Chinese Question Answering using the DLT System at NTCIR 2005

Richard F. E. Sutcliffe*
Natural Language Engineering and
Web Applications Group
Department of Computer Science
University of Essex, Wivenhoe Park
Colchester CO4 3SQ, UK
rsutcl@essex.ac.uk

Jia Xu Michael Mulcahy
Documents and Linguistic Technology Group
Department of Computer Science
and Information Systems
University of Limerick
Limerick, Ireland
Jia.Xu@ul.ie Michael.Mulcahy@ul.ie

Abstract

The DLT Group took part in the CLQA task for Chinese. With a basic system we achieved 14% overall.
Keywords: Question Answering, Chinese

1. Introduction

This article outlines the participation of the Documents and Linguistic Technology (DLT) Group in the Cross-Language Question Answering (CLQA) Task of the fifth NTCIR Workshop (NTCIR-5). We took part in the Chinese to Chinese (C-C) subtask. Having undertaken Question Answering (QA) tasks at TREC and CLEF, our strategy was to adapt our previous systems, adding any new components which were necessary for Chinese. The main objective was to become familiar with Chinese natural language processing and information retrieval.

The article is structured as follows. In Section 2 we outline the architecture of our system and in addition describe the changes to each component which were necessary for working with Chinese. Section 3 summarises our results. Finally, Section 4 gives our conclusions for the project and outlines future extensions to the system.

2. Architecture of DLT System

2.1 Outline

The architecture of the DLT System is standard and comprises query type identification, query analysis, retrieval query formulation, document retrieval, text file parsing, named entity recognition and answer entity selection. These are described in turn below.

2.2 Query Type Identification

A fundamental assumption of most QA systems is that any input query falls into one of a set of pre-defined categories. Our system is of this type and there are just thirteen types: `who`, `what_country`, `what_city`, `what_company`, `how_much`, `how_much_money`, `how_many3`, `when`, `when_month`, `when_month_date`, `when_year`, `when_taiwan_year` and `unknown`. Typically we use many more types (for example in CLEF 2005 we had 70) but due to limited time we only implemented those which we predicted would occur frequently in the test questions. As it turned out, our prediction was quite accurate as 59.5% of the queries fell into just three types (see Table 1), `who` (77), `what_city` (24) and `what_country` (18), total 119/200. Example queries of these types are shown in Table 2.

For English and French, the type of a question is identified using simple keyword combinations. The same approach was used for Chinese. To do this it was first necessary to tokenise the query. All text segmentation in the project was carried out using the Mandarin Tools tokeniser [1].

Generally this simple approach proved as effective in Chinese as it is in English. However, it is dependent on the accuracy of the segmentation. The same can of course be said about all Natural Language Processing tasks for Chinese. Categorisation performance can be discerned from Table 3 and is discussed later.

As in previous systems we employ the pseudo-type 'unknown' for queries which are not recognised as being of any specified type. The treatment of such queries is an interesting topic which has been investigated by others e.g. [2]. We comment on our own approach later.

2.3 Query Analysis

The objective of this stage is to analyse the query and extract from it likely phrases and keywords which can be used for document retrieval.

* On Sabbatical from University of Limerick.

The first step is tokenisation using the Manadrin Tools segmenter. Following this, pseudo Part-of-Speech tagging was carried out. This was done using word lists plus some simple heuristics in order to assign a likely POS to each token. A phrasal parser was then applied to the result, and any phrases recognised (some just being individual tokens) were extracted for further processing.

The phrases used for NTCIR were as follows:

- `number` – a number in either Arabic or Hanzi digits;
- `quote` – text between Chinese open and close quotes;
- `cap_nou_prep_det_seq` – an English title in ASCII, e.g. `University of Limerick`;
- `all_cap_wd` – an ASCII token all in capitals, e.g. `IBM`;
- `foreign_name_wd` – a token of any length which is on a list of words (one or more Hanzi) used for foreign names;
- `wd` – an ASCII word (presumed to be English) or a Chinese noun, verb, gerund, adjective or adverb as indicated by our Chinese word lists;
- `long_wd2`, `long_wd3`, `long_wd4` – any token comprising two, three or four Hanzi respectively which has not been recognised in a previous category.

2.4 Retrieval Query Formulation

Each phrase identified in the previous stage was assigned a weight based on its type. The weight reflects how important we consider the phrase to be in identifying documents containing the answer. The weights used were `number`: 2, `quote`: 3, `cap_nou_prep_det_seq`: 2, `all_cap_wd`: 3, `foreign_name_wd`: 3, `wd`: 1, `long_wd2`: 1, `long_wd3`: 2, `long_wd4`: 3

These weights reflect our intuition that quotes, words all in capitals, foreign name words and words of length four Hanzi are the most important in terms of defining what the query is about. After this are numbers, English titles and words of three Hanzi. The remaining classes of construct are the least important.

Phrases were then conjoined into a boolean retrieval query with the phrase of lowest weight first.

2.5 Document Retrieval

For the retrieval phase, the Lucene search engine was used [3]. We have implemented a framework that uses Lucene in a flexible way which allows us to index various languages by simply changing configuration parameters or by writing an appropriate tokeniser for the language. We have used the same indexing system for English text in TREC and CLEF and this was now adapted for Chinese text. The documents to be indexed are read in as Big5 and converted on the fly to Unicode UTF-8 for indexing. During retrieval, A query in Big5 is converted to Unicode before searching the index.

A key issue in the development of the system was how to tokenise the text. For example should individual Hanzi (i.e. Chinese idiographs) be indexed or is it better to segment the text linguistically and then index under these meaningful tokens, each possibly comprising several Hanzi? Also, what should be done about English words made up of ASCII characters?

There were two options available to us from the Lucene Sandbox (a Lucene repository of third party contributions): a CJK tokeniser and a Chinese tokeniser. The CJK tokeniser splits the Chinese text into overlapping pairs of Hanzi. The Chinese tokeniser splits the text into individual Hanzi. A third option would be the use of a linguistic stemmer. We decided to opt for the Chinese tokeniser, which resulted in each Hanzi being indexed separately independently of the word in which it happened to occur in any text. The intention was to use the exact phrase facility of the Lucene query language (which we also incorporated into our system) to retrieve just the appropriate passages, based on our linguistic segmentation of the query string.

Concerning ASCII characters, a contiguous sequence of these was tokenised with ASCII punctuation and white space characters being considered as separators. ASCII characters are a subset of the Big5 encoding and are used in the document collection to denote company names etc. which have not been translated into Chinese.

Following our practice in TREC and CLEF, the text collection was indexed sentence-by-sentence with each sentence being considered a separate 'document'. In order to do this we developed a sentence recogniser for Big5 Chinese based on some simple heuristics. For example, most Chinese sentences end with a Hanzi-sized (i.e. double-byte) punctuation character. These characters are easy to recognise since they can not be confused with the ASCII punctuation marks, unlike in English texts where the same same symbols ('.' etc) both terminate sentences and occur in other contexts such as computer filenames, after initials in proper names and so on.

In order to retrieve documents, the boolean query composed in the previous stage was submitted to Lucene using the standard Lucene query language which supports the usual functions such as exact phrases and boolean operators. If the query resulted in no documents being returned, it was simplified by removing the first term (i..e. the one with the lowest weight) and then re-submitted. This process was continued until some documents were returned or no further simplifications were possible. The idea behind

the query term weights is that the least important search terms are eliminated from the query first.

2.6 Text File Parsing

Text file parsing simply extracts the text of a 'document' (i.e. sentence) from its XML tags.

2.7 Named Entity Recognition

A fundamental assumption of our systems so far is that corresponding to a particular query type is a set of one or more Named Entity (NE) types. So for example if the query type is `who` then the expected NE type is `proper_name`. The Chinese system uses 13 NE types: `proper_name`, `country`, `city`, `province`, `year`, `taiwan_year`, `date`, `month`, `full_date`, `sum_of_money`, `num`, `num_unit`, and `title`. These are recognised using a mixture of lists and grammars defined over tokens. During NE recognition, all instances of NEs of the appropriate types are identified in each candidate document after first segmenting it. At the same time, any key phrases from the Query Analysis phrase are also identified. Based on an analysis of the training queries, we decided to consider all queries of type `unknown` as if they were `who`, thus using the NE type `proper_name`.

2.8 Answer Entity Selection

The final stage of QA is the identification of a particular NE within a document to return as the answer to the question. This is done by scoring each NE instance using a measure which incorporates the number of co-occurring key phrases, their assigned weights and their distance from the NE. Specifically, the distance between a candidate NE and a key phrase is measured in words, e.g. if the phrase is adjacent to the NE its distance is 1, if one word separates them it is 2 and so on. the reciprocal of this distance is taken and this is multiplied by the weight assigned to the phrase. The sum of all such values is taken to provide an intermediate score for the NE. The final score is this intermediate score multiplied by the Lucene score assigned to the containing document. Following this process, the highest scoring NE is returned.

3. Runs and Results

We submitted just one run of our system on the 200 test queries. The results are shown in Table 3.

For the performance of our system in query classification, refer to the first two columns of the table. 177 of the 200 queries were categorised correctly, i.e. 88.50%. This compares with 82.00% on factoids at CLEF 2005 (French queries) [4] and 89.13% on factoids at TREC 2004 (English queries) [5]. We can conclude from this that a simple keyword-based classification is adequate for all three languages and in particular that any inaccuracy of Chinese segmentation does not significantly affect the performance relative to the other languages, as far as we can judge without normalising the question complexity across the three QA tasks.

Concerning unknown queries, 38 (i.e. 19.00%) were correctly placed in this category. What this means is that 81.00% of queries fall within the designed scope of our QA system despite the relatively small number of question types we used.

Turning to question answering performance, this is shown in the remaining columns of Table 3. While the ineXact (X) measure is not used at NTCIR we have added this column by our own analysis – officially all these are Wrong (W). The total number of right answers (R) returned by our system was 28 out of 200, 27 following correct query type classification and 1 despite incorrect classification. This amounts to a Strict performance of 14.00% overall and compares with 39/230 factoids i.e. 16.96% at TREC 2004 (Run 2, English monolingual) and 30/150 factoids i.e. 20.00% at CLEF 2005 (Run 1, French-English cross-lingual). If we include the 13 queries we have assigned X (11 following correct classification and 2 following incorrect classification) we arrive at a Lenient measure of 41 out of 200 'right', i.e. 20.50%.

The vast majority of right answers are produced following a correct classification of query type. As an intermediate strategy in developing the system, we considered all `unknown` queries to be of type `who`, because this was the most frequent category in the training set. In the case of one query incorrectly assigned the category `unknown`, it actually was a `who` question which the system could answer – see Table 3, column 'R' under Incorrect Classification, row 'unknown'.

4. Conclusions

As stated at the start, our major objective was to gain some familiarity with Chinese NLP. This has clearly been achieved since we were able to develop a working system. What is more, most of the components were essentially the same as in our other systems. From this we conclude the Chinese NLP is not fundamentally different from English or French NLP (the other QA languages we have worked on).

Regarding our performance of 14.00% overall this compares favourably with our figures of 16.96% at TREC 2004 and 20.00% at CLEF 2005. These are well below the figures achieved by the best groups of course, but given our background and resources they show that we are making steady progress in developing our QA systems.

A fundamental design decision was to index the document collection using single Hanzi, independent

of their linguistic segmentation. This appears to have been wise as it gives us the flexibility to consider linguistic tokens via exact Lucene searches at retrieval time or to ignore these as appropriate. In this project we took the former course.

Due to shortage of time, we used far fewer query types than in our other systems – just 14 rather than around 70. However, it is interesting that this made very little difference to the results since most queries belong to only a few distinct categories. This can be seen from Table 1: 77 fall into category `who` alone, with 24 in `what_city`. So a system based on just one category alone could handle 77/200 or 38.50% of all the queries while one based on two could handle 77+24/200 or 50.50% of them. This result strongly suggests we should concentrate on 'deep' performance within a few categories rather than 'shallow' performance across many categories.

References

[1] http://www.mandarintools.com/

[2] C.L.A. Clarke, G.V. Cormack, G. Kemkes, M. Laszlo, T.R. Lynam, E.L. Terra, P.L. Tilker. Statistical Selection of Exact Answers. NIST Special Publication 500-251: The Eleventh Text REtrieval Conference (TREC 2002), National Institute of Standards and Technology, Washington, 2003, 823-831.

[3] http://lucene.apache.org/java/docs/

[4] R.F.E. Sutcliffe, I. Gabbay, M. Mulcahy, A. O'Gorman. Cross-Language French-English Question Answering using the DLT System at CLEF 2004. Proceedings of the Cross Language Evaluation Forum, CLEF 2004, Bath, UK, 16-17 September, 2004, 305-309.

[5] R.F.E Sutcliffe, I. Gabbay, K. White, A. O'Gorman, M. Mulcahy. Question Answering using the DLT System at TREC 2004. NIST Special Publication 500-261: The Thirteenth Text REtrieval Conference (TREC 2004), National Institute of Standards and Technology, Washington, 2005.

Query Type	Number
who	77
what_country	18
what_city	24
what_company	6
how_much	11
how_much_money	4
how_many3	1
when	4
when_month	1
when_month_date	3
when_year	8
when_taiwan_year	2
unknown	41
Subtotal	**200**

Table 1: Frequency of Query Types in the Test Collection

Question Type	Example Question	Translation
who	CLQA1-ZH-T1009-00 請問誰發明了大易輸入法？	Who invented the Dayi input method?
what_city	CLQA1-ZH-T1021-00 前南聯總統米羅塞維奇受審的聯合國戰犯法庭位於何處？	Where did the UN War Crime Tribunal which tried the former president of Yugoslavia, Milosevic, take place?
what_country	CLQA1-ZH-T1105-00 請問全球規模最大的德國漢諾威CeBIT電腦展中，2001年的最大參展國家是？	What is the biggest participating country in Cebit, world's largest computer expo, in 2001?
when	CLQA1-ZH-T1183-00 中華民國與美國是何時斷交發生？	When did the United States break diplomatic relationship with the Republic of China?
when_year	CLQA1-ZH-T1036-00 請問歐洲聯盟決議於何年開始統一使用歐元？	In which year did the European Union start to use Euro?

Table 2: Sample Queries in the Test Collection

Query Type	Class.		Correct Classification				Incorrect Classification			
	C	NC	R	X	U	W	R	X	U	W
who	67	7	16	5	1	45	0	2	0	5
what_country	13	3	3	0	0	10	0	0	0	1
what_city	23	1	3	0	0	20	0	0	0	3
what_company	2	1	0	0	0	2	0	0	0	1
how_much	11	0	0	0	0	11	0	0	0	0
how_much_money	4	0	0	0	0	4	0	0	0	0
how_many3	1	0	0	0	0	1	0	0	0	0
when	4	0	1	0	0	3	0	0	0	0
when_month	1	0	0	0	0	1	0	0	0	0
when_month_date	3	0	0	2	0	1	0	0	0	3
when_year	8	3	4	0	0	4	0	0	0	0
when_taiwan_year	2	0	0	0	0	2	0	0	0	0
unknown	38	8	0	4	0	34	1	0	1	6
Subtotal	177	23	27	11	1	138	1	2	1	19

Table 3: Results