Final Year Project Plan

A Knowledge-based Question Answering System

Proposed by Bai Zongling
Jiang Ling
Xu Bing

The University of Hong Kong
30 Sept. 2015

Summary

A Knowledge-Based Question-Answering system (KB-QA system) is a system that answers natural language questions based on data provided by a knowledge base. This project will study various components of a classic KB-QA system and implement a demonstrable prototype of such system. The baseline of the prototype is that only fact-based questions ("factoid" questions) such as "Who is the president of US?" could be answered. Several open source tools and frameworks (i.e., Stanford CoreNLP, GATE, OpenNLP, etc.) will be used after further analyzation. A few limitations of previous research will be chosen to be further explored. For example, we expect to find a way to store data that can achieve a better balance between the volume of database and system response time. A detailed final report, a demonstrable KB-QA prototype and a well-organized website will be delivered to show our project.

1.Introduction

In the Age of Big Data, extraction of accurate and useful information from numerous sources becomes increasingly important. Mainstream information retrieval techniques are popular search engines like Google and Yahoo. However, they are facing their limitations: search engines are unable to understand natural language question and provide a short answer. The engines usually retrieve tons of relevant documents and the users have to go through those documents to find the necessary information. In contrast, a question answering system could process human language and generate an intuitive answer based on its knowledge base. Users can benefit from QA system because it eliminates user's overhead of information filtering, processing and integrating, all handled by the system. In this project we first study the basic components of a KB-QA system and then implement a KB-QA system prototype. The remainder of the project plan is organized as follows. Section 2 presents related works. Section 3 describes the scope of the project, while section 4 indicating deliverables at different stages and presenting a detailed schedule. Section 5 presents the main approach adopted. Finally section 6 evaluates possible risks and challenges and prepares corresponding solutions.

2.Related Research

There are mainly three challenges lying before we managed to let machines answering human being's questions. Naturally, they are how to understand the question, how to identify the information location and how to grab information.

One fundamental component of a QA system is the source of information. This is the partition separating two types of QA system which are knowledge-based question answering (KBQA) system and information retrieval based question answering (IRQA) system.[4]A chart below compares their pros and cons.

category	pros	cons
Specific corpus, ontology (like YAGO)	● more accurate for specific domain ● fast retrieving time	● limited scope ● human highly involved ● information not timely
open source knowledge base (need information extractor to extract so many assertions)	● timely ● large scope ● less human factor involved	noisy knowledge due to huge resources

As in the chart, many structured KBs have been released, including YAGO[5], DBpedia[6], and Freebase[7]. They are brought up and verified by human, thus KBQA could achieve more accuracy than IRQAs. However, one requirement for KBQA's successful responding is that we need to translate user's sentences and phrases to match the entities and properties stored in the KB. A good tool is PATTY1.Previous works also used other methods such as pattern matching, but it's still difficult to be sufficiently accurate. Especially when QA is asked a complex question like 'who is the architect of the tallest building in China?', system needs to try matching predicate and produce queries one by one, which results in unpredictable errors.

Each type of knowledge base has their own advantages and disadvantages. There is the balance between volume and accuracy. Although the World Wide Web is not as structured and easily operated on as existing ontology like YAGO, its massive size brings us way more information. Data redundancy is the most important implication of the Web's volume - each item of information has potentially been stated in various ways, in many different documents. It’s not necessarily a bad news when it comes to redundancy, on the contrast, it is an advantage here. A question-answering system can take advantage of data redundancy in two ways: as a surrogate for complicated natural language processing and as a make-up for poor document quality.
Consider the question "When did Wilt Chamberlain score 100 points?" And here are two possible answers. (From MIT researchers.)

> (1) Wilt Chamberlain scored 100 points on March 2, 1962 against the New Yorks Knicks.
> (2) On December 8, 1961, Wilt Chamberlain scored 78 points in a triple overtime game. It was a new NBA record, but Warriors coach Frank McGuire didn’t expect it to last long, saying, “He’ll get 100 points someday.” McGuire’s prediction came true just a few months later in a game against the New York Knicks on March 2.

You could see that getting answer from (1) directly is easier to extract information form (2).System just need to do keyword matching or entity matching etc. But it is kind of based on large data storage to have higher chance to store the corresponding sentences which almost match the question exactly. Otherwise the system is forced to do the complex work such as natural language processing, performing extra calculations.

Besides, data redundancy could serve as a guard against web information being untrustworthy. A question answering system could make use of related statements locating differently of web to increase its reliability.

From this point of view, web-based question answering system has many attractive properties. However, it is just empty talk if we couldn't have effective ways to retrieve data from web. Here is when search engine comes to rescue, meaning that we could utilize search engines to do related documents or websites searching and collecting.

An observation from MIT researchers showed that the empirical distribution of user queries turns out to quantitatively obey Zipf's Law - a small fraction of question types account for a significant portion of all question instances. Many questions ask for the same type of information, differing only in the specific object questioned e.g. "What is the population of India?"," What is the population of the United States?", "What is the population of Australia?" So after the discussion above, we should take advantage of data redundancy, employs a combined approach: instead of using the Web directly to answer questions, we treat the Web as an auxiliary to validate candidate answers extracted from a primary, more authoritative, corpus.(maybe YAGO). This approach is already adapted by MultiText (another question answering system)

3.Scope

Building an intelligent and powerful QA system is not an easy task. Both MIT's START project and IBM's Watson take excellent researchers and scientists many years' efforts. To reduce the difficulty of implementation, we will choose a well-developed knowledge base instead of a large unstructured text database to be the prototype's "database", such as YAGO2, which extracts information from sources like Wikipedia, WordNet, etc.

In addition to the choice of information source, question type is also an important factor in system complexity. There are mainly two types of questions: factoid questions (or fact-based questions) and narrative questions (or subjective questions). The answer of a factoid question is usually a short answer with single phase [3] (i.e., "What is the calories of an apple?"), while narrative question usually requires rational thinking (i.e., "How do you think of the movie Gone with the Wind?").

4.Deliverables & Schedule

Deliverables:

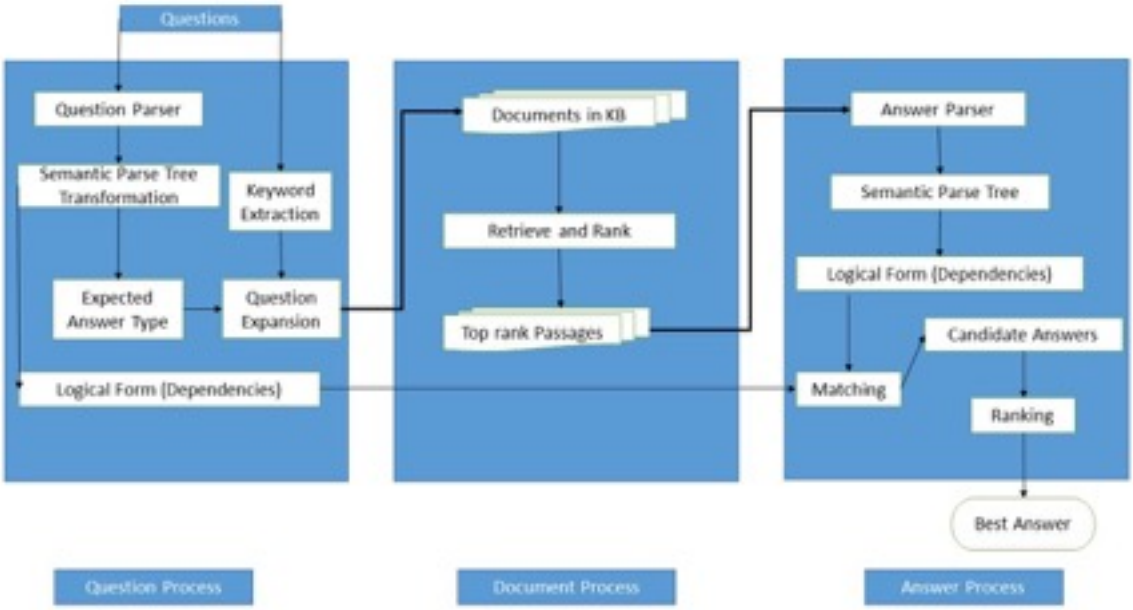
1. A brief study report about our understanding of a classic KB-QA system after reading related books and research papers.
2. An evaluation report of possible tools for prototype implementation.
3. A demo website accepting users' questions and returning parsing result for that question.
4. An interim report.
5. A demo website accepting users' questions and returning documents selected.
6. A demo website accepting users' questions and returning sentences selected.
7. A demo website accepting users' questions and returning sentences parsed.
8. A demo website accepting users' questions and returning all answers.
9. A demo website accepting users' questions and returning top 5 answers with confidence level.
10. A demo website including 20 test examples.
11. A final report including introduction of classic KB-QA system, tools selection, implementation details, further exploration on one or two problems detected, etc.
12. A website describing our final year project including project plan, interim report, final report and all the auxiliary supporting materials.

Schedule:

Index	General	Task	Date
1	Deliverables of Phase 2: · Preliminary implementation · Detailed interim report	Study report	01/11/2015
2		Evaluation report	15/11/2015
3		Demo website phase 1: return parsing result	10/01/2016
4		Interim report	17/01/2016

5	Deliverables of Phase 3: · Finalized tested implementation · Final report	Demo phase 2: return documents selected	24/01/2016
6		Demo phase 3: return sentences selected	7/02/2016
7		Demo phase 4: return sentences parsed	21/02/2016
8		Demo phase 5: return all answers	7/03/2016
9		Demo phase 6: return top 5 answers with confidence level	20/03/2016
10		Demo phase 7: 20 test examples	24/03/2016
11		Final report	17/04/2016
12		Complete website	17/04/2016
13	Final Presentation		18-22/01/2016
14	Project Exhibition		03/05/2016

5.Project Approach



The implementation of a KB-QA system prototype is composed of 3 main parts in answering a factoid question, question processing, document processing and answer processing.

Take the question “Who built the first question answering system?” On one hand, the parser shall return a parse tree like the follows (using the CMU parser):

```
(S who
  (S (VP built
       (NP the
         (ADJP first)
         question answering system))))
```

The expected answer type can be found through an Answer Type Hierarchy like WordNet. Also this semantic form of parse tree is transformed into logical form so that it can be used to map the potential answers’ logical forms to find the best match in the future step. On the other hand, NER (Name Entity Recognizer) extracts keywords from the question and expand or rewrite the question to create queries.

Then the IR engine retrieves documents from knowledge base with a ranking of relevance. And the system ranks the passages in each document according to the occurrences of keywords and selects the top hundreds ones.

Finally, the procedures in the answer processing part are similar to those in question processing except the matching between question’s logical form and candidate answers’ logical forms.

6.Risk

Risk Identification	Possibility	Impact	Mitigation
Server not capable handling computation on huge data volume	High	High	Ask for technical assistance from department
Synchronization between teammate's subtasks	Medium	High	Using subversion control system like GIT
Complexity of programming	High	Mediu m	Narrow down the scope of searching domain
Schedule deadline not met	High	Mediu m	Set meeting within groups to report progress of own tasks

References:

[1] Zadeh, L. A. (2006). From search engines to question answering systems—The problems of world knowledge, relevance, deduction and precisiation.Capturing Intelligence, 1, 163-210.

[2] Question answering. (2015, August 20). In Wikipedia, The Free Encyclopedia. Retrieved 13:20, September 3, 2015, from https://en.wikipedia.org/w/index.php?title=Question_answering&oldid=676923590

[3] Wang, M. (2006). A Survey of Answer Extraction Techniques in Factoid Question Answering. *Computational Linguistics*, *1*(1).

[4] Bollacker, K., Evans, C., Paritosh, P., Sturge, T., & Taylor, J. (2008, June). Freebase: a collaboratively created graph database for structuring human knowledge. In Proceedings of the 2008 ACM SIGMOD international conference on Management of data (pp.1247-1250). ACM.

[5] Suchanek, F. M., Kasneci, G., & Weikum, G. (2007, May). Yago: a core of semantic knowledge. In Proceedings of the 16th international conference on World Wide Web (pp. 697-706). ACM.

[6] Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., ... & Bizer, C. (2013). DBpedia-a large-scale, multilingual knowledge base extracted from wikipedia. Semantic Web Journal.

[7] Bollacker, K., Evans, C., Paritosh, P., Sturge, T., & Taylor, J. (2008, June). Freebase: a collaboratively created graph database for structuring human knowledge. In Proceedings of the 2008 ACM SIGMOD international conference on Management of data (pp. 1247-1250). ACM.