TREC 2006 Q&A Factoid: TI Experience

Satish Balantrapu
Artificial Intelligence Division
TrulyIntelligent Technologies Pvt. Ltd.
satishrao.balantrapu@trulyintelligent.com

Monis Khan
Artificial Intelligence Division
TrulyIntelligent Technologies Pvt. Ltd.
monis@trulyintelligent.com

Ayyappa Nagubandi
Artificial Intelligence Division
TrulyIntelligent Technologies Pvt. Ltd.
ayyappa@trulyintelligent.com

Abstract:

This is the first attempt of Artificial Intelligence Division of TrulyIntelligent Technologies Pvt. Ltd. at the TREC2006 Question Answering track. As any Question Answering (QA) system typically involves Question Analysis, Document Retrieval, Answer Extraction and Answer Ranking and this being our first attempt and with certain constraints of time and resources, we developed some modules of our QA system in line with already existing approaches, for example document retrieval, pattern based answer extraction and web boosting. But there are areas where we tried our ideas and got quite encouraging results particularly, Question Analysis, Constraint based Answer Extraction and Answer Ranking which are described in this paper.

I. Question Answering System:

In this year's QA Track, we participated in the Main task. We put the most effort in Factoid questions and very little on List and Other questions. Our system basically has two QA cores, one is typically pattern based and the other one implements constraint based heuristics for retrieving the final answer. If pattern based core fails to retrieve answer then only constraint based core is applied. Both these cores share modules but differ from answer extraction part. As this being our first attempt and with the time constraint, we used different tools like GATE, MINIPAR, LUCENE, WORDNET, STANFORD TAGGER, LT CHUNKER and GOOGLE API of different Universities/Organizations for developing our QA system. We extracted the Target text, Question ID, Question type and Question from the given XML file. Then the question is mapped onto the "MAP_QUESTIONS" file which is an XML file containing the answer type of the question, the Lucene search pattern and Google search pattern. The keywords for Lucene search pattern and Google search pattern are obtained from the target text. If the target text is Named Entity, the target text itself forms the Lucene and Google search patterns. On the other hand, if the target text is an event, Minipar is run on the target text and extracts the nouns of the event which form the Lucene and Google search patterns. Thus, the system is integration of two different approaches one is pattern based (offline system as decision for answer extraction is taken before hand i.e. when pattern are prepared manually) and another is online system, where answer extraction is done dynamically using various scoring mechanisms. To see distinction between two, let us look at modules that a vanilla QA system consists:

1 .Question Analysis
2. Document Retrieval
3. Answer Extraction
4. Answer Ranking.

The online system differs in last two modules i.e. Answer extraction and ranking. We will describe each module separately and mark

differences for offline and online system and then we will picture our overall system.

1. Question Analysis

Analyzing the question involves two subtasks.

a. Keyword extraction
   Keywords extracted are used for document retrieval.
b. Answer Type identification
   Answer type is the category of the answer for the question asked.
   Answer type is extremely important information for answer extraction and ranking.

a. Keyword Extraction:

The keywords of the question are obtained by running Parts Of Speech (POS) tagger on the question. The Nouns, Verbs and Adjectives are considered as the keywords of the question.

Example:

Question:

What was the date of the 1999 All-Star Game?

Keywords**:

date, 1999, all-star, game

** aux verbs are ignored

Same keyword extraction technique is used for extracting keywords from question series target. Both keywords from question and question series target are used for document retrieval.

b. Answer Type identification:

Here, the question is analyzed to know of what it is asking i.e. whether the answer type of the question is a person, Organization, Location, Quantity, Date etc. (For Example, in general: When questions refer to Date, Who questions usually refer to Person, Where questions refer to Location etc.). Based on the question, the subject, predicate or object is extracted by running Parser and its semantic closeness to different entities like Person, Organization, and Location etc. (qTarget List) is obtained by using WordNet. The closest entity to which it matches is considered as the Target of the question.

Example:

Question:

Which country received the largest loan ever granted by the IMF?

Parser Output (Minipar):

fin C:whn:N country
country N:det:Det which
fin C:i:V receive
receive V:subj:N country
receive V:obj:N loan
loan N:det:Det the
loan N:mod:A large
loan N:vrel:V grant
grant V:mod-before:A ever
grant V:obj:N loan
grant V:by-subj:Prep by
by Prep:pcomp-n:N IMF
IMF N:det:Det the

In case of which questions directly followed by noun phrase, we pick subject i.e. country in this case as the answer type.

Semantic closeness to qTarget list:

Below is the semantic closeness scores of each NE type with country

Person	0.142857142857143
Location	0.333333333333333
Product	0.125
Number	0.142857142857143
Currency	0.111111111111111
Organization	0.25
Date	0.125
Occupation	0.125
Money	0.111111111111111
Title	0.125

Since location has got highest score it becomes

answer type.

Answer Type: Location

2. Document Retrieval:

An index is developed for the available documents using Lucene and the documents are stored in the database. The keywords of the question and questions series target (generated by Question Analysis module) are sent to the index and in turn obtain the relevant document numbers. These document numbers are used to retrieve actual documents from database.

3. Answer Extraction:

At this step offline (pattern based approach) and online system (constraint based heuristics approach) branch out, both adapt completely different approaches to extract potential answers, we will describe each separately.

Offline Answer Extraction:

After obtaining answer type a decision is made that is what pattern set should be applied for the concerned question. Then the patterns are applied on the retrieved documents and the environments are extracted which may contain potential answer. We pick only those NE's which are of answer type category. That NE list is our set of potential answers.

Example:

Question:

When was Shakespeare born?

Answer Type: date

Some sample Patterns for Date of Birth:

<NAME>\W*.{0,50}\Q(\E\W*.{0,8}-

<NAME>\W*.{0,50}was\W*.{0,15}born\W*.{0,30}on.{0,50}

<NAME>\W*.{0,50}born.{0,50}

<NAME>\W*.{0,50}was\W*.{0,15}born\W*.{0,50}and.{0,50}

<NAME>\W*.{0,50}was\W*.{0,15}born\W*.{0,30}in.{0,30}

\W*.{0,50}<NAME>'s\W*birthday\W*.{0,100}today's\W*birthdays\W*.{0,500}<NAME>

<NAME>\W*.{0,400}birth\W*\Q-\E\W*.{0,40}

NAME is the entity whose date of birth is to be extracted, which is, Shakespeare in this case.

All the above patterns are applied on the top retrieved text documents retrieved by Lucene Search engine and database for extracting the environments.

Sample Text Snippet:

ALBANY, N.Y. _ We`re making much of the turn of the century this year. With William Shakespeare`s birthday coming up on April 23, an Academy Award-winning film about him playing in the theaters, and Harold Bloom`s new book, ``Shakespeare: The Invention of the Human,`` on best-seller lists around the country, let`s pause for a moment to consider the turbulent times that Shakespeare faced in the waning days of the 16th century. When Shakespeare was born in 1564, Queen Elizabeth had been on the throne for six years, and she would remain the only monarch Shakespeare knew for most of his life. It was a heady time in England. In 1588, the small navy of that small island had defeated Spain, the greatest sea power on Earth. The religious conflicts that plagued the continent and even plagued Britain before Elizabeth`s reign were largely absent due to the queen`s shrewd tolerance of religious differences……

The pattern:

"Shakespeare\W*.{0,50}born.{0,50}" extracts the sentence "Shakespeare was born in 1564, Queen Elizabeth had been on the throne f" from

which the required Answer type (Date in this case) is taken as the possible answer.
With target text as the Google search pattern, the top ranked document's text is extracted from the web and again the above patterns are applied on these documents to extract the environments.
Now using web boosting strategy, the final right answer is declared.

Online answer extraction:

Before extracting the possible answers of the given question, potential sentences containing possible answers are extracted.
In sentence extraction step following sentence pruning conditions are applied:

1. sentences containing minimum number(two in our experiment) of keywords,
2. the required Answer Type NE should be in sentence and
3. most important keyword should be in sentence . This is done by semantic ranking keywords with Answer Type.

Now from the final list of sentences we extract all those NE s which match our Answer Type, these become our potential answers.
Finally we have list of potential answers, extracted from the documents.
From both offline and online systems we have list of potential answers, these answers are passed to answer ranking module to decide actual answer.

4. Answer Ranking

Like answer extraction module, we have different approaches for offline and online answer ranking, but one common technique that both systems share for answer ranking is web boosting. So we will first discuss web boosting, then we will go on describing the actual answer ranking.

Web boosting:

After extracting the potential answers, we go to web and apply same process of answer extraction described above to the documents retrieved from the web and get potential answers from the web.
Now we have,
a. potential answers from the documents
b. potential answers from the web.
Web boosting technique is used to find evidence for the right answer, based on the assumption 'the frequency of right answer will be more as compared to wrong ones'. So in web boosting step we add frequencies of 1 and 2 for all answers, thus getting the boosted list of potential answers.

Offline answer ranking:

Ranking in this case is quite simple. We web boost the potential answers extracted from documents and declare highest frequency answer as the correct answer.

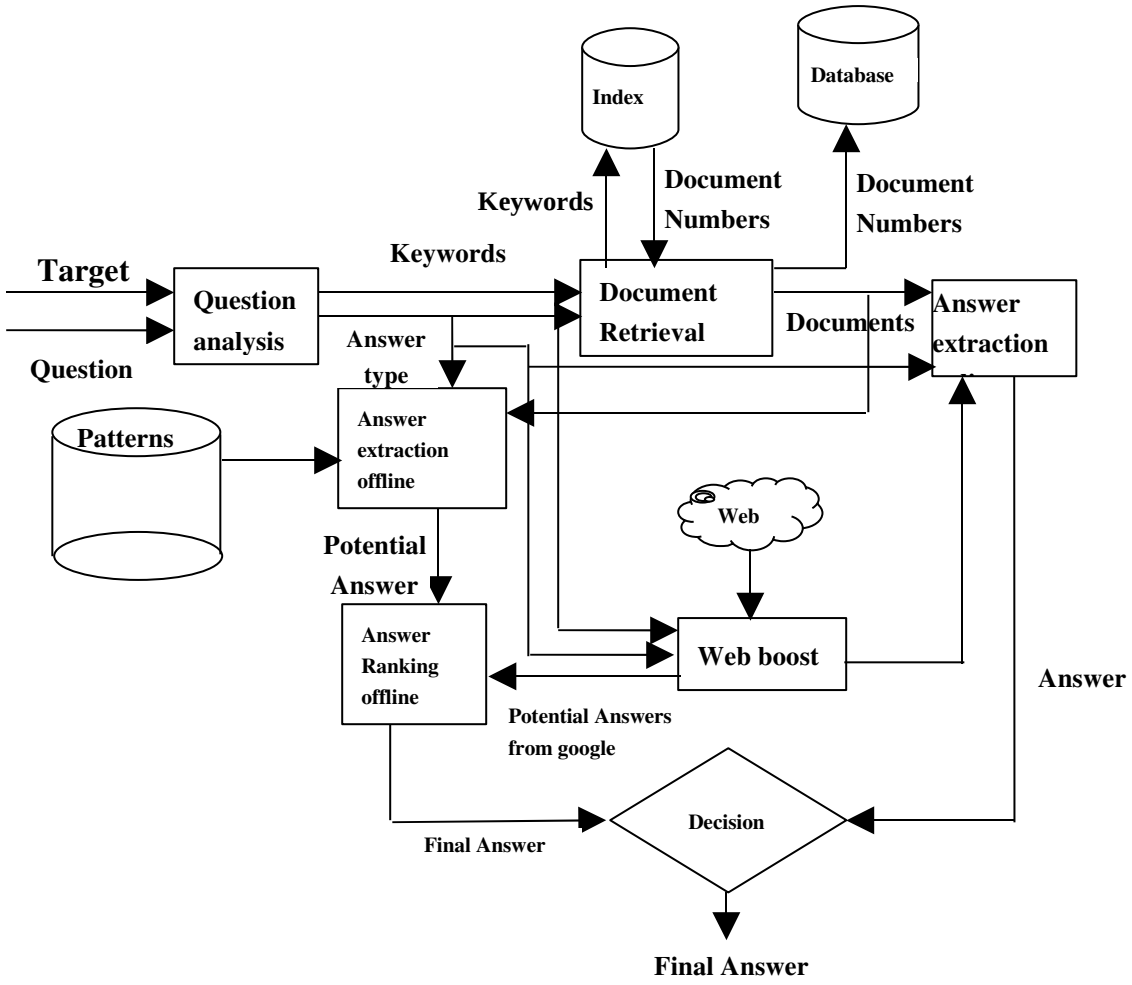
Online Answer Ranking:

The rank of potential answer is calculated based on three parameters

1 F, frequency of answer (web boosted frequency)
2 R, rank no. of index – assumption being that indexer will retrieve more relevant docs first.
3 K, The no. of keywords matched.

RANK OF POTENTIAL ANSWER = F + (1/R) + K
- all parameters are normalized to 1.
The highest ranked potential answer is declared as correct answer.
Now we have seen each module separately, let us look at the overall architecture of system.

Figure 1. Overall Architecture of TIQA2006



II. Overall Architecture:

As seen in Figure 1, once question analysis and document retrieval is done the offline answer extraction and ranking try to obtain answer, but if offline system fails to get answer we run online system. The reason for running offline system before online system is that, since offline system uses manually prepared patterns, if it is able to get answer there is a greater possibility that answer will be exact.

III. Results

Results		TIQA2006
Factoid Question	# Globally Right	77
	# Locally Correct	4
	# Inexact	27
	# Unsupported	10
	# Wrong	285
	Accuracy	0.191
List Question	Average F score	0.059
Other Question	Average F score	0.007

Table 1 Performance of TIQA2006

We submitted one run, TIQA2006 for the main task. The results are given in Table 1. The number of inexact answers is quite high, one main reason for this is the granularity of the NE identifier.The NE identifier we used is too general, it does not identify quantities like weight, length etc. Another interesting observation that we would like to share is that we got correct answer in first 10 potential answers ranked list more than 90% of time and hence answer ranking needs more refinement.

IV. Conclusions and Future work

As this being our first attempt, we concentrated mainly in answering Factoid questions and give little importance for List and Other questions. In the future, we have to come up with new approaches for answering List and Other questions along with improving the existing approach for Factoid questions. Another area where we want to improve is Named Entity identifier so that we can identify quantities, more detailed job titles etc. We would like to incorporate logical prover in our system for answer ranking.

V. References

Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Michael Junk, and Chin-Yew Lin. 2000. Question answering in Webclopedia. In *Proceedings of the Ninth Text RetrievalConference (TREC-9)*.

Ellen M. Voorhees and Dawn M. Tice. 2000. Overview of the TREC-9 question answering track. In *Proceedings of the Ninth Text REtrieval Conference (TREC-9)*.

Ellen M. Voorhees. 2001. Overview of the TREC 2001question answering track. In *Proceedings of the 2001 Text REtrieval Conference (TREC 200)*.

Soubbotin and S.M. Soubbotin. 2001. Patterns of potential answer expressions as clues to the right answers. In *Proceedings of the 2001 Text REtrieval Conference(TREC 2001)*.

Voorhees, E. and Buckland, L.P., Eds. (2003*). Proceedings of the Twelve Text Retrieval Conference TREC 2003*.

Voorhees, E. and Buckland, L.P., Eds. (2004*). Proceedings of the Thirteenth Text Retrieval Conference TREC 2004*.