Description of NTU System at TREC-10 QA Track

Chuan-Jie Lin and Hsin-Hsi Chen
Department of Computer Science and Information Engineering
National Taiwan University
Taipei, TAIWAN, R.O.C.
E-mail: cjlin@nlg2.csie.ntu.edu.tw; hh_chen@csie.ntu.edu.tw
Fax: +886-2-23628167

1. Introduction

In the past years, we attended the 250-bytes group. Our main strategy was to measure the similarity score (or the informative score) of each candidate sentence to the question sentence. The similarity score was computed by sums of weights of co-occurred question keywords.

To meet the requirement of shorter answering texts proposed in this year, we adapt our system, and experiment on a new strategy that is focused on named entities only. The similarity score is now measured in terms of the distances to the question keywords in the same document. The MRR score is 0.145. Section 2 will deal with our work in the main task.

We also attended the list task and the context task this year. In the list task, the algorithm is almost the same as the one in the main task except that we have to avoid duplicate answers and find the new answers at the same time. Positions of the candidates in the answering texts should be considered. We will talk about this in Section 3.

In the context task, how to keep the context, and what the answers of the previous questions can help are the main issues. In our strategy, the answers of the first question are kept when answering the subsequent questions, but the answers of the other ones (denoted by question *i*) are kept only if question *i* has a co-referential relationship to its previous one. Section 4 will describe this strategy in more detail.

2. Main Task

In the previous 250-bytes task, we measured the similarity of the question sentence and each sentence in the relevant documents, and reported the top 5 sentences with the highest scores and with the question focus words. In our experiment, the real answer sometimes lies in the sentence that is not so “similar” to the question. It becomes harder to extract text shorter than 50 bytes and containing the answer in this manner. Therefore, we experiment on another strategy, which is "candidate-focused" rather than "sentence-focused".

After reading a question, the system first decides its question type and keywords as usual. Now every named entity in the relevant documents becomes our answer candidate. For each candidate, we find out its distances to the question keywords in the same document, and sum up the reciprocals of these distances. One question keyword only contributes once, i.e., if a keyword occurs more than once, only the one nearest to the candidate contributes the score. Moreover, we assign higher weights to the keywords that are named entities. After scoring all the candidates, the highest top five are proposed, together with the texts surrounding the candidates within 50 bytes. The texts are extracted in such a way that the candidates can be placed in the middle.

In our experiment, we found that if there is a question keyword right preceding or following the candidate, it will dominate the score despite of the other question keywords. To solve this problem, we divide the distance by three, i.e., we consider three words as a unit to measure the distance. The scoring function is shown as follows:

$$score(x) = \sum_{t \in Q \cap D} \frac{1}{\left\lceil \min\left(\left|pos_D(t) - pos_D(x)\right|\right)/3 \right\rceil} \times weight(t) \qquad (1)$$

where x is an answer candidate, Q is the question sentence, D is the document currently examined, t is a term occurring in both Q and D, and $pos_D(t)$ is one of the occurrence positions of t in D.

The algorithms of deciding question type and extracting named entities are the same as those in last year, which was proposed in Lin and Chen (2000). If we cannot tell which question type a question belongs to, or the question type is not concerned with a named entity, we consider every kind of entities as candidates. To extract different answers as more as possible, we ignore those answering texts whose named entity answers have appeared in the previous answering texts.

Two runs were submitted this year. When question keywords were prepared in the first run *qntuam1*, variants of ordinary words (inflections of verbs, plural forms of nouns, etc.) and named entities (adjective forms of country names, abbreviations of organization names, *etc*.) are added into the keyword bag. Stems of keywords are also added with a lower weight. Note that no matter how many variants or stems of a keyword are matched in a document, only one of them contributes the score. We select the one that can contribute the highest score.

In the second run *qntuam2*, the synonyms and explanations provided by WordNet (Fellbaum *Ed.*, 1998) are also added, with lower weight to reduce the noise.

Moreover, if there are m words in an explanation text, and n words occur in the document, the matching score of this explanation is defined as $\sqrt{n/m} \times weight(e)$, where *weight*(*e*) is the weight of this explanation.

MRRs of these two runs are 0.145 and 0.101 under strict strategy, respectively.

3. List Task

List task is a new task beginning in this year. A question does not only ask for its information need but also a specified number of answers. Therefore, the system has to offer different answers to the specified number. An example is Question 1:

Question 1: Name 20 countries that produce coffee.

In this case, the system is asked to provide 20 names of different countries. Besides deciding which country produces coffee, the system also has to decide if the answer is duplicated, or if two answers are identical to each other.

The main algorithm to this task is almost the same as the main task. The only difference is that we extract the answering text in the manner that the candidates will be located at the beginning. By this way, if more than one answer appears in the same sentence, the previously proposed candidates will not appear again in the subsequent answering texts. The algorithm of the main task has already ignored the same answers (which is lexically identical), so we do not do other things to check answer identity.

Two runs were submitted as the same as those in the main task. Scores of the average accuracy are 0.18 and 0.14, respectively.

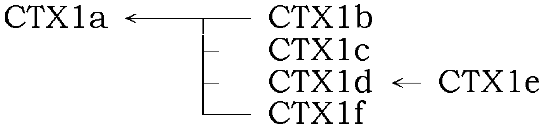
4. Context Task

There is another new task this year. A series of questions are submitted, which are somewhat relative to the previous questions. For example, in Question CTX1:

a. Which museum in Florence was damaged by a major bomb explosion in 1993?
b. On what day did this happen?
c. Which galleries were involved?
d. How many people were killed?
e. Where were these people located?
f. How much explosive was used?

Question CTX1a asks the name of the museum. Question CTX1b continues to ask the date of the event mentioned in Question CTX1a, so this question and its answer are important keys to Question CTX1b. Question CTX1c asks more details of

Question CTX1a, but irrelevant to Question CTX1b. So is Question CTX1d. But Question CTX1e refers to both Question CTX1a and CTX1d. We can draw a dependency graph of this series of questions as below:



If a question is dependent on one of its previous question, it is obvious that the information relative to this previous question is also important to the present question. Thus the system has to decide the question dependency.

We proposed a simple strategy to judge the dependency. Because the first question is the base question of this series, every subsequent question is dependent to the first one. After reading a question, if there is an anaphor or a definite noun phrase whose head noun also appears in the previous question, we postulate that this question is dependent on its previous question.

Next issue is that how we can use the dependency information in finding answers as well as its context information. After answering a single question, the system has located some answering candidates together with documents and segments of texts in which these candidates appear. Such information can be used to answer its subsequent dependent questions, as well as the keywords of the question itself. Note that context information can be transitive. In the above example, Question CTX1e consults the information that Question CTX1d itself owns, and Question CTX1d refers to, i.e., Question CTX1a.

In our experiment, we only consider the keywords and their weights as the context information. Furthermore, we assign lower weights to the keywords in the context information so that the importance of recent keywords cannot be underestimated. The answers to the previous question remain their weights because they are new information. The question type is decided by the present question.

The accompanying issue is that how confident an answer is included in the context information. This is because we may find the wrong answers in the preceding questions and those errors may be propagated to the subsequent questions. Moreover, do these five answers have the same weight? Or we trust the answers of the higher ranks than those of the lower ones, or only the top one is considered?

These issues are worthy of investigating, but not yet implemented in the experiment of this year. We assign weights to the previous answers according to the following equation:

$$weight(x) = weight\_NE(x) \times \sqrt{(6 - rank(x))/5} \times weight\_PreAns(x) \qquad (2)$$

where *weight_NE*(*x*) assigns higher weight if *x* is a named entity; *rank*(*x*) is the rank of *x*, and *weight_PreAns*(*x*) is a discount to the previous answers because they may be wrong. The square root part tries to assign higher weights to the higher-ranked answers.

Because only relevant documents to the first questions are provided, and we do not implement an IR system on TREC data, we cannot do a new search when answering the subsequent questions. Our solution is to search the same relevant set of the first question.

We submitted one run this year. Its main algorithm followed the first run of the main task.

There is still no formal evaluation of this task. The MRR of all 42 question of our result is 0.139. 4 of the first questions are correctly answered. Answers of at least one of the subsequent questions can also be found in each of these 4 series. Only one of the series is fully answered.

5. Discussion

Comparing the results of two runs of the main task and the two runs of the list task, we can find that synonyms and explanations introduce too much noise, so that the performance is worse. However, paraphrase is an important problem in question answering. Explanation provides only one of the paraphrases, thus we have to do more researches on paraphrases.

After investigation of the results of the list task, we found that there is a small bug when reporting answers. Although duplicate answers were neglected, equivalent answers were not. In other words, adjective forms of country names were regarded as different answers to their original names, which produced redundancy and lowered the performance.

In this year, the question types of many questions are not named entities. Many of them in the main task are "definition" questions. For example,

Question 896: Who was Galileo?
Question 897: What is an atom?

In our system, we only take named entities as answer candidates, so we cannot answer such type of questions, and the performance is rather worse than that of last year.

The same problem happened in the context task, too. Therefore, it is not obvious that our proposed model to the context task is good or bad. Further investigation and experiment are needed to verify this point.

References

Lin, C.J. and Chen, H.H., “Description of NTU System at TREC-9 QA Track,” *Proceedings of The Ninth Text REtrieval Conference* (*TREC-9*), 2000, pp. 389-406.

Fellbaum, C. *Ed*. (1998) *WordNet: An Electronic Lexical Database*.