Answer Selection and Confidence Estimation

Jinxi Xu, Ana Licuanan, Jonathan May, Scott Miller and Ralph Weischedel

BBN Technologies
50 Moulton Street
Cambridge, MA 02138
{jxu,alicuan,jmay,szmiller,weisched}@bbn.com

Abstract

We describe BBN's Question Answering work at TREC 2002. We focus on two issues: answer selection and confidence estimation. We found that some simple constraints on the candidate answers can improve a pure IR-based technique for answer selection. We also found that a few simple features derived from the question-answer pairs can be used for effective confidence estimation. Our results also confirmed earlier findings that the World-Wide Web is a very useful resource for answering TREC-style factoid questions.

1. Introduction

Answer selection and confidence estimation are two central issues in question answering (QA). The goal of answer selection is to choose from a pool of answer candidates the most likely answer for a question. The goal of confidence estimation is to compute $P(correct|Q, A)$, the probability of answer correctness given a question Q and an answer A. A QA system can use the confidence score to decide whether or not to show the user the answer. This is important because showing an incorrect answer has negative consequences: it not only burdens and but also may mislead the user with incorrect information.

For answer selection, we used a HMM-based IR system (Miller et al, 1999) to first select documents that are likely to contain answers to a question and then rank candidate answers based on the answer contexts using the same IR system. Then we used a few constraints to re-rank the candidates. Such constraints include whether a numerical answer quantifies the correct noun, whether the answer is of the correct location sub-type and whether the answer satisfies the verb arguments of the question.

For confidence estimation, direct estimation of $P(correct|Q,A)$ is impossible because it would require virtually unlimited training data. Instead, we computed the probability based on a few features that concern Q and A. A similar technique was used by Ittycheriah et al, 2001. The features were empirically selected with two criteria in mind: being able to predict answer correctness and having a small dimensionality. The features we used are the answer type of the question, the number of matched question words in the answer context and whether the answer satisfies the verb arguments of the question.

We also experimented with using the World Wide Web to supplement the TREC corpus for answer finding. Our results confirmed the positive findings reported in earlier studies (Dumais et al, 2002). We also found that the frequency of an answer in the returned Web pages is a strong predictor of answer correctness.

The next section describes the base QA system (i.e. without using the Web), including answer selection and confidence estimation. Section 3 describes how to use the Web to supplement the base system. Section 4 lays out our TREC 2002 QA results. Section 5 summaries this work. In our experiments, we used the TREC9&10 questions for training. To be consistent with the TREC 2002 QA track, only factoid questions in TREC9&10 were used for training.

2. The Base System

2.1 Answer Selection

Selecting the best answer for a question from the TREC corpus takes the following steps. First we used BBN's IR system (Miller et al, 1999) to select the top n documents from the TREC corpus. For the training questions, we set n=100, for efficiency considerations. For the test questions (i.e. TREC 2002 questions), we set n=300.

The question was then typed. A question classifier automatically assigned the question to one of the 30 types defined in our answer type taxonomy. (In some rare cases a question was assigned to more than one type. For convenience of discussion, we will assume one type per question). Similar to taxonomies used in other QA systems, ours includes common named entities such as persons, dates, locations, numbers, monetary amounts and so forth.

Then the candidate answers were ranked. The pool of candidates consists of all occurrences of all named entities in the top documents that match the answer type of the question. Named entities in the documents were recognized using BBN's IdentiFinder system (Bikel et al, 1999). The candidates were first ranked using BBN's IR system. To score a candidate, every text window that has the candidate at the center and has less than 60 words (for efficiency considerations) was scored against the question by the IR engine. The score for the candidate took that of the highest-scored window. The purpose of using multiple passages is to avoid choosing the optimal passage length, which is known to a tricky problem. A similar strategy was used by Kaszkiel for document retrieval (Kaszkiel & Zobel, 2001).

Then the candidates were re-ranked, by applying the following constraints:

1. If the question asks for a number, check whether the answer quantifies the same noun in the answer context as in the question.
2. If the question looks for a sub-type of locations (e.g. a country, state or city), check whether the answer is of that sub-type. We employed lists of countries, states and cities for this purpose. This constraint is useful because our taxonomy does not distinguish different kinds of locations.
3. Check if the answer satisfies the verb arguments of the question. For example, if the question is "Who killed X", a preferred candidate should be the subject of the verb "killed" and X should be the object of "killed" in the answer context. Verb arguments were extracted from parse trees of the question and the sentences in the corpus. We used BBN's SIFT parser (Miller et al, 2000) for verb argument extraction.

Candidates that satisfy the above constraints were ranked before those that do not. The highest ranked candidate was chosen as the answer for the question. The constraints produced a 2% absolute improvement on the training questions.

2.2 Confidence Estimation

We used three features to estimate $P(correct|Q,A)$. One feature is whether the answer satisfies the verb arguments of the question. This is a Boolean feature and we denote it *VS*. Using the training questions, we obtained $P(correct|VS$ is $true)=0.49$ and $P(correct|VS$ is $false)=0.23$, which clearly indicate *VS* is predictive of answer correctness.

The second feature is a pair of integers (m, n), where m is the number of content words in common between the question and the text window containing the answer, and n is the total number of content words in the question. The text window has a fixed length (30 words) and has the answer at the center. Like $P(correct|VS)$, $P(correct|m,n)$ were also computed from the training questions. Table 1 shows $P(correct|m, n)$ when n=4. As expected, $P(correct|m, n)$ monotonically increases with m when n is fixed.

	m=0	m=1	m=2	m=3	m=4
$P(correct|m,n=4)$	0.10	0.10	0.12	0.19	0.34

Table 1: $P(correc|m,n)$ when n=4, computed from training questions.

The third feature is T, the answer type of the question. Table 2 shows $P(correct|T)$ as computed from the training questions. As expected, some types of questions (e.g. Person) are more likely to result in a correct answer than other types of questions (e.g. Animal).

T	$P(correct|T)$
Location	0.24
Person	0.39
Date	0.26
Quantity	0.21
Cardinal Number	0.35
Organization	0.36
Animal	0.0
Misc	0.14

Table 2: $P(correct|T)$, computed from training questions. Types with too few training questions were put into the Misc category.

Since we do not have enough training data to directly estimate $P(correct|VS, m, n, T)$, we computed $P(correct|Q, A)$ by fitting $P(correct|VS)$, $P(correct|m,n)$ and $P(correct|T)$ in a mixture model:

$$P(correct|Q, A) \approx P(correct|VS, m, n, T) \approx P(correct|VS)\times 1/3 + P(correct|m,n)\times 1/3 + P(correct|T)\times 1/3$$

Since the parameters were pre-computed from the training questions, the computing of $P(correct|Q,A)$ for new *Q-A* pairs requires only a few table lookups.

3. Using the Web for Answer Finding

Some studies reported positive results using the World Wide Web to supplement the TREC corpora for question answering (Dumais et al, 2002). The idea is simple: the enormous amount of data on the Web makes it possible to use very strict, precision oriented search criteria that would be impractical to apply on the much smaller TREC corpora.

We experimented with using the Web for question answering too. In our experiments, we used the Web search engine Google because of its efficiency and coverage. Similar to (Dumais et al, 2002), we used two forms of queries, exact and non-exact. The former rewrites a question into a declarative sentence while the latter is a conjunction of all content words in the question. For efficiency considerations, we only looked for answers within the top 100 hits for each Web search. Furthermore, we confined to the short summaries returned by Google rather than using the whole Web pages in order to further cut the processing cost. The summaries were processed using BBN's IdentiFinder. The most frequent named entity that matches the answer type of the question was extracted as the answer.

For an answer from the Web, we computed confidence based on its frequency in the Google summaries. Since a lot of the data on the Web is of dubious quality, we also checked if the same answer was also produced by the base system from the TREC corpus. The confidence of a Web answer was computed as:

$P(correct|Q, A) \approx P(correct|F, INTREC)$

where F is the frequency of A in the Google summaries. *INTREC* is *true* if A was also returned by the base system from the TREC corpus. *INTREC* is *false* otherwise.

Figure 1 plots the probability of answer correctness as a function of F and *INTREC*. The figure shows that there is a strong correlation between F and answer correctness. The probability of answer correctness also strongly depends on the Boolean feature *INTREC*.

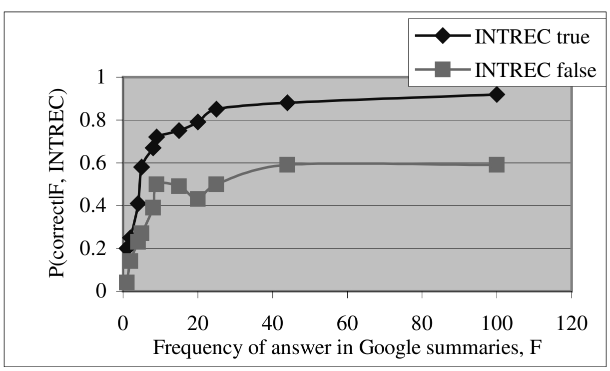

Figure 1: *P*(*correctness* | *F, INTREC*), computed from training questions

The question-answer pairs from the Web were merged with the ones produced by the base system. Since for the TREC2002 QA track each question can only have one answer, we chose the one with the higher confidence score if the Web answer and the answer from base system are different for a question. If the answers are the same for a question, the confidence score took the maximum of the score from base system and the score from the Web.

4. TREC2002 Results

We measured our TREC 2002 QA results using two metrics. The first is *unranked average precision*, which is the percentage of questions for which the answer is correct. The second metric is *ranked average precision*[1], which is the official metric adopted by TREC 2002 (Voorhees, 2003). Given a ranked list of question-answer pairs, the ranked average precision is computed by the following algorithm:

```
sum = 0;
num_correct = 0;
for rank i from 1 to N
{
  if answer to question at rank i is
correct
  {
    num_correct++;
  }
  sum += num_correct/i;
}
final_score = sum/N;
where N is the total number of
questions. For TREC2002, N=500.
```

Although the ranked average precision does not directly reflect how well the system computes confidence scores, it correlates strongly with the quality of confidence estimation because it rewards systems that place correct question-answer pairs ahead of incorrect ones.

One way to determine how well a confidence estimation method works is to compare its ranked average precision with that of a baseline that produces random confidence scores and that of a upper-bound that gives confidence 1 to all correct question-answer pairs and confidence 0 to all incorrect ones. It is easy to verify that the expected value of the ranked average precision for the baseline is the same as the unranked average precision.

Table 3 shows the ranked scores of the base system and the Web-supplemented system together with the baseline (unranked) and the upper-bound scores. The results show that our confidence estimation techniques work reasonably well: The ranked score is significantly better than the

[1] In TREC, this metric is called *confidence-weighted score*. We call it *ranked average precision* because it only takes into account the ranking order of the question-answer pairs.

baseline for both runs. This is especially true for the Web-supplemented run, where the difference between the ranked score and the upper-bound is very small. The Web-supplemented run is significantly better than the base system, confirming findings published in earlier studies (Dumais et al, 2002).

	Baseline score (unranked)	Ranked score	Upper-bound score
Base system	0.186	0.257	0.498
Web-supplemented	0.284	0.499	0.641

Table 3: Unranked, ranked and upper-bound scores for base and Web-supplemented runs on TREC2002

5. Conclusions

We described our question answering work for the TREC2002 QA track. In particular, we have explored two problems: answer selection and confidence estimation. We found that some simple constraints can improve a pure IR-based technique for answer selection. Our confidence estimation techniques used a few simple features such as answer type, verb argument satisfaction, the number of question words matched by the answer context and answer frequency in the retrieved Web pages. Performance scores on the TREC2002 test-bed show that our confidence estimation techniques work reasonably well. Our results also confirmed findings by other researchers that the Web is a useful resource for answering TREC-style factoid questions.

References

D. Bikel, R. Schwartz, R. Weischedel, "An Algorithm that Learns What's in a Name," Machine Learning, 1999.

S. Dumais, M. Banko, E. Brill, J. Lin and A. Ng. “Web Question Answering: Is More Always Better?”. In Proceedings of ACM SIGIR 2002.

A. Ittycheriah, M. Franz, W-J. Zhu and A. Ratnaparkhi, 2001. “IBM’s Statistical Question Answering System.” In TREC 9 Proceedings.

M. Kaszkiel and J. Zobel, "Effective Ranking with Arbitrary Passages", Journal of the American Society for Information Science (JASIS), Vol 52, No. 4, February 2001, pp 344-364.

D. Miller, T. Leek, and R. Schwartz, 1999. “A hidden markov model information retrieval system.” In Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1999.

S. Miller, H. Fox, L. Ramshaw, and R. Weischedel. 2000. A Novel Use of Statistical Parsing to Extract Information from Text. In *Proceedings of the North American Association for Computational Linguistics*.

E. Voorhees, 2003. “Overview of the TREC 2002 Question Answering Track”. In TREC 2002 Proceedings, to appear.