Question Answering Using Structured and Unstructured Data

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**

1.1 Motivation . . . 1

1.1.1 Factoid vs Non-factoid Questions . . . 3

1.2 Research Questions . . . 3

1.3 Research Plan . . . 3

1.3.1 Step 1 (Chapter 3) . . . 3

1.3.2 Step 2 (Chapter 4) . . . 3

1.3.3 Step 3 (Chapter 5) . . . 3

1.3.4 Research Timeline . . . 3

1.4 Contributions and Implications . . . 3

2 Related Work **4**

2.1 Factoid question answering . . . 4

2.1.1 Text-based question answering . . . 4

2.1.2 Knowledge base question answering . . . 6

2.1.3 Hybrid question answering . . . 7

2.2 Non-factoid question answering . . . 9

2.3 User interactions . . . 10

2.4 Summary of Related Work . . . 10

3 Structured and Unstructured Data for Factoid Question Answering **11**

3.1 Relation Extraction for Knowledge Base Completion . . . 11

3.1.1 Relation extraction from Question-Answer pairs . . . 11

3.2 Semantic Text Annotations for Hybrid Question Answering . . . 11

3.2.1 Approach . . . 12

3.2.2 Evaluation . . . 13

3.3 Summary . . . 14

4 Non-factoid Question Answering **15**

4.1 The architecture of the system . . . 15

4.1.1 Question Analysis . . . 15

4.1.2 Structure of the Web Page for Candidate Generation and Scoring . . . 16

4.1.3 Answer Summarization . . . 16

4.2 Evaluation . . . 17

4.3 Summary . . . 17

5 Users and Question Answering Systems **19**
5.1 Search Hints for Complex Informational Tasks 19
5.2 Using the Wisdom of a Crowd for Question Answering 19
5.3 Clarification Questions 19

6 Discussion and Implication **20**

Bibliography **21**

1 Introduction

1.1 Motivation

THIS IS AN OLDER VERSION. WILL REVISE AFTER I'M DONE WITH MAIN SECTIONS.

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 [6] and Freebase [18]). A hybrid approach to question answering [11, 44] 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 [29] or question-answer pairs [12]. Improvements were made by extending the lexicon using Wikipedia and patterns expressing certain predicates obtained via distant supervision [9, 19, 82, 99, 122]. 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

Table 1.1: Pros and cons of structured and unstructured data sources for factoid and non-factoid question answering

	unstructured data	structured data
factoid questions	Text + easy to match against question text + cover a variety of different information types - each text phrase encodes a limited amount of information about mentioned entities	Knowledge Bases + aggregate all the information about entities allow complex queries over this data using special languages (e.g. SPARQL) - hard to translate natural language questions into special query languages - KBs are incomplete (missing entities, facts and properties)
non-factoid questions	Text + contain relevant information to a big chunk of user needs - hard to extract semantic meaning of a paragraph to match against the question (lexical gap)	Question-Answer pairs + easy to find a relevant answer by matching the corresponding questions - cover a smaller subset of user information needs

Table 1.1 lists pros and cons of structured and unstructured data sources for factoid and non-factoid question answering.

The main focus of research in automatic question answering was on **factoid questions**, which inquire about a certain fact and can be answered with a short phrase, such as an entity name, date or number. Such questions cover, although big and an important, but only a fraction of user information needs. Questions that don't fall into this category are usually referred to as **non-factoid**.

—-

Over the decades of research in factoid question answering, two relatively separate approaches have emerged: text-centric, or TextQA and knowledge base-centric, or KBQA. Each approach has its own advantages and disadvantages (Table 1.1). Billions of documents on the web contain all kinds of knowledge about the world, which can be retrieved to answer user questions. However, each individual statement includes a very limited amount of information about mentioned entities. On the other side, modern open domain large scale knowledge bases, such as dbPedia[2], YAGO[73], Freebase[3], WikiData[4], etc., contain millions of entities and facts about them, and are quite effective in answering some of the user questions. However, knowledge bases have their own disadvantages:

- knowledge bases are inherently incomplete [36], even the largest existing resources miss a lot of entities, facts and properties, that might be of interest to some users.

- it's quite challenging to translate a natural language question into a structured language,

[2]http://wiki.dbpedia.org/
[3]http://www.freebase.com
[4]https://www.wikidata.org/

such as SPARQL, to query a knowledge base [12].

One way to improve the situation with knowledge base incompleteness is to extract missing information from other data sources, *e.g.* [25, 27, 36, 38, 50, 64]. In my thesis I focus on one particular data source, that didn't receive enough attention in the relation extraction literature, namely question-answer pairs. Section 3.1 will describe our experiments and results in utilizing this data to improve knowledge base coverage. Unfortunately, relation extraction isn't perfect either and there are both precision and recall losses. Alternatively, in my thesis I propose a new hybrid approach to question answering, which leverages a combination of text and knowledge base data to improve every stage of question answering process (Section 3.2).

1.1.1 Factoid vs Non-factoid Questions

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

The field of automatic questions answering has a long history of research and dates back to the days when the first computers appear. By the early 60s people have already explored multiple different approaches to question answering and a number of text-based and knowledge base QA systems existed at that time [90, 91]. In 70s and 80s the development of restricted domain knowledge bases and computational linguistics theories facilitated the development of interactive expert and text comprehension systems [5, 88, 111, 110]. The modern era of question answering research was motivated by a series of Text Retrieval Conference (TREC[1]) question answering shared tasks, which was organized annually since 1999 [102]. A comprehensive survey of the approaches from TREC QA 2007 can be found in [33]. An interested reader can refer to a number of surveys to track the progress made in automatic question answering over the years [52, 4, 105, 63, 80, 3, 49].

The main focus of research in automatic question answering was on factoid questions. However, recently we can observe an increased interest in non-factoid question answering, and as an indicator in 2015 TREC started a LiveQA shared task track[2], in which the participant systems had to answer various questions coming from real users of Yahoo! Answers[3] in real time.

In the rest of the chapter I will describe related work in factoid (Section 2.1) and non-factoid (Section 2.2) question answering with the focus on data sources used.

2.1 Factoid question answering

Since the early days of automatic question answering researches explored different sources of data, which lead to the development of two major approaches to factoid question answering: text-based (TextQA) and knowledge base question answering (KBQA) [90]. We will first describe related work in TextQA (Section 2.1.1), then introduce KBQA (Section 2.1.2) and in Section 2.1.3 present existing techniques for combining different information sources together.

2.1.1 Text-based question answering

A traditional approach to factoid question answering over text document collections, popularized by TREC QA task, starts by querying a collection with possibly transformed question and retrieving a set of potentially relevant documents, which are then used to identify the answer. Information retrieval for question answering has certain differences from traditional IR methods [62], which are usually based on keyword matches. A natural language question contains certain information, that is not expected to be present in the answer (*e.g.* the keyword who, what, when, *etc.*), and the answer statement might use language that is different from the question (lexical gap problem). On the other side, there is a certain additional information about expected answer statement, that a QA system might infer from the question (*e.g.* we expect to see in a number in response to the "how many" question). One way to deal with this problem is to transform the question in certain

[1] http://trec.nist.gov
[2] http://trec-liveqa.org/
[3] http://answers.yahoo.com/

ways before querying a collection [1, 23]. Raw text data might be extended with certain semantic annotations by applying part of speech tagger, semantic role labeling, named entity recognizer, *etc.* . By indexing these annotations a question answering system gets an opportunity to query collection with additional attributes, inferred from the question [17, 124].

The next stage in TextQA is to select sentences, that might contain the answer. One of the mostly used benchmark datasets for the task, proposed in [107], is based on TREC QA questions and sentences retrieved by participating systems[4]. The early approaches for the task used simple keyword match strategies [57, 94]. However, in many cases keywords doesn't capture the similarity in meaning of the sentences very well and researches started looking on syntactic information. Syntactic and dependency tree edit distances and kernels allow to measure the similarity between the structures of the sentences [81, 87, 51, 123, 106]. Recent improvements on the answer sentence selection task come are associated with the deep learning techniques, *e.g.* recursive neural networks using sentence dependency tree [58], convolutional neural networks [127, 84], recurrent neural networks [98, 104]. Another dataset, called WikiQA [119], raises a problem of answer triggering, *i.e.* detecting cases when the retrieved set of sentences don't contain the answer.

To provide a user with the concise answer to his factoid question QA systems extract the actual answer phrase from retrieved sentences. This problem is often formulated as a sequence labeling problem, which can be solved using structured prediction models, such as CRF [123], or as a node labeling problem in an answer sentence parse tree [74].

Unfortunately, passages include very limited amount of information about the candidate answer entities, *i.e.* very often it doesn't include the information about their types (person, location, organization, or more fine-grained CEO, president, basketball player, *etc.*), which is very important to answer question correctly, *e.g.* for the question "*what country will host the 2016 summer olympics?*" we need to know that `Rio de Janeiro` is a city and `Brazil` is the country and the correct answer to the question. Therefore, a lot of effort has been put into developing answer type typologies [55, 54] and predicting and matching expected and candidate answer types from the available data [67, 68, 79]. Many approaches exploited external data for this task, I will describe some of this efforts in Section 2.1.3.

Very large text collections, such as the Web, contain many documents expressing the same information, which makes it possible to use a simpler techniques and rely on redundancy of the information. `AskMSR` QA system was one of the first to exploit this idea, and achieved very impressive results on TREC QA 2001 shared task [22]. The system starts by transforming a question into search queries, extracts snippets of search results from a web search engine, and consider word n-grams as answer candidates, ranking them by frequency. A recent revision of the AskMSR QA system [100] introduced several improvements to the original system, *i.e.* named entity tagger for candidate extraction, and additional semantic similarity features for answer ranking. It was also observed, that modern search engines are much better in returning the relevant documents for question queries and query generation step is no longer needed. Another notable systems, that used the web as the source for question answering are `MULDER`[65], `Aranea` [70], and a detailed analysis of what affects the performance of the redundancy-based question answering systems can be found in [69].

---

[4] A table with all known benchmark results and links to the corresponding papers can be found on http://aclweb.org/aclwiki/index.php?title=Question_Answering_(State_of_the_art)

2.1.2 Knowledge base question answering

Earlier in the days knowledge bases were relatively small and contained information specific to a particular domain, *e.g.* baseball statistics [48], lunar geology [111], geography [128]. However, one of the main challenges in KBQA is mapping between natural language phrases to the database concepts, which raises a problem of domain adaption of question answering systems.

Recent development of large scale knowledge bases (*e.g.* dbPedia [6], Freebase [18], YAGO [95], WikiData[5] shifted the attention towards open domain question answering. Knowledge base question answering approaches can be evaluated on an annual Question Answering over Linked Data (QALD[6]) shared task, and some popular benchmark dataset, such as Free917 [30] and WebQuestions [12]. A survey of some of the proposed approaches can be found in [101].

A series of QALD evaluation campaigns has started in 2011, and since then a number of different subtasks have been offered, *i.e.* since QALD-3 includes a multilingual task, and QALD-4 formulated a problem of hybrid question answering. These tasks usually use dbPedia knowledge base and provide a training set of questions, annotated with the ground truth SPARQL queries. The hybrid track is of particular interest to the topic of this dissertation, as the main goal in this task is to use both structured RDF triples and free form text available in dbPedia abstracts to answer user questions.

The problem of lexical gap and lexicon construction for mapping natural language phrases to knowledge base concepts is one of the major difficulties in KBQA. The earlier systems were mainly trained from question annotated with the correct parse logical form, which is expensive to obtain. Such an approach is hard to scale to large open domain knowledge bases, which contain millions of entities and thousands of different predicates. An idea to extend a trained parser with additional lexicon, built from the Web and other resources, has been proposed by [29]. However, most of the parses of the question produce different results, which means that it is possible to use question-answer pairs directly [12]. PARALEX system ([41]) construct a lexicon from a collection of question paraphrases from WikiAnswers[7]. A somewhat backwards approach was proposed in ParaSempre model of [13], which ranks candidate structured queries by first constructing a canonical utterance for each query and then using a paraphrasing model to score it against the original question. Another approach to learn term-predicate mapping is to use patterns obtained using distant supervision [76] labeling of a large text corpus, such as ClueWeb [121]. Such labelled collections can also be used to train a KBQA system, as demonstrated by [82]. Such an approach is very attractive as it doesn't require any manual labeling and can be easily transfered to a new domain. However, learning from statements instead of question answer pairs has certain disadvantages, *e.g.* question-answer lexical gap and noise in distant supervision labeling. Modern knowledge bases also contain certain surface forms for their predicates and entities, which makes it possible to convert KB RDF triples into questions and use them for training [19]. Finally, many systems work with distributed vector representations for words and RDF triples and use various deep learning techniques for answer selection. A common strategy is to use a joint embedding of text and knowledge base concepts. For example, character n-gram text representation as input to a convolutional neural network can capture the gist of the question and help map phrases to entities and predicates [125]. Joint embeddings can be trained using multi-task learning, *e.g.* a system can learn to embed a question and candidate answer subgraph using question-answer pairs and question paraphrases at the same time ([19]). Memory Networks, developed by the

[5]http://www.wikidata.org
[6]www.sc.cit-ec.uni-bielefeld.de/qald/
[7]https://answers.wikia.com/

Facebook AI Lab, can also be used to return triples stored in network memory in a response to the user question [20]. This approach uses embeddings of predicates and can answer relatively simple questions, that do not contain any constraints and aggregations. To extend deep learning framework to more complex questions, [35] use multi-column convolutional neural network to capture the embedding of the entity path, context and type.

As for the architecture of KBQA systems, two major approaches have been identified: semantic parsing and information extraction. Semantic parsing starts from question utterances and work to produce the corresponding semantic representation, *e.g.* logical form. The model of [12] uses a CCG parser, which can produce many candidates on each level of parsing tree construction. A common strategy is to use beam search to keep top-k options on each parsing level or agenda-based parsing [14], which maintains current best parses across all levels. An alternative information extraction strategy was proposed by [122], which can be very effective for relatively simple questions. A comparison of this approaches can be found in [121]. The idea of the information extraction approach is that for most of the questions the answer lies in the neighborhood of the question topic entity. Therefore, it is possible to use a relatively small set of query patterns to generate candidate answers, which are then ranked using the information about how well involved predicates and entities match the original question.

Question entity identification and disambiguation is the key component in such systems, they cannot answer the question correctly if the question entity isn't identified. Different systems used NER to tag question entities, which are then linked to a knowledge base using a lexicon of entity names [12, 13, 115]. However, NER can easily miss the right span and the whole system would fail to produce the answer. Recently, most of the state-of-the-art system on WebQuestions dataset used a strategy to consider a reasonable subset of token n-grams, each of which can map to zero or more KB entities, which are disambiguated on the answer ranking stage [120, 9, 99]. Ranking of candidates can be done using a simple linear classification model [120] or a more complex gradient boosted trees ranking model [9, 99].

Some questions contain certain conditions, that require special filters or aggregations to be applied to a set of entities. For example, the question "*who won 2011 heisman trophy?*" contains a date, that needs to be used to filter the set of heisman trophy winners, the question "*what high school did president bill clinton attend?*" requires a filter on the entity type to filter high schools from the list of educational institutions, and "*what is the closest airport to naples florida?*" requires a set of airports to be sorted by distance and the closest one to be selected. Information extraction approaches either needs to extend the set of candidate query templates used, which is usually done manually, or to attach such aggregations later in the process, after the initial set of entities have been extracted [99]. An alternative strategy to answer complex questions is to extend RDF triples as a unit of knowledge with additional arguments and perform question answering over n-tuples [126]. In [109] authors propose to start from single KB facts and build more complex logical formulas by combining existing ones, while scoring candidates using paraphrasing model. Such a a template-free model combines the benefits of semantic parsing and information extraction approaches.

2.1.3 Hybrid question answering

A natural idea of combining available information sources to improve question answering has been explored for a long time. WordNet lexical database [75] was among the first resources, that were adapted by QA community [56, 78], and it was used for such tasks as query expansion and definition extraction. Wikipedia[8], which can be characterized as an unstructured and semi-structured (infoboxes) knowledge base, quickly became a valuable resource for answer extraction and verification [2, 24]. Developers of the Aranea QA system noticed that structured knowledge bases are very effective in answering a significant portion of relatively simple questions [70]. They designed a set of regular expressions for popular questions that can be efficiently answered from a knowledge base and fall back to regular text-based methods for the rest of the questions.

One of the major drawbacks of knowledge bases is their incompleteness, which means that many entities, predicates and facts are missing from knowledge bases, which limits the number of questions one can answer using them. One approach to increase the coverage of knowledge bases is to extract information from other resources, such as raw text[76, 61, 50], web tables[26], *etc.* . However, the larger the knowledge base gets, the more difficult it's to find a mapping from natural language phrases to KB concepts. Alternatively, open information extraction techniques ([38]) can be used to extract a surface form-based knowledge base, which can be very effective for question answering. Open question answering approach of [40] combines multiple structured (Freebase) and unstructured (OpenIE) knowledge bases together by converting them to string-based triples. User question can be first paraphrased using paraphrasing model learned from WikiAnswers data, then converted to a KB query and certain query rewrite rules can be applied, and all queries are ranked by a machine learning model.

SPOX tuples, proposed in [116], encode subject-predicate-object triples along with certain keywords, that could be extracted from the same place as RDF triple. These keywords encode the context of the triple and can be used to match against keywords in the question. The method attempts to parse the question and uses certain relaxations (removing SPARQL triple statements) along with adding question keyphrases as additional triple arguments. As an extreme case of relaxation authors build a query that return all entities of certain type and use all other question terms to filter and rank the returned list.

However, by applying information extraction to raw text we inevitably lose certain portion of the information due to recall errors, and extracted data is also sometimes erroneous due to precision errors. In [114], authors propose to use textual evidence to do answer filtering in a knowledge base question answering system. On the first stage with produce a list of answers using traditional information extraction techniques, and then each answer is scored using its Wikipedia page on how well it matches the question. Knowledge bases can also be incorporated inside TextQA systems. Modern KBs contain comprehensive entity types hierarchies, which were utilized in QuASE system of [96] for answer typing. In addition, QuASE exploited the textual descriptions of entities stored in Freebase knowledge base as answer supportive evidence 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.

Another great example of a hybrid question answering system is IBM Watson, which is arguably the most important and well-known QA systems ever developed so far. It was designed to play the Jeopardy TV show[9]. The system combined multiple different approaches, including text-based, relation extraction and knowledge base modules, each of which generated candidate answers, which are then pooled together for ranking and answer selection. The full architecture of the system is well described in [42] or in the full special issue of the IBM Journal of Research and Development [43]. YodaQA [10] is an open source implementation of the ideas behind the IBM Watson system.

[8] http://www.wikipedia.org

[9] https://en.wikipedia.org/wiki/Jeopardy!

2.2 Non-factoid question answering

During earlier days of research non-factoid questions received relatively little attention. TREC QA tasks started to incorporate certain categories of non-factoid questions, such as definition questions, during the last 4 years of the challenge. One of the first non-factoid question answering system was described in [93] and was based on web search using chunks extracted from the original question. The ranking of extracted answer candidates was done using a translation model, which showed better results than n-gram based match score.

The growth of the popularity of community question answering (CQA) websites, such as Yahoo! Answers, Answers.com, *etc.* , contributed to an increased interest of the community to non-factoid questions. Some questions on CQA websites are repeated very often and answers can easily be reused to answer new questions, [72] studies different types of CQA questions and answers and analyzes them with respect to answer re-usability. A number of methods for similar question retrieval have been proposed [15, 89, 37, 59]. WebAP is a dataset for non-factoid answer sentence retrieval, which was developed in [118]. Experiments conducted in this work demonstrated, that classical retrieval methods doesn't work well for this task, and multiple additional semantic (ESA, entity links) and context (adjacent text) features have been proposed to improve the retrieval quality.

Candidate answer passages ranking problem becomes even more difficult in non-factoid questions answering as systems have to deal with larger piece of text and need to "understand" what kind of information is expressed there. One of the first extensive studies of different features for non-factoid answer ranking can be found in [97], who explored information retrieval scores, translation models, tree kernel and other features using tokens and semantic annotations (dependency tree, semantic role labelling, *etc.*) of text paragraphs. Alignment between question and answer terms can serve as a good indicator of their semantic similarity. Such an alignment can be produced using a machine learning model with a set of features, representing the quality of the match [108]. Alignment and translation models are usually based on term-term similarities, which are often computed from a monolingual alignment corpus. This data can be very sparse, and to overcome this issue [46] proposed higher-order lexical semantic models, which estimates similarity between terms by considering paths of length more than 1 on term-term similarity graph. An alternative strategy to overcome the sparseness of monolingual alignment corpora is to use the discourse relations of sentences in a text to learn term association models [86].

Questions often have some metadata, such as categories on a community question answering website. This information can be very useful for certain disambiguations, and can be encoded in the answer ranking model [129]. The structure of the web page, from which the answers are extracted can be very useful as well. Wikipedia articles have a good structure, and the information encoded there can be extracted in a text-based knowledge base, which can be used for question answering [92]. Information extraction methods can also be useful for the more general case of non-factoid question answering. For example, there is a huge number of online forums, FAQ-pages and social media, that contain question-answer pairs, which can be extracted to build a collection to query when a new question arrives [31, 60, 117, 34, 66].

Most of the approaches from TREC LiveQA 2015 combined similar question retrieval and web search techniques [112, 85, 103]. Answers to similar questions are very effective for answering new questions [85]. However, we a CQA archive doesn't have any similar questions, we have to fall back to regular web search. The idea behind the winning system of CMU [103] is to represent each answer with a pair of phrases - clue and answer text. Clue is a phrase that should be similar to the given question, and the passage that follows should be the answer to this question.

2.3 User interactions

IN PROGRESS...

An interesting approach for knowledge base construction through dialog with the user has been proposed by [53].

A very nice crowdsourcing method to obtain answers to tail information needs was proposed by [16]. Question query-url pairs are first mined from query logs, and then the wisdom of a crowd is used to extract and save answers to these questions.

Certain questions cannot be answered by machines only due to reasons such as lack of the appropriate data. Crowdsourcing was explored as one of the options to bridge this knowledge gap and assist the machine in matching, ranking and result aggregation [45]. Wisdom of a crowd can also be exploited to answer more difficult quiz questions [7].

[21] retrieves Wikipedia statements that support user answers for opinion questions.

A number of works have focused on studying and estimating different factors of user satisfaction with question answering systems [77, 71]

Interactive TREC ciQA...

2.4 Summary of Related Work

Most previous work in ...

3 Structured and Unstructured Data for Factoid Question Answering

There are multiple ways to marry unstructured and structured data for joint question answering: convert structured data to unstructured format or vice versa, convert all data to certain intermediate representation or to leave them as is and link the data sources. In my thesis I focus on two approaches: relation extraction for knowledge base completion, and semantic annotation of text for hybrid question answering.

3.1 Relation Extraction for Knowledge Base Completion

The information on the web is stored in multiple different forms, such as natural language statements, tables and infoboxes, images *etc.* . In this work I focus on yet another source of information: question-answer pairs. Community question answering archives contain hundreds of millions of question and corresponding answers. Information expressed in these pairs might be hard to extract or not exist at all in other formats.

3.1.1 Relation extraction from Question-Answer pairs

PUT SUMMARY OF MY NAACL STUDENT RESEARCH WORKSHOP PAPER.

3.2 Semantic Text Annotations for Hybrid Question Answering

Converting unstructured information into structured form by extracting knowledge from text suffers from certain quality losses. Existing relation extraction tools aren't perfect, in particular due to recall losses a lot of information is left behind. Moreover, extractions contain certain level of incorrect information due to precision losses. These errors cap the upper bound on the question answering system performance.

Here I propose to utilize the synergy of structured and unstructured data, and exploit the advantages of each of them to overcome the limitations of the other. More particularly, I propose to annotate and index mentions of knowledge base entities in text documents. Such a representation induces a special kinds of edges to the knowledge base, and allows one to traverse this edges in both directions. These links open up many opportunities for QA reasoning, *e.g.* retrieving all the information about the entity by going from a mention to a KB entity, finding relations between entities by retrieving text passages that mention both of them, extracting candidate evidence by retrieving passages that mention question and answer entities along with some question terms, and so on.

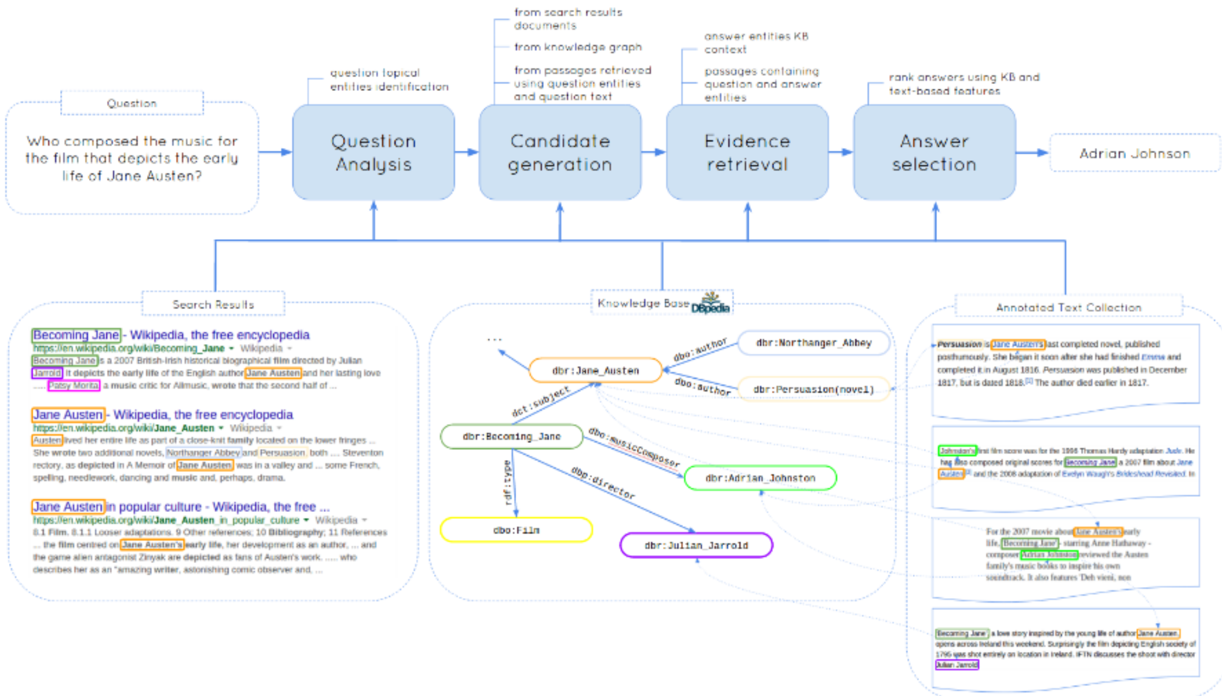

Figure 3.1: Architecture of a hybrid factoid question answering system, that uses a combination of structured knowledge base and unstructured text data

3.2.1 Approach

The architecture of the hybrid QA model I propose is presented on Figure 3.1. Here are the main stages of the question answering process:

- **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 [32] 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. [100].
- **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 neighborhood 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 mentioning question and answer entities and the question text
- **Rank candidate**: rank candidate answers using evidence extracted from the KB as well as from text

Let's consider an example question "*Who composed the music for the film that depicted the early life of Jane Austen?*" from the QALD dataset[1] (Figure 3.1). Even though it's quite easy to identify the "`Jane Austen`" entity in the question, the knowledge base (dbPedia in this example) cannot help us to determine which movie is being referred to. However, there are a lot of documents on the web, that do mention the "`Becoming Jane`" movie and say what is it about. Unfortunately, extracting the name of the composer from these documents is quite challenging, but this task can be easily accomplished by checking the value of the `musicComposer` property in the knowledge base. At the end, for each candidate answer entity, we have all the KB information and passages that mention this entity as evidence to help with the correct answer selection.

3.2.2 Evaluation

Knowledge base QA

I THINK HERE WE CAN INCLUDE OUR RESULTS ON TEXT2KB.

Most of the recent work on knowledge base question answering and semantic parsing have been evaluated on the WebQuestions dataset [12], 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[2] The proposed approach will be compared against the previous results[3] 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 [47].

New factoid question answering dataset. 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.

Text-based QA

WE WILL ANNOTATE TREC DATASETS WITH ENTITIES!!!!

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 [40, 96]. The proposed system can use the web as the corpus and query it using Bing Search API[4]. Freebase and Reverb extractions [39] are

[1]http://greententacle.techfak.uni-bielefeld.de/ cunger/qald/
[2]http://mturk.com/
[3]http://goo.gl/sePBja
[4]https://datamarket.azure.com/dataset/bing/searchweb

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[5], 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.

3.3 Summary

In this section we considered two different ways of combining unstructured and structured data to improve factoid question answering. Relation extraction from question-answer pairs aims at filling some gaps in KB fact coverage, whereas semantic annotations of text documents provides a way to incorporate information available in unstructured text documents for reasoning along with KB data to improve the performance of factoid question answering.

Factoid questions represent just a part of user information needs. Many problems require more elaborate response, such as a sentence, list of instructions or in general a passage of text. Such questions are usually referred to as non-factoid questions and they will be the focus of the next Chapter.

[5] http://trec-liveqa.org/

4 Non-factoid Question Answering

In this chapter I summarize the proposed work in developing a non-factoid question answering system. In particular, Section 4.1 described the general architecture of the system, and the following chapters describe the proposed improvements to different stages of the pipeline.

4.1 The architecture of the system

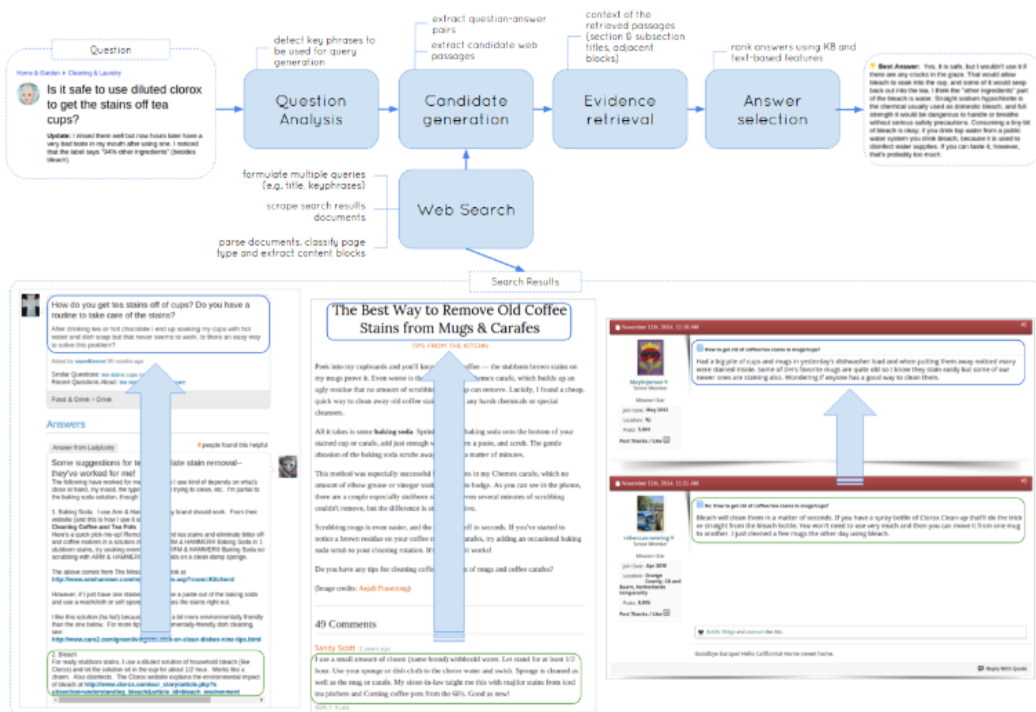

Figure 4.1: Using web page structure information for non-factoid question answering

The architecture of the system I develop is somewhat standard and contains 4 main stages: question analysis, candidate generation, evidence retrieval and answer selection. Figure 4.1 depicts the question-answering pipeline.

4.1.1 Question Analysis

Questions that users post to CQA websites often contain title and body and can be relatively long and contain multiple important and not so important context information and details. The success of the retrieval-based question answering system depends on the information it finds in a collection. Extra long search queries are not very efficient and can return few or zero results. Therefore, there is a problem of summarizing the user question. Some strategies often used include considering first part of the question, *e.g.* title of the question only. However, often the most important part

of the question isn't the title, or the title doesn't contain all the crucial information, *e.g.*
Title: *Diet please please help asap?*
Body: *I want to lose weight, at least 4 stone but I don't know what I should eat :(what should I have for breakfast lunch and dinner? Should I exercise a little? Please help!!*

To summarize the question I propose to use recent advances in the field of deep learning, in particular a model similar to [83]. In more detail, I propose to train neural network based summarization model to generate the summary of the question, which will be used as a query to retrieve similar questions. Such a model can be specifically trained to maximize retrieval performance on a collection of question-answer pairs, retrieved by systems in LiveQA TREC 2015 and labeled by NIST assessors.

4.1.2 Structure of the Web Page for Candidate Generation and Scoring

The diversity and complexity of non-factoid questions pose additional challenges for automatic question answering systems. To answer such questions a system often needs to provide a whole paragraph of text, *e.g.* TREC LiveQA'15 limits the answers to a maximum of 1000 characters. Therefore, a candidate answer becomes much longer, which requires additional attention on candidate generation and ranking stages.

Existing techniques usually extract one or more consecutive sentences not exceeding the maximum answer length, pool them together and rank using certain feature representation, by large ignoring the context information from the page where the answer was extracted from. In the system I develop I propose to utilize the structure of the web page for both candidate generation and scoring.

To generate better candidates I'm going to utilize the structure of retrieved web pages. The previous analysis showed [85], that many search results retrieved for the question are FAQs, forums or other community question answering websites. From such resources it's beneficial to extract question answer pairs, which can be done either be designing wrapper for popular websites, utilizing semantic annotations, such as https://schema.org or by applying some of the automatic question-answer extraction methods [31]. For other resources, page segmentation techniques, similar to [28], can split a page into semantically information blocks, which will help to make sure that candidate don't contain disjoint and unreadable information.

After a candidate passage is extracted, we need to score it as a potential answer. Often, a passage taken out of context is hard to understand even for human. Therefore, I propose to include the context information, *i.e.* some features, representing the web page where the passage was taken from and its location there (*e.g.* adjacent passages, which are often related [118]).

4.1.3 Answer Summarization

Unlike factoid questions, where evidence from all retrieved passages is usually aggregates, a traditional non-factoid question-answering system simply returns the top scoring passage as the answer. However, such an answer can contain a lot of redundant or irrelevant information, whereas other good candidate passages may contain supplemental information or different opinion. Therefore, an idea to summarize passages and generate the final answer sounds natural in this scenario. The winning approach from TREC LiveQA'15 included a combination strategy, that simply put together multiple top scoring passages while the answer doesn't exceed the maximum length [103]. In my thesis I'm planning to explore the answer summarization problem in more detail. More specifically, I propose to explore deep learning techniques, that lie in the core of recent successes in text generation, *e.g.* image caption generation and machine translation models [8, 113]. Answer summarization problem can be posed as answer generation problem using recurrent neural network, that has information about the question and retrieved passages, and it can be trained using existing CQA question-answer pairs and retrieved passages, that can be assumed as the source of the answers. Alternative and simpler approach is to solve this problem as sequential answer selection problem, where a model is trained to predict the next sentence in the answer. Such a model can be trained on a collection of questions and answer sentences, which will provide the model information on both answer discourse coherence and relevance.

4.2 Evaluation

Evaluation of complete non-factoid question answering systems is complicated due to the variability of answer language, the quality of which is impossible to estimate using manually created answer patterns (as is the case for factoid TREC QA dataset). A manual judgment of answers is needed, and luckily TREC LiveQA 2016 is offering such an opportunity. The model I'm developing will participate in the shared task, which will allow us to evaluate it against other competing approaches.

The analysis of individual components can be performed using labeled data from TREC LiveQA 2015, which includes passages extracted from Yahoo! Answers as well as regular web documents. In more detail, the performance of the answer summarization module will be estimated by similar questions retrieval performance, *e.g.* Precision @ N since we are interested in retrieving more relevant answers in TopN rather than good ranking within the retrieved group. To make results reproducible a collection of Yahoo! Answers QnA pairs from the WebScope[1] and Lucene IR library will be used for question retrieval.

The dataset to evaluate the effectiveness of using web page structure for answer scoring will be derived from TREC LiveQA 2015 labels, which include a big number of passages, that were generated from regular web pages. Therefore, the problem of evaluation of answer scoring methods can be posed as passage ranking problem and metrics, such as Precision@1, can be used as a quality measure.

Finally, the answer summarization module is the most difficult to evaluate, because it's result is a free form text, the quality of which needs to be manually labeled. Therefore, for this task I will refer to the wisdom of a crowd and use Amazon Mechanical Turk to label the quality of answers and compare to the top scoring passages for the same question.

4.3 Summary

The proposed research directions target various aspects of non-factoid question answering pipeline and can help improve both precision and recall of existing systems. The results of this work can help to move in the direction of systems, that produce a natural language response by analyzing

[1] http://webscope.sandbox.yahoo.com/catalog.php?datatype=l

the information available on the web. In the next chapter, I will describe the proposed work and results on interactions between a question answering system and users.

5 Human Interaction with Question Answering Systems

Modern automatic question answering systems are still far from AI machines, that we often imagine or see in the movies. Many user information needs are still left unanswered by existing automatic question answering systems. For example, only 36% of answers of a winning approach from TREC LiveQA 2015 shared task were judged good or excellent. Therefore, in this chapter I will discuss the research on interactions between a human and a question answering system.

More specifically, Section 5.1 describes results on studying the effect of hints on user behavior for solving complex informational tasks. In Section 5.2 I propose research on using crowdsourcing for improving the performance of a question answering system. And Section **??** shifts the focus towards a dialogue between a user and a QA system, in particular towards clarification questions, that are often needed to understand user's question.

5.1 Search Hints for Complex Informational Tasks

SUMMARIZE SEARCH HINTS PAPER.

5.2 Using the Wisdom of a Crowd for Question Answering

This is our experiment for LiveQA answer crowdsourcing.

5.3 Clarification Questions

This is the work of Pavel Braslavsky.

6 Discussion and Implication

OLD, NEED REVISION

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] A. M. N. Allam and M. H. Haggag. The question answering systems: A survey. *International Journal of Research and Reviews in Information Sciences (IJRRIS)*, 2(3), 2012.

[4] A. Andrenucci and E. Sneiders. Automated question answering: Review of the main approaches. In *null*, pages 514–519. IEEE, 2005.

[5] I. Androutsopoulos, G. D. Ritchie, and P. Thanisch. Natural language interfaces to databases–an introduction. *Natural language engineering*, 1(01):29–81, 1995.

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

[7] B. I. Aydin, Y. S. Yilmaz, Y. Li, Q. Li, J. Gao, and M. Demirbas. Crowdsourcing for multiple-choice question answering. In *AAAI*, pages 2946–2953, 2014.

[8] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *ICLR*, 2015.

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

[10] P. Baudiš. Yodaqa: a modular question answering system pipeline. In *POSTER 2015-19th International Student Conference on Electrical Engineering*, pages 1156–1165, 2015.

[11] 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.

[12] 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.

[13] 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.

[14] J. Berant and P. Liang. Imitation learning of agenda-based semantic parsers. *Transactions of the Association for Computational Linguistics*, 3:545–558, 2015.

[15] D. Bernhard and I. Gurevych. Combining lexical semantic resources with question & answer archives for translation-based answer finding. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 728–736. Association for Computational Linguistics, 2009.

[16] M. S. Bernstein, J. Teevan, S. Dumais, D. Liebling, and E. Horvitz. Direct answers for search queries in the long tail. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 237–246. ACM, 2012.

[17] M. W. Bilotti, P. Ogilvie, J. Callan, and E. Nyberg. Structured retrieval for question answering. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 351–358. ACM, 2007.

[18] 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.

[19] 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.

[20] A. Bordes, N. Usunier, S. Chopra, and J. Weston. Large-scale simple question answering with memory networks. *arXiv preprint arXiv:1506.02075*, 2015.

[21] L. Braunstain, O. Kurland, D. Carmel, I. Szpektor, and A. Shtok. Supporting human answers for advice-seeking questions in cqa sites. In *Advances in Information Retrieval*, pages 129–141. Springer, 2016.

[22] E. Brill, S. Dumais, and M. Banko. An analysis of the askmsr question-answering system. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 257–264. Association for Computational Linguistics, 2002.

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

[24] 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.

[25] M. J. Cafarella, A. Halevy, D. Z. Wang, E. Wu, and Y. Zhang. Webtables: Exploring the power of tables on the web. *Proc. VLDB Endow.*, 1(1):538–549, Aug. 2008.

[26] M. J. Cafarella, A. Halevy, D. Z. Wang, E. Wu, and Y. Zhang. Webtables: exploring the power of tables on the web. *Proceedings of the VLDB Endowment*, 1(1):538–549, 2008.

[27] M. J. Cafarella, J. Madhavan, and A. Halevy. Web-scale extraction of structured data. *SIGMOD Rec.*, 37(4):55–61, Mar. 2009.

[28] D. Cai, S. Yu, J.-R. Wen, and W.-Y. Ma. Vips: a visionbased page segmentation algorithm. Technical report, Microsoft technical report, MSR-TR-2003-79, 2003.

[29] 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.

[30] Q. Cai and A. Yates. Large-scale semantic parsing via schema matching and lexicon extension. In *ACL (1)*, pages 423–433. Citeseer, 2013.

[31] G. Cong, L. Wang, C.-Y. Lin, Y.-I. Song, and Y. Sun. Finding question-answer pairs from online forums. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 467–474. ACM, 2008.

[32] 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*, 2014.

[33] 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.

[34] S. Ding, G. Cong, C.-Y. Lin, and X. Zhu. Using conditional random fields to extract contexts and answers of questions from online forums. In *ACL*, volume 8, pages 710–718. Citeseer, 2008.

[35] L. Dong, F. Wei, M. Zhou, and K. Xu. Question answering over freebase with multi-column convo-

lutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, volume 1, pages 260–269, 2015.

[36] 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.

[37] H. Duan, Y. Cao, C.-Y. Lin, and Y. Yu. Searching questions by identifying question topic and question focus. In *ACL*, pages 156–164, 2008.

[38] O. Etzioni, M. Banko, S. Soderland, and D. S. Weld. Open information extraction from the web. *Commun. ACM*, 51(12):68–74, Dec. 2008.

[39] 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.

[40] 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.

[41] A. Fader, L. S. Zettlemoyer, and O. Etzioni. Paraphrase-driven learning for open question answering. In *ACL*. Citeseer, 2013.

[42] D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager, et al. Building watson: An overview of the deepqa project. *AI magazine*, 31(3):59–79, 2010.

[43] D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager, et al. This is watson. *IBM Journal of Research and Development*, 56, 2012.

[44] 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.

[45] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin. Crowddb: Answering queries with crowdsourcing. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*, SIGMOD '11, pages 61–72, New York, NY, USA, 2011. ACM.

[46] D. Fried, P. Jansen, G. Hahn-Powell, M. Surdeanu, and P. Clark. Higher-order lexical semantic models for non-factoid answer reranking. *Transactions of the Association for Computational Linguistics*, 3:197–210, 2015.

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

[48] B. F. Green Jr, A. K. Wolf, C. Chomsky, and K. Laughery. Baseball: an automatic question-answerer. In *Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference*, pages 219–224. ACM, 1961.

[49] P. Gupta and V. Gupta. A survey of text question answering techniques. *International Journal of Computer Applications*, 53(4), 2012.

[50] R. Gupta, A. Halevy, X. Wang, S. E. Whang, and F. Wu. Biperpedia: An ontology for search applications. *Proc. VLDB Endow.*, 7(7):505–516, Mar. 2014.

[51] M. Heilman and N. A. Smith. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1011–1019. Association for Computational Linguistics, 2010.

[52] L. Hirschman and R. Gaizauskas. Natural language question answering: the view from here. *natural language engineering*, 7(4):275–300, 2001.

[53] B. Hixon, P. Clark, and H. Hajishirzi. Learning knowledge graphs for question answering through conversational dialog. In *Proceedings of the the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA*, 2015.

[54] E. Hovy, U. Hermjakob, and D. Ravichandran. A question/answer typology with surface text patterns. In *Proceedings of the Second International Conference on Human Language Technology Research*, HLT '02, pages 247–251, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

[55] 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.

[56] E. H. Hovy, U. Hermjakob, and C.-Y. Lin. The use of external knowledge of factoid qa. In *TREC*, volume 2001, pages 644–52, 2001.

[57] A. Ittycheriah, M. Franz, and S. Roukos. Ibm's statistical question answering system-trec-10. In *TREC*, 2001.

[58] M. Iyyer, J. L. Boyd-Graber, L. M. B. Claudino, R. Socher, and H. Daumé III. A neural network for factoid question answering over paragraphs. In *EMNLP*, pages 633–644, 2014.

[59] J. Jeon, W. B. Croft, and J. H. Lee. Finding similar questions in large question and answer archives. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, CIKM '05, pages 84–90, New York, NY, USA, 2005. ACM.

[60] V. Jijkoun and M. de Rijke. Retrieving answers from frequently asked questions pages on the web. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, CIKM '05, pages 76–83, New York, NY, USA, 2005. ACM.

[61] V. Jijkoun, M. De Rijke, and J. Mur. Information extraction for question answering: Improving recall through syntactic patterns. In *Proceedings of the 20th international conference on Computational Linguistics*, page 1284. Association for Computational Linguistics, 2004.

[62] M. Keikha, J. H. Park, W. B. Croft, and M. Sanderson. Retrieving passages and finding answers. In *Proceedings of the 2014 Australasian Document Computing Symposium*, page 81. ACM, 2014.

[63] 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.

[64] N. Kushmerick. *Wrapper induction for information extraction.* PhD thesis, University of Washington, 1997.

[65] C. Kwok, O. Etzioni, and D. S. Weld. Scaling question answering to the web. *ACM Transactions on Information Systems (TOIS)*, 19(3):242–262, 2001.

[66] B. Li, X. Si, M. R. Lyu, I. King, and E. Y. Chang. Question identification on twitter. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 2477–2480. ACM, 2011.

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

[68] X. Li and D. Roth. Learning question classifiers: the role of semantic information. *Natural Language Engineering*, 12(3):229–249, 2006.

[69] J. Lin. An exploration of the principles underlying redundancy-based factoid question answering. *ACM Transactions on Information Systems (TOIS)*, 25(2):6, 2007.

[70] J. Lin and B. Katz. Question answering from the web using knowledge annotation and knowledge mining techniques. In *Proceedings of the twelfth international conference on Information and knowledge management*, pages 116–123. ACM, 2003.

[71] Y. Liu, J. Bian, and E. Agichtein. Predicting information seeker satisfaction in community question answering. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 483–490, New York, NY, USA, 2008. ACM.

[72] Y. Liu, S. Li, Y. Cao, C.-Y. Lin, D. Han, and Y. Yu. Understanding and summarizing answers in community-based question answering services. In *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 497–504, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

[73] F. Mahdisoltani, J. Biega, and F. Suchanek. Yago3: A knowledge base from multilingual wikipedias. In *7th Biennial Conference on Innovative Data Systems Research*. CIDR Conference, 2014.

[74] C. Malon and B. Bai. Answer extraction by recursive parse tree descent. *ACL 2013*, page 110, 2013.

[75] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[76] 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.

[77] C.-S. Ong, M.-Y. Day, and W.-L. Hsu. The measurement of user satisfaction with question answering systems. *Information & Management*, 46(7):397–403, 2009.

[78] 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.

[79] 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.

[80] J. M. Prager. Open-domain question-answering. *Foundations and trends in information retrieval*, 1(2):91–231, 2006.

[81] V. Punyakanok, D. Roth, and W.-t. Yih. Mapping dependencies trees: An application to question answering. In *Proceedings of AI&Math 2004*, pages 1–10, 2004.

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

[83] A. M. Rush, S. Chopra, and J. Weston. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[84] C. d. Santos, M. Tan, B. Xiang, and B. Zhou. Attentive pooling networks. *arXiv preprint arXiv:1602.03609*, 2016.

[85] D. Savenkov. Ranking answers and web passages for non-factoid question answering: Emory university at trec liveqa. In *Proceedings of TREC*, 2015.

[86] R. Sharp, P. Jansen, M. Surdeanu, and P. Clark. Spinning straw into gold: Using free text to train monolingual alignment models for non-factoid question answering. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics-Human Language Technologies (NAACL HLT)*, 2015.

[87] D. Shen, G.-J. M. Kruijff, and D. Klakow. Exploring syntactic relation patterns for question answering. In *Natural Language Processing–IJCNLP 2005*, pages 507–518. Springer, 2005.

[88] E. H. Shortliffe and B. G. Buchanan. A model of inexact reasoning in medicine. *Mathematical biosciences*, 23(3):351–379, 1975.

[89] 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*, WWW '12, pages 759–768, New York, NY, USA, 2012. ACM.

[90] R. F. Simmons. Answering english questions by computer: A survey. *Communications of ACM*, 8(1):53–70, Jan. 1965.

[91] R. F. Simmons. Natural language question-answering systems: 1969. *Commun. ACM*, 13(1):15–30, Jan. 1970.

[92] P. Sondhi and C. Zhai. Mining semi-structured online knowledge bases to answer natural language questions on community qa websites. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 341–350. ACM, 2014.

[93] R. Soricut and E. Brill. Automatic question answering using the web: Beyond the factoid. *Information Retrieval*, 9(2):191–206, 2006.

[94] M. M. Soubbotin and S. M. Soubbotin. Patterns of potential answer expressions as clues to the right answers. In *TREC*, 2001.

[95] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM, 2007.

[96] 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.

[97] M. Surdeanu, M. Ciaramita, and H. Zaragoza. Learning to rank answers to non-factoid questions from web collections. *Computational Linguistics*, 37(2):351–383, 2011.

[98] M. Tan, B. Xiang, and B. Zhou. Lstm-based deep learning models for non-factoid answer selection. *arXiv preprint arXiv:1511.04108*, 2015.

[99] 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.

[100] 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.

[101] C. Unger, A. Freitas, and P. Cimiano. An introduction to question answering over linked data. In *Reasoning Web. Reasoning on the Web in the Big Data Era*, pages 100–140. Springer, 2014.

[102] E. M. Voorhees. The trec question answering track. *Natural Language Engineering*, 7(04):361–378, 2001.

[103] D. Wang and E. Nyberg. Cmu oaqa at trec 2015 liveqa: Discovering the right answer with clues. In *Proceedings of TREC*, 2015.

[104] 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.

[105] M. Wang. A survey of answer extraction techniques in factoid question answering. *Computational Linguistics*, 1(1), 2006.

[106] M. Wang and C. D. Manning. Probabilistic tree-edit models with structured latent variables for textual entailment and question answering. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1164–1172. Association for Computational Linguistics, 2010.

[107] M. Wang, N. A. Smith, and T. Mitamura. What is the jeopardy model? a quasi-synchronous grammar for qa. In *EMNLP-CoNLL*, volume 7, pages 22–32, 2007.

[108] Z. Wang and A. Ittycheriah. Faq-based question answering via word alignment. *arXiv preprint arXiv:1507.02628*, 2015.

[109] Z. Wang, S. Yan, H. Wang, and X. Huang. Large-scale question answering with joint embedding and proof tree decoding. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1783–1786. ACM, 2015.

[110] R. Wilensky, D. N. Chin, M. Luria, J. Martin, J. Mayfield, and D. Wu. The berkeley unix consultant project. *Computational Linguistics*, 14(4):35–84, 1988.

[111] W. A. Woods and R. Kaplan. Lunar rocks in natural english: Explorations in natural language question answering. *Linguistic structures processing*, 5:521–569, 1977.

[112] G. Wu and M. Lan. Leverage web-based answer retrieval and hierarchical answer selection to improve the performance of live question answering. In *Proceedings of TREC*, 2015.

[113] K. Xu, J. Ba, R. Kiros, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*, 2015.

[114] K. Xu, Y. Feng, S. Reddy, S. Huang, and D. Zhao. Enhancing freebase question answering using textual evidence. *arXiv preprint arXiv:1603.00957*, 2016.

[115] K. Xu, S. Zhang, Y. Feng, and D. Zhao. Answering natural language questions via phrasal semantic parsing. In *Natural Language Processing and Chinese Computing*, pages 333–344. Springer, 2014.

[116] M. Yahya, K. Berberich, S. Elbassuoni, and G. Weikum. Robust question answering over the web of linked data. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 1107–1116. ACM, 2013.

[117] J.-M. Yang, R. Cai, Y. Wang, J. Zhu, L. Zhang, and W.-Y. Ma. Incorporating site-level knowledge to extract structured data from web forums. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 181–190, New York, NY, USA, 2009. ACM.

[118] L. Y. Yang, Q. Ai, D. Spina, R.-C. Chen, L. Pang, W. B. Croft, J. Guo, and F. Scholer. Beyond factoid qa: Effective methods for non-factoid answer sentence retrieval. In *Proceedings of ECIR'16*, 2016.

[119] Y. Yang, W.-t. Yih, and C. Meek. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018. Citeseer, 2015.

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

[121] X. Yao, J. Berant, and B. Van Durme. Freebase qa: Information extraction or semantic parsing? *ACL 2014*, page 82, 2014.

[122] 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.

[123] X. Yao, B. Van Durme, C. Callison-Burch, and P. Clark. Answer extraction as sequence tagging with tree edit distance. In *HLT-NAACL*, pages 858–867. Citeseer, 2013.

[124] X. Yao, B. Van Durme, and P. Clark. Automatic coupling of answer extraction and information retrieval. In *ACL (2)*, pages 159–165, 2013.

[125] W.-t. Yih, X. He, and C. Meek. Semantic parsing for single-relation question answering. In *ACL (2)*, pages 643–648. Citeseer, 2014.

[126] P. Yin, N. Duan, B. Kao, J. Bao, and M. Zhou. Answering questions with complex semantic constraints on open knowledge bases. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1301–1310. ACM, 2015.

[127] L. Yu, K. M. Hermann, P. Blunsom, and S. Pulman. Deep learning for answer sentence selection.

arXiv preprint arXiv:1412.1632, 2014.

[128] J. M. Zelle and R. J. Mooney. Learning to parse database queries using inductive logic programming. In *Proceedings of the national conference on artificial intelligence*, pages 1050–1055, 1996.

[129] G. Zhou, T. He, J. Zhao, and P. Hu. Learning continuous word embedding with metadata for question retrieval in community question answering. In *Proceedings of ACL*, pages 250–259, 2015.