ICT Experiments in TREC-11 QA Main Task

Hongbo Xu, Hao Zhang, Shuo Bai
Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
hbxu@software.ict.ac.cn
http://www.ict.ac.cn/

Abstract

This is the second time we participate in the TREC-QA track. We put emphasis on candidate passage ranking and answer matching. As to named entity tagging, we applied the latest version of GATE and did some succeeding work aiming at our goal. This paper presents our methods in detail.

Keywords: TREC-QA, candidate passage ranking, answer matching

1. Introduction

We took part in the TREC-QA track for the second time this year. Of the main and list subtasks, we still undertook the main subtask. Three QA runs have been submitted for evaluation.

The document set for TREC-11 QA track has been changed to the new AQUAINT disk set released by AQUAINT Data Set Organization. The QA main task has several differences from previous years' tasks. Each question requires exactly one response, and the question set should be ordered by confidence in the response. The score assigned to each question will be 1 if the judgment is correct, and 0 otherwise. A measure that is an analogue to document retrieval's uninterpolated average precision will be used to score the run as a whole. The measure is computed as:

[sum for i=1 to 500 (#-correct-up-to-question-i/i)] / 500

Obviously, this measure will reward systems that correctly rank questions it answered correctly before questions it answered incorrectly.

Inspired by experiments on web track, we changed the weighting method of SMART to meet the need of TREC documents. We've made many experiments on candidate passages ranking to seek a better method and proper parameters. Another focus of our efforts is to improve the precision of answer extracting and matching. Aiming at the measure of TREC-11, we give priority to questions with types that were processed well in last year.

2. System Description

Our TREC-11 QA system is based on last year's. SMART[2], pairing sentences module, candidate passage ranking module and GATE[2] are used to retrieve relevant documents from data set and produce ranked named entities as candidate answers. Question analyzer analyses every question to identify the question type and keywords. Answer extracting and matching module matches the question type with the named entities. Answer outputting module outputs the most credible named entity and orders the question set by confidence in the answer. Figure 2.1 illustrates the whole architecture.

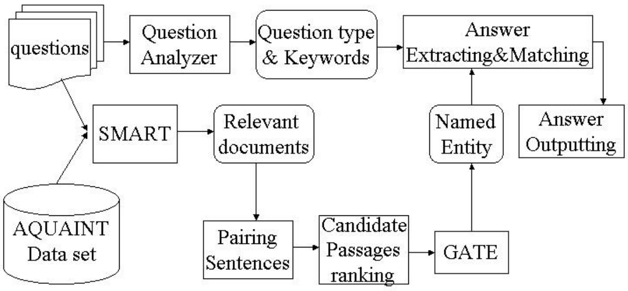

Figure 2.1: Architecture of the TREC-11 main QA System

3. Pairing sentences and Candidate passage ranking

In experiments on web track, we find that the weighting method of SMART is not so fit for TREC documents, so we modify its weighting module, taking *(log(tf)+1)*idf* instead of *tf*idf.* Then we use the revised SMART to retrieve 50 relevant documents for each question from the AQUAINT data set.

Subsequently, we parse these documents into sentences, and then assemble a candidate passage every two successive sentences that both have keywords in common with the question. The algorithm presented last year in section 4.4 of paper[2] is still used to rank the candidate passages for each question. Our new experiments show that step6 of the algorithm is of little help. So we devise a new method as an alternative. The score added to a candidate passage *P* is computed by:

$$\beta \ * count\_m / (count\_q + count\_k)$$

Where *count_m* is the number of matching keywords between the question and the candidate passage *P*, *count_q* the number of keywords in the question and *count_k* the number of keywords in *P*. β is an experiential parameter.

To lighten the burden of GATE in next process, for each question we only reserve the top 10 or 20 ranked candidate passages for named entity tagging.

4. Answer extracting, matching and outputting

We still use GATE as our Named Entity tagger. The latest version of GATE 2.0 (*released on March 15, 2002*) realizes the function to process a document set serially, which not only saves the time on loading modules when processing, but also allows us processing more candidate passages for one question. GATE 2.0 also optimizes the identification of type LOCATION, PERCENT, ORGANIZATION and PERSON. As to type NUMBER and MEASUREMENT, we still need to take some succeeding steps to assemble an integrated NUMBER or MEASUREMENT entity.

As in last year, we use a question analyzer to identify the question type and keywords of each question by two kinds of rules: keyword-based and template-based[2]. The main difference from last year is that we've made a consistency check on these rules to eliminate the collision when applying them.

Answer extracting, matching and outputting module compares the question type with the named entities in candidate passages and chooses the most credible named entity as final output.

To optimize for TREC-11 measure, we should order the question set by confidence in the answer. Without experiments supporting, we intuitively give priority to questions with types that were processed well by our system in last year.

5. Results and Analysis

We have submitted three runs for QA main task. In *ICTQA11a* and *ICTQA11b* we produce question answers from the top 10 candidate passages, the difference between them is that the candidate passages are ranked with different strategies. *ICTQA11b* and *ICTQA11c* have the same ranking strategies. But in *ICTQA11c*, question answers are derived from the top 20 candidate passages. Table 5.1 shows the evaluation results.

RunID	Confidence-weighted score	Wrong #	Unsupported #	Inexact #	Right #	Precision of recognizing no answer	Recall of recognizing no answer
ICTQA11a	0.091	445	9	4	42	10/58 =0.172	10/46 =0.217
ICTQA11b	0.084	440	7	6	47	9/69 =0.130	9/46 =0.196
ICTQA11c	0.088	435	8	9	48	9/47 =0.191	9/46 =0.196

Table 5.1 Statistics over all 500 questions of our runs in TREC-11

Our system does badly on most question types, except that the No Answer, DATE and LOCATION types are done a little better. Since only one response is allowed for each question, we think it's the simple answer matching strategy that does so much harm to the performance. Lack of time, many experiments aborted, so we had to give up trying our new answer matching methods. In addition, applying syntactic and semantic parsing technique should be a good approach to solve the problem on answer extracting and matching.

6. Conclusions

We've participated in the TREC-QA track for two times. By communicating with friends from China and abroad, we've learned much. We've also realized that there is a long way for us to go on QA research. But we are sure to do better in the future.

Acknowledgements

This research is supported by the national 973 fundamental research program under contact of G1998030413, the Institute Youth Fund under contact 20016280-9 and the Institute Youth Fund under contact 20026180-24. We give our thanks to all the people who have contributed to this research and development, in particular Bin Wang, Xueqi Cheng, Yitao Zhang, Sujian Li and Huaping Zhang.

[1] E. Voorhees, Overview of the TREC 2001 Question Answering Track, In *The Tenth Text REtrieval Conference (TREC 10)*, page 42, 2001.

[2] B. Wang, H. Xu, Z. Yang, Y. Liu, X. Cheng, D. Bu, S. Bai, TREC-10 Experiments at CAS-ICT: Filtering, Web and QA, In *The Tenth Text REtrieval Conference (TREC 10)*, page 109, 2001.

[3] M.M. Soubbotin, Patterns of Potential Answer Expressions as Clues to the Right Answers, In *The Tenth Text REtrieval Conference (TREC 10),* page 293, 2001.

[4] S. Alpha, P. Dixon, C. Liao, C. Yang, Oracle at TREC 10: Filtering and Question-Answering, In *The Tenth Text REtrieval Conference (TREC 10),* page 423, 2001.