experiences, which can make my work more interesting for different communities.

B. ADVISOR STATEMENT

My name is Eugene Agichtein[12], and I am Denis Savenkov's Ph.D. advisor. Denis is entering his 5th year as a PhD student at the Emory Mathematics & Computer Science department. He is expected to present his thesis proposal in the next few months, and to defend his thesis within a year, likely by December 2016. Denis has broad interests within Information Retrieval, and has done excellent work on a number of topics in IR. In particular, he and his team placed in the top 3 teams at the Yandex Click prediction challenge, and subsequently won the "Search Engine Prediction" challenge, publishing a full SIGIR paper on the novel techniques they developed. Denis has also done in-depth studies of community question answering data, resulting in a full ICWSM paper as well as interesting analysis of how (not) to support searchers during complex tasks. The analysis of difficult, or failed, searches, led Denis to become interested in question answering, as a way of attacking some of the long standing challenges in search. I believe Denis has the skills and the creativity to tackle the ambitious task he proposes, on combining information from unstructured text and knowledge bases for question answering. Attending the doctoral consortium at WSDM will give Denis a chance to interact with leading experts in Web search, social media analysis and data mining, and especially those with complementary expertise to mine. As an IR researcher, I am reasonably confident on advising Denis on the aspects of his thesis related to IR and QA over unstructured (text) data, but am less familiar with the work done in KD and linked data communities, both in academia or industry. As Denis' thesis spans both areas, input from researchers who work on constructing and utilizing KBs, knowledge graphs and other structured sources is critical. Participating in DC would be particularly beneficial to Denis this year, as he is at the stage of developing his thesis proposal.

[12]http://www.mathcs.emory.edu/∼eugene/