LCC-DCU C-C Question Answering Task at NTCIR-5

Bin Wang[1], Gareth J.F. Jones[2]
[1]LCC group, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100080, China
[2]School of Computing, Dublin City University, Dublin 9, Ireland
wangbin@ict.ac.cn, Gareth.Jones@computing.dcu.ie

Abstract

This paper describes the work for our participation in the NTCIR-5 Chinese to Chinese Question Answering task. Our strategy is based on the "Retrieval plus Extraction" approach. We first retrieve relevant documents, then retrieve short passages from the above documents, and finally extract named entity answers from the most relevant passages. For question type identification, we use simple heuristic rules which can cover most questions. The Lemur toolkit with the OKAPI model is used for document retrieval. Results of our task submission are given and some preliminary conclusions drawn.

Keywords: *NTCIR, Question Answering, Information Retrieval, Information Extraction*

1 Introduction

LCC and DCU participated jointly in the Cross-Language Question Answering (CLQA) task at NTCIR-5 for the first time. We chose the Chinese-Chinese (C-C) Question Answering (QA) subtask as the first step this time, and hope to extend this in the future to the full English-Chinese QA task. Our method is based mainly on the information retrieval with information extraction ("IR+IE") approach. That is, we first retrieve documents and short passages that may contain correct answers, and then seek to identify answers through the application of information extraction techniques on the retrieved items.

The NTCIR-5 C-C QA task can be simply summarized as: given 200 factoid Chinese questions, retrieve the named entity answers from a collection of 901,446 news articles taken from UDN.COM spanning a period of two years. Both the questions and news articles are encoded with BIG5 (a Traditional Chinese encoding mode). Since we have little experience in Traditional Chinese processing, we transformed all the Chinese data into GBK codes and used GBK-based Simplified Chinese processing tools.

For question type analysis, we used some simple heuristic rules based on word segmentation and part-of-speech tagging. For document retrieval, we applied the Lemur [1] toolkit with the OKAPI model. For answer retrieval, we divided documents into smaller units, retrieved the most relevant units and extracted the named entity with the highest score.

We submitted two results, one is official, and the other is not. The difference is that the former is based on a real system, while the latter is a simulated run.

The remaining parts of this report are organized as follows: Section 2 describes our C-C system architecture; Section 3 introduces the preprocessing and post-processing

[1] See Lemur project website http://www.cs.cmu.edu/lemur/

components; Section 4 gives question type analysis; Section 5 and 6 respectively report document retrieval and answer retrieval processes. In Section 7, experimental results and some analysis are given. Finally, conclusions and future work are summarized in Section 8.

2 System Architecture

Our C-C QA system is illustrated in Fig. 1.

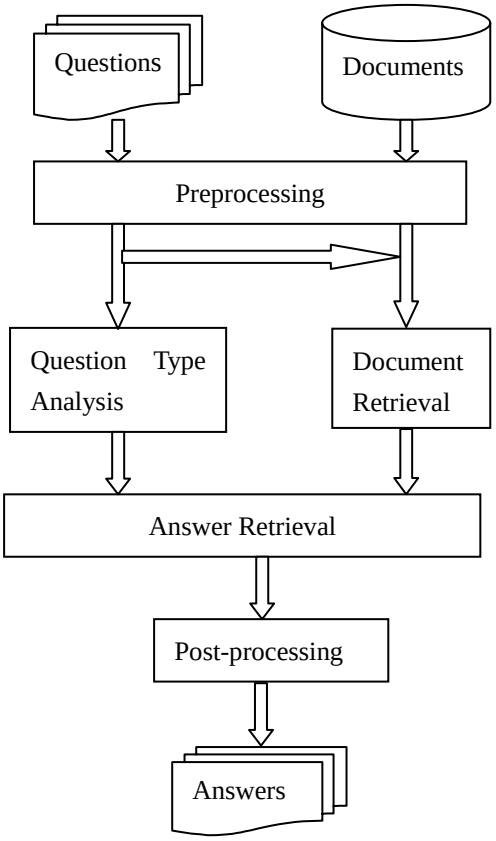

Fig. 1 C-C system architecture

The operation of the system proceeds as follows:

1) All the data are transformed to GBK code;

2) All the transformed data are segmented into words and tagged with part-of-speech;

3) Each question is analyzed to determine its type;

4) News articles are indexed and the top-*k* documents for each question retrieved;

5) Retrieved documents are divided into small units, the name entities in the units are identified, and the most relevant units for each question retrieved;

6) All the possible named entities in the retrieved units are ranked and the most likely answer extracted;

7) The answers are transformed back into BIG5 code.

Step 1 and 2 respectively correspond to the preprocessing module in Fig. 1. Step 5 and 6 correspond to the answer retrieval module.

3 Preprocessing and Post-processing

As we pointed out above, all the data in this task, including questions and documents are written in Traditional Chinese and encoded with BIG5. However, since we have little experience with Traditional Chinese processing, we simply transform all the data into GBK code, and then all the characters are transformed into Simplified format[2], so we can apply our Simplified Chinese processing techniques. Although our Simplified Chinese processing tools are independent of the encoding or character type themselves, they are trained based on Simplified Chinese corpora. Another problem is that Simplified Chinese and Traditional Chinese use different words, terms and grammars. Thus, our tools may not be very appropriate for this task, but we cannot easily develop new tools for Traditional Chinese.

The code transformation tool is called Textpro [3], which is free software and can transform between the two Chinese codes-BIG5 and GBK. In addition, it can also convert Simplified characters and Complex characters to each other within GBK code. As we know, some BIG5 characters do not have corresponding GB (an encoding mode usually used for Simplified Chinese) characters. GBK has all the characters in BIG5 and GB, so we used GBK as the target

[2] GBK contains both BIG5 and GB characters. But some pairs of BIG5 and GB characters share the same meaning.

[3] See http://www.fodian.net/tools/TextPro5.zip

code. After all the data have been transformed into GBK and Simplified characters, we segment them into Chinese words and meanwhile recognize the named entities. Here, ICTCLAS[1], a Chinese segmentation and part-of-speech tagging tool, which we developed ourselves, is used. ICTCLAS also combines a named entity identification module. Although ICTCLAS is trained based on Simplified Chinese corpora - news articles from Chinese People's Daily, we found that ICTCLAS can generate reasonable segmentation results and named entities for the transformed CLQA document set. A problem for the current C-C QA task is that ICTCLAS can only recognize a limited number of named entity types: Person, Location, Organization and Time. This is not sufficient for the C-C QA task, where we need nine types of named entities. We were not able to revise our ICTCLAS system for this task due to time constraints. In our work, we use some special nouns to predict the other types of named entities.

After preprocessing, we get segmented and POS tagged documents and questions. These form the input data for later processing.

Like preprocessing, post-processing transforms the answers back to BIG5 code and organizes the data into the submission format after the answers are extracted.

4 Question Type Analysis

For factoid questions, the question type can be defined as the named entity type to be returned as the answer. In the NTCIR-5 C-C QA task, there are nine types of named entities: ORGANIZATION, PERSON, LOCATION, ARTIFACT, DATE, TIME, MONEY, PERCENT and NUMEX. For convenience, we combine DATE, TIME, MONEY, PERCENT and NUMEX into a type called NUMBER (sometimes we use its original type). Thus in our work, question type analysis is to assign one of the above four labels to each question. A number of methods have been proposed for question type analysis, such as machine learning[2], identification based on chunking[3] or parsing[4]. In our work, we used some heuristic rules to identify the question type. The rules are as follows:

1) If there is an interrogative or no interrogative but special numeral (e.g. "几") in the question, then

1.1) If the interrogative or the special words obviously correspond to one question type (e.g., "谁" – PERSON, "哪里", "哪儿" – LOCATION-, "多少" - NUMBER, "几" – NUMBER), then return the correct type;

1.2）Else if there is a quantifier or a numeral followed by a quantifier following the interrogative and it can determine the question type (e.g., "哪位"-PERSON, "哪一位", "哪家"-ORGANIZATION), then return the type;

1.3) Else find the question "hotspot", which can indicate the answer point (e.g., the hotspot of question "哪个**作家**写了日出" is "**作家**"), here we simply use the closest noun (if more than one noun occurs continuously, we use the final noun) to the interrogative which is not proper (theoretically speaking, it should be the head noun of the closest noun phrase following the interrogative, and if no noun follows, it may be the noun closest to the interrogative, e.g. "写西游记的作家是哪个"). Identify the question type based on the semantic class of the noun. (e.g., "哪个作者/人/作家"-PERSON; "哪家公司"--ORGANIZATION)

2) Else if there is no interrogative or special words in the question, find the closet noun to the auxiliary verb such as "是", "为" or word "在" or other word which is at the last word position of the question.

In addition, as we pointed out, we don't distinguish different numeric types strictly. If we can't determine which numeric type a question belongs to, we just label it with "NUMBER".

Another problem is about the semantic class of a noun. We use TONGYICI CILIN [5], a Chinese synonym dictionary, to label a word's semantic class. CILIN has a hierarchical class structure which includes three levels of classes (see Fig. 2) - 12 first level classes (denoted by A, B, etc.), 84 second level classes (denoted by a, b, etc.) and 1428 third level classes (denoted by 01, 02, etc.). And for real use, there are still smaller classes under the third level classes (denoted by 01, 02, etc. which are located in the sub-tree of the third level classes and the leaves of the class tree).

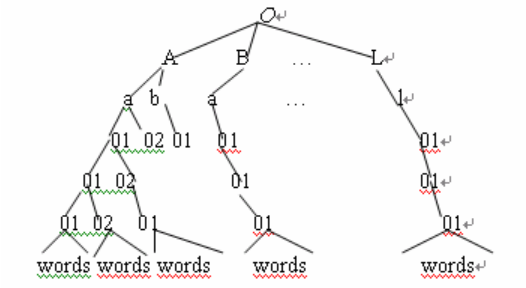

Fig. 2 CILIN's class structure.

Some useful classes for our task and their corresponding question types are listed below:

Table 1 CILIN's classes and their question types.

Class Code of CILIN	Question Type
A	PERSON
Ca	TIME
Cb	LOCATION
Di,Dm	ORGANIZATION

If the head word cannot be found in CILIN or labeled with another class code, we label it with ARTIFACT.

We found our simple heuristic rules can correctly identify the type of most questions (more than 75%). However, there is still work to do. First, we need a larger training set. Second, semantic labeling needs to be stricter.

5 Document Retrieval

We used the Lemur toolkit to perform document retrieval. Lemur is developed for language modeling Information Retrieval model by CMU&UMASS. However, it also includes traditional methods such as the vector-space model (VSM) and some probabilistic models such as OKAPI [6].

In our experiments, we tried a simple TFIDF VSM model and the BM25 OKAPI model to retrieve relevant documents. Although these two methods retrieve different results (at least the ranks of retrieved documents are different), the final answer results are actually very similar. So, finally we chose BM25 OKAPI model.

Some stop words such as interrogatives and other common stop words were eliminated from questions. We also tried to double the weight of words (e.g., proper nouns) that seem to be more important, but the results were not clearly improved.

6 Answer Retrieval

We have three steps to get the named entity answer. First, we divide each of the top-k (we set k=20) documents into bi-gram sentences (BS), that is, regarding every two contiguous sentences as a new retrieval unit, thus a document with *m* sentences would be divided to *m*-1 BS. Second, we retrieve the highest ranking BS according to some scoring scheme. Finally, we rank each named entity in the BS and extract the most appropriate named entity as the answer.

In the second step, the similarity score between a BS and a question is simply defined as the ratio of the length of the co-occurrence words in both of them to the length of the question. More words the BS and the question share, higher the similarity score is. We think that this scoring scheme can be regarded as a simple version of the VSM or OKAPI model. However, it can be more flexible if we want add more considerations, e.g., if we consider the

factor of order it is not difficult to modify the definition, e.g., if the question and the BS share two contiguous words, the score should be higher than two separate words, we can modify the definition easily by adding a suitable weight or adjusting the score in some other way.

In the final step, the closest named entity to the question words is selected as the answer, which has the correct type. If no named entity is identified, we simply chose the closest noun. A named entity that already exists in the question will not be selected.

7 Results and Analysis

We submitted two runs: one is official run, the other is not. The difference between them is that the former run is done by a real system, and the latter run is simulated by hand. In this manual run we constructed queries by hand, and used some Edit tools to find the answers by key word matching. The results are listed in Table 1.

Table 2 Submitted results.

Run ID	MRR	Accuracy
lcc-c-c-01	0.100	0.100
Lcc-c-c-u-02	0.235	0.235

According to the task guideline, for each question only one answer should be returned, thus the MRR is equal to Accuracy.

Due to time limitations, our QA system had many bugs (the system was not completely finished and not robust) at the time of submission. Some of them led directly to error results. After eliminating such bugs and extending some elements of the system, our performance improves to 0.29. Most ARTIFACT type questions return wrong results, because we have no taggers that can identify the name of a program, a film or a book now. To address problems such as this, we need a stronger named entity recognizer. Simplified Chinese tools for traditional Chinese also led to some errors. Other errors also occur in document retrieval, bi-sentence retrieval and answer extraction. The detailed analysis will be reported in a future paper.

8 Conclusions and Future Work

This is our first attempt at a Chinese QA task. We find our results interesting but we need to do more work.

Our future work plans include:

1) Analyze the reasons for errors and find the key problems;

2) Seek more suitable tools or adapt our current tools to Traditional Chinese processing;

3) Use enhanced language modeling IR method to improve the document retrieval performance;

4) Develop stronger tools for different named entity identification;

5) Explore more elaborate scoring schemes to improve the whole performance.

Acknowledgement

This work is supported by a China-Ireland Science and Technology Collaboration Research Fund award under Grant No. CI-2004-12 and China High Technology 973 project under Grant No. 2004CB318109. Thanks to Huaping Zhang for assistance with the Textpro and ICTCLAS tools and providing an unofficial run result.

References

[1]Huaping Zhang Hong-Kui Yu, De-Yi Xiong, Qun LIU; HMM-based Chinese Lexical Analyzer ICTCLAS, In Proceedings of the second SIGHAN Workshop affiliated with 41st Annual Meeting of the Association for Computational Linguistics, Sapporo Japan, 2003

[2]Dell Zhang, Wee Sun Lee, Question Classification using Support Vector Machines, in Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,

pp 26-32, Toronto, Canada, 2003.

[3]Seung-Hoon Na, In-Su Kang, Sang-Yool Lee, Jong-Hyeok Lee, Question Answering Approach Using a WordNet-based Answer Type Taxonomy, in Proceedings of the 11th Text REtrieval Conference (TREC2002), NIST, 2003.

[4]U. Hermjakob, Parsing and Question Classification for Question Answering, in Proceedings of the Workshop on Open-Domain Question Answering at ACL-2001, 2001.

[5]Mei, Jia-Ju, Yi-Ming Zhu, Yun-Qi Gao, Hong-Xiang Yin, Tongyici CiLin (Chinese Synonym Forest), Shanghai Press of Lexicon and Books, 1983.

[6]S.E.Robertson, S.Walker, S.Jones, M. Hancock-Beaulieu and M.Gatford, Okapi at TREC-3. In Proceedings of the Third Text REtrieval Conference (TREC-3). NIST, 1995.