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 Motivation

The ability to answer user questions with precise and concise information is a hard problem with a long history of research. Various data sources are available for candidate answer generation, two major ones are unstructured text corpora, and structured knowledge bases (e.g. dpPedia [3] and Freebase [9]). A hybrid approach to question answering [5, 26] 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 [17] or question-answer pairs [6]. Improvements were made by extending the lexicon using Wikipedia and patterns expressing certain predicates obtained via distant supervision [4, 10, 41, 44, 52]. 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.

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

[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

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

There are a number of works, focusing on the future research directions in QA, *e.g.* [13].

In [29] authors propose a hierarchical classification of question targets, the top level qtargets are abstract, semantic, syntactic, role (reason ro manner), slot. For different targets, authors build a set of answer templates. This typology was used in authors Webclopedia TREC QA system.

2.1 Factoid question answering

One of the major factoid QA system components is lexical answer type prediction. One of the most famous answer type hierarchies was developed by [34], who also designed a dataset and machine learning approach for answer type prediction.

2.1.1 Text-based question answering

2.1.2 Knowledge base question answering

There are multiple review papers on the topic of question answering over linked data, *e.g.* [46].

In the beginning of research in semantic question answering, the focus was mainly on domain-specific questions. In particular, GeoQuery[...] dataset was very popular.

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. PARALEX system ([25]) construct a lexicon from a collection of question paraphrases from WikiAnswers[1]. A somewhat backwards approach was proposed in ParaSempre model of [7]. ParaSempre rank 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.

The earlier systems were mainly trained from question annotated with the correct parse logical form, which is expensive to obtain. To learn a good mapping from input phrases to knowledge base concepts such systems need to have appropriate training data, from which they can learn such mapping. An idea to extend a trained parser with additional lexicon, trained from web and other resources, has been proposed by [17].

Most of the parses produce different results, which means that it is possible to use question-answer pairs directly [6]. This work also introduces a new question-answer dataset - WebQuestions, which quickly became a quite popular benchmark for knowledge base question answering systems. The model of [6] uses CCG parser, which can produce many candidate on each parsing stage. A common strategy is to use beam search to keep top-k options on each parsing level or agenda-based parsing [8], which maintains current best parses across all levels.

Semantic parsing approaches start from the question utterances and work to produce its logical form, which means that the actual KB data doesn't help until its queried with the parsing result and incorrect decisions made during parsing can decrease systems' performance. An alternative information extraction strategy was proposed by [52], which can be very effective for relatively

[1] https://answers.wikia.com/

simple questions. A comparison of this approaches can be found in [51]. 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's tractable to rank candidate queries that retrieve the neighborhood entities using features, that will estimate a mapping between the entities and predicates used in the query and question phrases.

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. To identify the question topical entity one needs to find a span of tokens that refers to an entity and then map it to the correct KB concept. Named entity recognition (NER) is often used on the first stage, and a lexicon trained from various resources maps the phrase to a KB entity. Neither NER nor lexicon are perfect, and decisions made on this stage of the question answering process are usually irreversible. Therefore, it's beneficial to generate a larger candidate pool and postpone the decision. For example, one can replace detected named entities with a reasonable subset of word n-grams, each of which can map to one or multiple KB entities. The disambiguation can be done on candidate answer ranking stage, where we will have the information on queries generated from each of the candidate and how well do these queries map to tokens in the question [50]. This approach was extended in [4] by considering a larger set of candidate query templates, features and adopting learning-to-rank to train the answer selection model.

Deep learning Triumph of deep learning in various areas, including natural language processing, couldn't skip question answering. 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 [53]. 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 ([10]). Memory Networks, developed by the Facebook AI Lab, can also be used to return triples stored in network memory in a response to the user question [11]. 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, [20] use multi-column convolutional neural network to capture the embedding of the entity path, context and type.

Complex questions 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 [44]. An alternative strategy to answer complex questions is to use n-tuples instead of RDF triples [54].

In [47] authors propose to start from single KB facts and build more complex logical formulas by combining existing ones, while scoring candidates using paraphrasing model. This is a template-free model, that combines the benefits of semantic parsing and information extraction approaches.

Less training data and unsupervised Using answers as a form of supervision to train knowledge base question answering system is still expensive. Distant supervision, commonly adopted for relation extraction [38], can be generalized to train semantic parsers [41], which makes it very attractive as it doesn't require any manual training data labeling and can be easily

transfered to a new domain.

2.1.3 Hybrid question answering

Information extraction is one approach of utilizing a vast amount of unstructured data to improve knowledge base question answering. A number of approaches for relation extraction for knowledge base construction have been proposed. ...

Alternatively, open information extraction techniques ([22]) can be used to extract a surface form-based knowledge base, which can be very effective for question answering. Open question answering approach of [24] 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.

In [48], authors propose to use textual evidence to do answer filtering. 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. A mapping from Wikipedia phrases and question phrases is trained. Such an approach allows one to filter highest mountain from the list of mountains if Wikipedia has the corresponding phrase.

SPOX tuples, proposed in [49], 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 questions 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. The problem of mapping question phrase and entities, types and predicates is solved using integer linear programming.

2.2 Non-factoid question answering

Some questions on CQA websites are repeated very often and answers can easily be reused, [36] studies different types of CQA questions and answers and analyzes them with respect to answer re-usability.

Some useful materials for the related work section:

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

The field of question answering has a long history of research and dates back to 60s (see [31] 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 [19]. Approaches proposed over the years can be largely classified by the type of the information used to find the answers into knowledge base and text-based systems.

2.3 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. [30, 33, 40]. Alternatively, the AskMSR system [12] (recently reviewed in [45]) 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 [35]. 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, 14]) and structured (e.g. Wordnet [39]), and demonstrated improved question answering performance. Recently [43] proposed to link entities from candidate answers to Freebase and use its type system and textual entity description for candidate scoring. However, most of the information in a KB is stored as relations between entities, therefore there is a big potential in using all available KB data to improve question answering.

2.4 Knowledge base QA

Recent development of large scale knowledge bases (e.g. dbPedia [3]) and Freebase [9]) motivated research in open domain question answering over linked data. Developed models can be compared on the annual QALD shared task [2] 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 [41, 52], question paraphrases clusters from WikiAnswers [7], Freebase triples rephrased as questions [10], and can be based on the embeddings of questions

[2] http://greententacle.techfak.uni-bielefeld.de/~cunger/qald/

and knowledge base entities and predicates [10, 44]. 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 [42]. In addition, I propose to use unlabeled text resources for candidate query ranking, which can help to generalize to unseen types of questions and questions about predicates never mentioned in the training set.

2.5 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 [26], OpenQA [24], YodaQA [5]. The main difference between such systems and the proposed research is that hybrid systems typically use separate pipelines to extract candidates from different sources and only merge the candidate set while ranking. I propose to extend the representation of each of the data sources for better candidate generation from the beginning.

3 Structured and Unstructured Data for Factoid Question Answering

Over the decades of research in factoid question answering, two relatively separate approaches have emerged: text-centric, or Text-QA 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[1], YAGO[37], Freebase[2], WikiData[3], 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 [21], 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, such as SPARQL, to query a knowledge base [6].

One way to improve the situation with knowledge base incompleteness is to extract missing information from other data sources, *e.g.* [15, 16, 21, 22, 28, 32]. 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).

3.1 Relation Extraction for Knowledge Base Construction

To tackle the first problem researchers have proposed automatic relation extraction methods, such as [38], which can automatically label sentences, that might express certain KB relation and learn a model to extract these relations. Unfortunately, not all the information available on the web is available in natural language statements. In my thesis I propose a method for automatic relation extraction from question answer pairs, which are available in big numbers on community question answering websites, forums, etc.

3.1.1 Relation extraction from Question-Answer pairs

This is my work from NAACL student research workshop.

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

3.1.2 Question-guided relation extraction

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

Hypothesis:

1. Patterns retrieved in response to the question are better in quality, we can boost them. We can try to verify this on some relation extraction dataset and questions from some query log. We can also try to use some KBQA dataset.

2. Patterns mined for questions should help question answering. This is essentially weak supervision for training knowledge base question answering using text based question answering.

Problems:

- How to extract new predicates? If we have a question, and a sentence is mentioning a pair of non-related entities, how can we make a new one?

- How to deal with more complex questions, that are not simple relations

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

This approach can also be applied to open IE. There are sentence selection methods for question answering. We can extract noun phrases (NP mentioned in question and supposedly answer NP), and then aggregate all sentences, that mention the same NPs together. Hypothesis is, that we can extract patterns, that will answer the same question for other entities.

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.

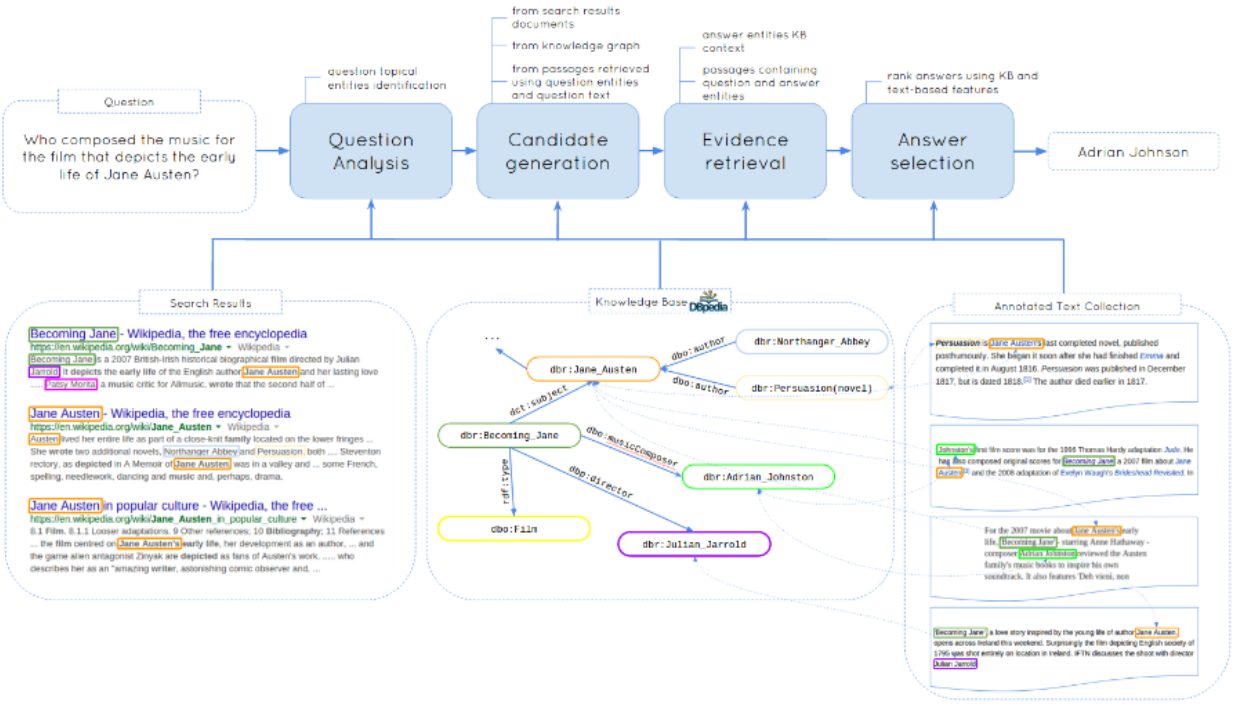

Figure 3.1: Architecture of a hybrid factoid question answering system, that uses a combination of structured knowledge base and unstructured text data

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 [18] 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. [45].
- **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[4] (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 [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[5] The proposed approach will be compared against the previous results[6] 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 [27].

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 [24, 43]. The proposed system can use the web as the corpus and query it using Bing Search API[7]. Freebase and Reverb extractions [23] are

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

[5] http://mturk.com/

[6] http://goo.gl/sePBja

[7] 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[8], 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.

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

4 Non-factoid Question Answering

Most of the questions that people have are not factoid and cannot be simply answered with a name, date or a number. Typically, such questions require a more elaborate fragment of text as an answer. Traditionally, question answering systems turn to web documents that might contain some relevant passages to be used as an answer.
In this chapter, I summarize the proposed work for improving automatic non-factoid question answering by better understanding the structure of web documents and relationships between their parts and fragments.

4.1 Utilizing the Structure of Web Pages

Non-factoid questions are typically answered with a relatively long paragraph of text[1]. 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 [**?**] 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.

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

- **CQA candidate generation**: retrieve a set of question-answer pairs by searching a CQA archive[2]
- **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 by building on techniques from existing research [**?**].

4.2 Evaluation

LiveQA

[1] TREC LiveQA'15 challenge limits the answer to 1000 characters
[2] https://answers.yahoo.com/

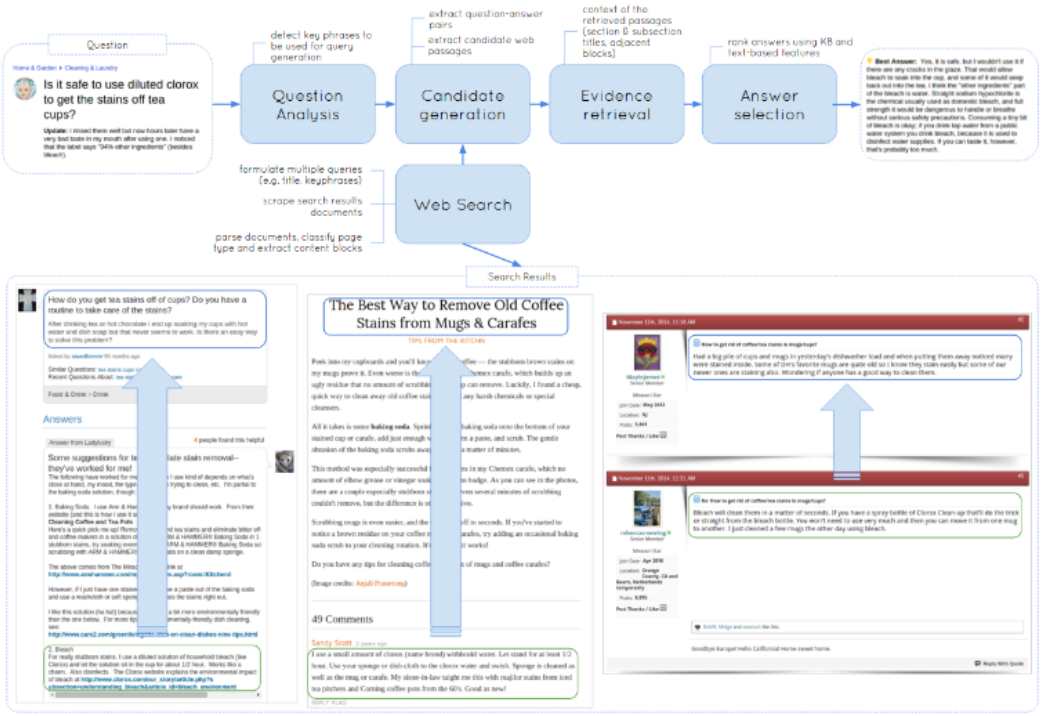

Figure 4.1: Using web page structure information for non-factoid question answering

4.3 Summary

5 Users and Question Answering Systems

5.1 Search Hints for Complex Informational Tasks

This is my SIGIR'14 short paper.

5.2 Clarification Questions

This is the work of Pavel Braslavsky.

5.3 Crowd-Sourcing Answers

This is our experiment for LiveQA answer crowdsourcing.

6 Discussion and Implication

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

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

Bibliography

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

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

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

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

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

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

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

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

[9] 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.

[10] 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.

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

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

[13] J. Burger, C. Cardie, V. Chaudhri, R. Gaizauskas, S. Harabagiu, D. Israel, C. Jacquemin, C.-Y. Lin, S. Maiorano, G. Miller, et al. Issues, tasks and program structures to roadmap research in question & answering (q&a). In *Document Understanding Conferences Roadmapping Documents*, pages 1–35, 2001.

[14] 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.

[15] 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.

[16] M. J. Cafarella, J. Madhavan, and A. Halevy. Web-scale extraction of structured data. *SIGMOD*

Rec., 37(4):55–61, Mar. 2009.

[17] 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.

[18] 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.

[19] 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.

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

[21] 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.

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

[23] 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.

[24] 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.

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

[26] 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.

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

[28] 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.

[29] 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.

[30] 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.

[31] 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.

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

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

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

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

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

[42] 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.

[43] 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.

[44] 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.

[45] 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.

[46] 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.

[47] 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.

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

[49] 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.

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

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

[52] 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.

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

[54] 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.