Peking University at the TREC-2005 Question and Answering Track

Jing He , Cheng Chen , Conglei Yao , Ping Yin , Yongjun Bao
School of Electronic Engineering and Computer Science
Peking University

Abstract

This paper describes the architecture and implementation of Tianwang QA system, which can work for the Main task and the Document Ranking task. The system is designed to extract candidate answer snippets from different pipelines, e.g. the high quality search engines' results, the frequently asked question (FAQ) set, and the well-structured web facts, etc..So the system need to process the Web documents, the FAQ corpus and the knowledge base (KB) from the structural web pages, besides analyzing the query, the TREC document retrieval and the answer merging. The external knowledge we made use of, i.e. FAQ and KB, are proved to be effective for our final results. We classify questions with SVM approaches, construct queries in Boolean way, retrieve and rank the passage with span model and extract answers using named entity technologies.

1. Introduction

Answering human beings' questions in exact words has long been studied from multiple research fields. From 1999, TREC held QA track to prompt it by raising different tasks, releasing the answer results and ranking the runs of each team. The tasks in 2005 includes main task, document ranking and relationship task.. Our group participate TREC QA track this year for the first time. Lack of experience as we are, we make good use of our ability in web crawling to seek for answer verification among external date sources. And we focus on the effect of information retrieval (IR) technologies, used in our Web search engines, on QA research. This paper is organized as follows: In Section 2 we give an overview of our system, including the architecture and the important components. In Section 3 we describe the components in detail and analyze the technology we use. In Section 4 we conclude the system and show the result of this evaluation.

2. Overwhelm of TianWang QA system

As mentioned above, our system, TianWang QA, is set up for the first time in this track.. However, we try many technologies and approaches in our system to deepen our research on the related fields. The system infrastructure are given below.

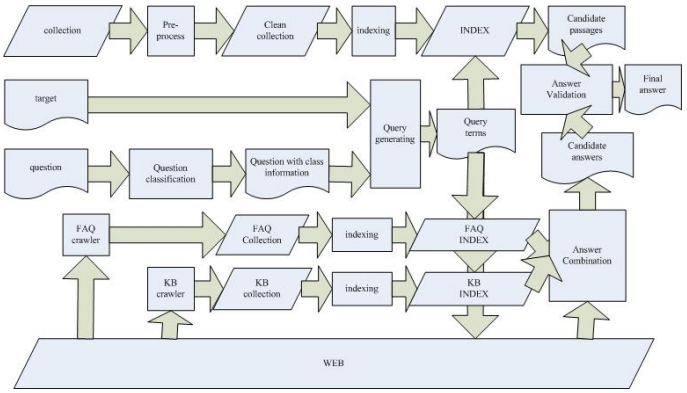

Figure 1. Big Picture of Tianwang QA system

First, the TREC document set is pre-processed and indexed. Second, the question terms are generated from the target and the question type. Third, the question terms are passed to the FAQ index interface, the KB interface and the web crawler. After the candidate answer snippets are returned from different source with ranking order, they are checked in the TREC document set by the answer validation part. And finally, the very matched answers are given.

3. Main System Components

Our system is a loose one. Every component is flexible that can be implemented with different approaches or algorithms. In this section, we describe every component in detail and analyze them.

3.1. Document Module

The purpose of Document module is to preprocess the collection. Most of the work is based on a generally used package Lucene. We parse the collection, remove some tags and stop words, stem (or not stem for different version) and construct the inverted files for the collection. The size of inverted file is more than 3 Gigabytes.

3.2. Query and Retrieval Module

This module is the base for the next four steps. In this part, the questions are classified. And the documents are retrieved for entity extraction component and Document Ranking task. We follow the two-layered question taxonomy [7], containing 6 coarse grained categories and 50 fine grained categories. We use SVM method with one against all strategy and bag of words feature to perform the classification [11]. Each question is labeled with only one category with maximum probability. The training dataset is provided by UIUC [7], which has 5,500 labeled questions. Each coarse grained category contains an on-overlapping set of fine grained categories. Since finer grained categories can benefit us in locating and verifying the plausible answers, we extend the question taxonomy with some categories to finer categories.

We construct Boolean queries from questions only and queries are constructed iteratively. The approach is similar to [9]. However, we use some linguistic knowledge as [4]. We analyze the POS of the question and extract the entities from it. We start the query with proper nouns and named entities. If the number of documents retrieved is in a predefined scope, the processing terminates. Otherwise, we should loose or contract the query. The strategies for loosing a query include drop some query term, replace the PHRASE query to AND query, add some synonymy or morphological extension and so on. The strategies for contraction include add terms, depending on their idf value. The processing continues until the number document retrieved satisfies a threshold. However, the documents retrieved from the Boolean query have no ranking. So we need to rank the retrieved documents. The most common approach is to calculate the cosine similarity. As [10] pointed out, span based approach is effective for QA, so we also consider distance between query words and get the documents ranking.

3.3. FAQ Corpus Module

As many previous work [2][3] pointed out, FAQ documents, which consist of a series of frequent asked questions and their authority answers, are important for QA. In order to get a FAQ collection, we tried two steps. First, we select some web sites, such as FAQ Archive, collecting FAQ documents in some domains; Second, we use Google to find some web pages containing some words such as ”FAQ””Frequent asked question”, etc. They are often the entry pages linked to FAQ web pages. Then we analyzed their anchor texts and link URL and to get the needed FAQ web pages. We finally have about 23,000 qualified web pages. We identify and classify the questions, build the Q-A pair and index the FAQ documents by Lemur.

If some FAQ documents matches the questions, the related QA pairs are located. We get the candidate answers based on combination score of question category similarity, the document relevance and QA pair relevance.

3.4. Knowledge Base Module

Enlightened by the idea of MIT’s Aranea [6][1][5], we choose high quality web content with little noises and regular organization as a kind of pre-built knowledge base. Our source data come from CIA factbook, 50states.com and the biography.com. The data extraction is based on pattern built manually, which is the reason why the scale of such kind of knowledge base is limited. However, [8] has verified that ten selected Web sources provides 47% answers for TREC-2001 QA track.

We design data model using the tri-tuple ¡object, property, value¿ to describe the entities in the structured or semi-structured web pages. Every entity, no matter how complex originally described, can be simplified into a series of property-value pairs. We expanded the index words of the properties in some scope using synonyms. As we consider the three parts of one tuple: 1) The object words, mainly named entity or proper nouns specifying a certain entity, rarely need expansion; 2) The value part, showing the facts about the entities such as number, date and so on, is not suitable to expand either. 3) The property part, derived from the entities’ description or attribution, limited in quantity and diverse in expression, is deserved to be expanded with synonyms or thesaurus.

Tuples are taken as documents when processing. The key words detected by GATE in the tuple elements are indexed, and the questions are converted into bags of terms. The indexed words come from each parts. In many cases, the question about certain entities should match property or object part. The index format is like: <term, tuple element, tuple id>.The tuple set is processed using Lucence. The result of a query is a set of tuple id-tuple element pair. The candidate answer selection strategy is: If two of the three tuple elements are matched, we extract another element. The extracted element should be in the entity category corresponding to question category.

3.5. Web Search Module

We choose Web as an information source for this task. We pick key words from the question and search them in Google. As the snippets are retrieved, we select them according to number of keywords matched threshold.

3.6. Answer Extraction and Validation

The answers may be extracted from document passages, FAQ or Search Engine snippets, but the approaches are similar. The entity types include GATE entities and some predefined entity type. We use WordNet and some Web resources to find list of entities and tag their type. We also write some regular expression to match some type of entities. So the extracted entities are from GATE, list or regular expression matching.

We assign scores to each entity extracted, and rank entities according to their scores. The score computation is similar to [7]. One problem is whether or not the entity's type matches the question category. In most cases, the matching is boolean. However, things are not always so simple. If the question category is NUM:date, the "full" dates are ranked above "year" dates. Conversely, the "year" dates are ranked above "full" dates for the question type NUM:date:year. Then entity score is based on the frequency of occurrences of a given entity within the passages. We use the occurrence frequency of an entity as its score. This score is as the second sort key, to impose a ranking on entities that are not distinguished by the first score component. Some entity normalization are needed in counting because same entity such as person name and date may be expressed differently. Then we merge entities from data sources and entities from knowledge base, and the entities from knowledge base are put ahead of all.

The entities, extracted from many information sources, may not be located in the AQUAINT data source. So we should filter these entities. Then the results filtered are the final results. For list questions, the first 5 entities are as answers; for factoid questions, the first entity is as the answer and for definition questions, the top 7 passage snippets containing any of the top 5 entities are as answers.

4. Conclusion and Future Work

In this paper, we describe why and how we design each of the components. The scores for factoid, list and other questions are 0.108, 0.053 and 0.025. So the system should be improved in multiple ways. The IR sub-system may be trained by the evaluation data of Document Ranking task this year. Question or Answer patterns may be used in new version. Also we may make use of some natural language processing technologies such as syntax and semantic analysis of sentence. KB and FAQ collection also should be expanded in size and be explored in a quantitative way.

Acknowledgment

This work described therein is supported in part by PRC Ministry of Education grant 20030001076 and by NSFC grant 60435020.

References

[1] Daniel Loreto Wesley Hildebrandt Matthew Bilotti Sue Felshin Aaron Fernandes Gregory Marton Boris Katz, Jimmy Lin and Federico Mora. Integrating web-based and corpus-based techniques for question answering. In *TREC2003*, 2003.

[2] R. Burke, K. Hammond, V. Kulyukin, S. Lytinen, N. Tomuro, and S. Schoenberg. Natural language processing in the faq finder system: Results and prospects, 1997.

[3] Robin D. Burke, Kristian J. Hammond, and Benjamin C. Young. Knowledge-based navigation of complex information spaces. In *AAAI/IAAI, Vol. 1*, pages 462–468, 1996.

[4] C. Cardie, V. Ng, D. Pierce, and C. Buckley. Examining the role of statistical and linguistic knowledge sources in a general-knowledge question-answering. In *Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP-2000)*, pages 180–187, 2000.

[5] Boris Katz Jimmy Lin. Question answering from the web using knowledge annotation and knowledge mining techniques. In *the twelfth international conference on Information and knowledge management*, 2003.

[6] Boris Katz Gregory Marton Jimmy Lin, Aaron Fernandes and Stefanie Tellex. Extracting answers from the web using knowledge annotation and knowledge mining techniques. In *TREC 2002*, 2002.

[7] X. Li and D. Roth. Learning question classifiers, 2002.

[8] Jimmy Lin. The web as a resource for question answering: Perspective and challenges. In *the third International Conference on Language Resources and Evaluation*, 2002.

[9] Horacio Saggion and Robert Gaizauskas. Exploring the performance of boolean retrieval strategies for open domain question answering. In *SIGIR 2004*, 2004.

[10] Jimmy Lin Aaron Fernandes Stefanie Tellex, Boris Katz and Gregory Marton. Quantitative evaluation of passage retrieval algorithms for question answering. In *SIGIR 2003*, 2003.

[11] D. Zhang and WS Lee. Question classification using support vector machines. In *SICIR 2003*, 2003.