Cross-Language French-English Question Answering using the DLT System at CLEF 2003

Richard F. E. Sutcliffe
Igal Gabbay
Aoife O’Gorman

Documents and Linguistic Technology Group
Department of Computer Science
and Information Systems
University of Limerick
Limerick, Ireland

+353 61 202706 Tel
+353 61 202734 Fax
Richard.Sutcliffe@ul.ie Igal.Gabbay@ul.ie
9935029@student.ul.ie Email
www.csis.ul.ie/staff/richard.sutcliffe URL

1. Introduction

This article outlines the participation of the Documents and Linguistic Technology (DLT) Group in the Cross Language French-English Question Answering Task of the Cross Language Evaluation Forum (CLEF). Our aim was to make an initial study of cross language question answering (QA) by adapting the system built for monolingual English QA for the Text REtrieval Conference (TREC) in 2002 (Sutcliffe, 2003). Firstly, therefore, we outline the architecture of the TREC system which formed the basis of the work reported on here. Secondly, the many changes made to allow cross-language QA are described. Thirdly, the runs performed are presented together with the results we obtained. Finally, conclusions are drawn based on our findings.

3. Architecture of the TREC 2002 DLT System

3.1 Outline

In this section we summarise the structure of the DLT system used at TREC which formed the starting point for CLEF. Changes subsequently made are documented in the next section. Overall flow of control was as follows. Firstly, we identified the query type and hence decided upon the related named entities for which we would be searching. Secondly, we parsed the 50 TOPDOCS text files, dividing them into textual units using the markup. These files were supplied by TREC and were generated by their own retrieval system in response to the input query. Thirdly, we searched for instances of the named entities in the textual units and marked any which were found. Fourthly, we identified the winning instance using one of two possible strategies: highest-scoring or most-frequent. These stages are now dealt with in more detail.

3.2 Query Type Identification

We studied questions of each type and developed simple keyword-based heuristics to recognise them. This crude approach was suprisingly effective. In TREC 2002 425 of the 500 queries were correctly classified.

3.3 Text File Parsing

Each document within the TOPDOCS file was divided into a series of segments corresponding to a short passage of text. First, text within a HEADLINE tag was extracted. Second, text within a TEXT tag was extracted and divided up into separate Ps. Finally a P was divided wherever three contiguous blanks were found. This last stage was to approximate sentence recognition. the resulting textual units were used in subsequent processing.

Question Type	Example Question	Google Translation
What_city	172 Quelle est la capitale de la Tchétchénie?	Which is the capital of Tchétchénie?
what_state	57 Quel est l' état indien qui a le plus grand nombre d' habitants?	Which is the Indian state which has the greatest number of inhabitants?
what_country	169 Dans quel pays le Mont Kilimanjaro se trouve-t-il?	In which country the Kilimanjaro Mount is it?
where	45 Où se trouve l' écosystème artificiel appelé "Biosphère 2"?	Where is the artificial ecosystem called "Biosphere 2"?
how_many3	76 Combien de salles de classe a-t-on construites en 1976 dans le district scolaire de Wilsona?	How many classrooms did one build in 1976 in the school district of Wilsona?
distance	164 Quelle est la longueur de la côte de la baie de Santa Monica?	Which is the length of the coast of bay of Santa Monica?
who	148 Qui est l' ambassadeur des États-Unis en Suisse?	Who is the ambassador of the United States in Switzerland?
when	178 Quand Shapour Bakhtiar est-il mort?	When Shapour Bakhtiar did he die?
unknown	160 Donner le nom d' un philosophe allemand.	To give the name of a German philosopher.

Table 1: Some of the Question Types used in the DLT system. The second column shows a sample question for each type. The translations resulting from submission to Google are listed in the third column

3.4 Named Entity Recognition

The type of question identified in the first step determined the type of named entity or entities to be searched for, as is standard practice in QA systems. Each segment identified as above was therefore inspected and all instances of appropriate named entities were tagged.

3.5 Answer Entity Selection

Two methods were used: highest_scoring and most_frequent. In the first, we returned the named entity occurring in a textual unit which matched the keywords in the query best, chosen from any of the 50 TOPDOCS documents. In the second, we returned the named entity which most frequently occurred in the vicinity of query keywords, observed across all occurrences of the entity in the 50 TOPDOCS documents. Both strategies were unsophisticated but sometimes one or other of them can perform well on a particular query type.

In the next section we explain how the above architecture was adapted for CLEF.

4. Architecture of the CLEF 2003 DLT System

4.1 Outline

The CLEF system had many similarities with the TREC one but also differed in a number of important respects. Flow of control was as follows. Firstly, the type of the French input query was identified along with the appropriate named entities. Secondly, the query was translated and processed in order to produce a search expression for retrieval. Thirdly, the search expression was submitted to a text retrieval engine yielding a set of documents. Fourthly, appropriate named entities were recognised in these. Fifthly, the winning named entity was identified.

4.2 Query Type Identification

As the input query was in French, our existing type identifier had to be re-written. This was undertaken by the third author as part of a separate project in which a French monolingual version of our TREC system was developed (O'Gorman, 2003). The first step was removal of diacriticals from the query, replacing each by the same letter minus the diacritical (e.g. 'á' becomes 'a' etc.). After this the query was converted to lower-case.

Finally, simple keyword combinations were used to identify the query type. 19 query types were used and some of these can be seen in Table 1 together with an example of each.

4.3 Query Translation and Re-Formulation

The next stage was translation of the French query into English. This was accomplished by submitting it to Google with original capitalisation and diacriticals (Google Translation, 2003). The result was then tokenised and stopwords were removed from any material not in double quotes.

4.4 Text Retrieval

Unlike in TREC, retrieval at CLEF was accomplished by indexing the documents ourselves. The document collection comprised 113,005 LA Times articles from 1994, 425MB in all. The search engine used was DTSearch (2000). Documents were indexed using the file segmentation option working with ‘<DOC>’ tags. The engine supports a number of search operators including distance between words (e.g. Word1 w/1 Word2 meaning Word1 must occur within 1 word of Word2) and boolean operators AND and OR. The next step therefore was to produce a search query on the basis of the terms identified in the previous step as follows:

- A 'w/1' connector was inserted between two capitalised words. It specifies that the second word must occur within one word from the first (in either direction). This feature was helpful because French to English translation often reverses the order of proper nouns.
- Double quotes were removed and the string of text within the double quotes was retained, but only if the first word after the double quote was untranslated. This was verified by checking the membership of the word in the original query list in French. Untranslated quotations were searched for exactly.
- An AND connector was inserted between all other terms.
- If the query included the atom 'did', it was removed. If the rest of the query contained a verb which was among the list of verbs common to TREC questions, the verb was replaced with its past tense form. For example, ‘die’ in questions like ‘How did X die?’ was replaced with ‘died’.

The query was then submitted to the engine and the first *n* DOCs returned were used in subsequent processing. In most cases *n* was 10 while in one experiment it was 20. Before the named entity recognition stage, each DOC was subdivided into separate P elements where present. These were then further subdivided wherever three contiguous blanks were found – a process which approximates to sentence recognition.

4.5 Named Entity Recognition

The set of English named entities used and the methods for recognising them in CLEF were very similar to TREC (Sutcliffe, 2003). Only two were added, general_name and planet. Type general_name recognises any sequence of up to five capitalised words interspersed by optional prepositions. It was inspired by Clarke et al. (2003) and is used in cases where the question type can not be determined. Planet uses a simple list of the eight planets.

4.6 Answer Entity Selection

As previously, two forms of answer selection were used, highest-scoring and most-frequent. In the highest-scoring method the named-entity instance is selected which occurs in the vicinity of the maximum number of keywords taken from the translated query, across all document passages. In the most-frequent strategy the named-entity occurring most frequently overall across all passages is chosen.

5. Runs and Results

5.1 Two Experiments

We submitted two runs. These only differed in respect of the answer selection strategy. The first run used the highest-scoring strategy working with the best 10 documents returned by the retrieval system. The second run adopted the most-frequent approach, also using the best 10 documents. The query classification module was the same for both runs.

Query Type	Classif.		Correct Classification								Incorrect Classification							
			Run 1				Run 2				Ru n 1				Run 1			
	C	NC	R	X	U	W	R	X	U	W	R	X	U	W	R	X	U	W
what_city	9	0	5	0	0	4	5	0	0	4	0	0	0	0	0	0	0	0
what_state	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
what_country	2	0	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0
what_continent	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
where	21	0	2	0	1	18	2	1	0	18	0	0	0	0	0	0	0	0
how_many3	17	1	0	0	0	17	0	0	0	17	0	0	0	1	0	0	0	1
distance	2	0	2	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0
speed	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
temp	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
population	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
who	34	7	7	0	0	27	7	0	0	27	0	0	0	7	0	0	0	7
when	16	0	1	1	0	14	1	1	1	13	0	0	0	0	0	0	0	0
colour	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
what_river	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
what_water_mass	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
what_mountain	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
what_mountain_range	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
what_planet	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
unknown	58	32	4	0	0	55	3	0	0	55	1	0	0	31	1	0	0	31
Totals	**159**	**41**	**22**	**1**	**1**	**136**	**21**	**2**	**1**	**135**	**1**	**0**	**0**	**40**	**1**	**0**	**0**	**40**

Table 2: Results by Query Type. The columns C and NC show the numbers of queries of a particular type which were classified correctly and not correctly. Those classified correctly are then broken down into Right, ineXact, Unsupported and Wrong for each of the two runs Run 1 and Run 2. Finally, those classified incorrectly are broken down in the same way.

5.2 Results

Results are summarised by query type in Table 2. In respect of query classification it shows for each query type the number of queries assigned to that type which were correctly categorised along with the number incorrectly categorised. The overall rate of success was 79.5% which is broadly comparable to the 85% achieved in TREC. However, 90 queries were classified as unknown, i.e. nearly half of the set of 200. 58 of these were correctly classified. This shows the need to add further query types to the system in order to reduce unknown queries to a minimum. In addition various phrases which were frequently used in queries were not anticipated. Examples include 'à quelle époque' and 'à quel moment', both of which were classified wrongly as unknown rather than when.

The performance of question answering in Run 1 can be summarised as follows. Out of the 159 queries classified correctly, 21 were answered correctly. Out of the remaining 41 queries classified incorrectly a further 2 were answered correctly. Overall performance was thus 23 / 200 i.e. 11.5%. Results for Run 2 were very similar. 21 of the 159 queries were answered correctly along with one of the 41 queries giving a total of 22 / 200 i.e. 11%. In both runs 117 questions were answered NIL. We discuss these results below.

5.3 Platform

We used a Dell PC running Windows 2000 and having 256 Mb RAM. The whole system was written Quintus Prolog Release 3.4.

6. Conclusions

Overall performance of our system was modest at 11.5% in Run 1 but this was not worse than that in TREC 2002 where the score was 10% in Run 1. The best performance was on queries of type what_city in which the system scored 5 / 9 i.e. 56%. Our system was simple, re-used as many components as possible and was constructed in a short timeframe. Our limited aim was to identify the key issues in cross-language QA and in this we were successful. We now consider the components of the system in turn before discussing our findings regarding the project in general.

Query categorisation was not a success with 90 queries classified as unknown. Performance on the remainder was 79.5%. To improve this, further examples need to be examined and appropriate keywords and phrases added. It is likely that our simple approach can give a better result even with such simple changes and without the need for complex analysis. Turning to query formulation and document retrieval, this was the first time we had used our own retireval engine – for TREC we used the TOPDOCS and O'Gorman (2003) searched the web with Google. The query formulation using boolean operators and exact searches for untranslated quotations achieved much higher precision than was possible with the TOPDOCS where the PRISE engine no doubt uses a vector space type algorithm. On the other hand such an approach can lead to no documents being returned in the first instance, whereupon a query relaxation technique must be adopted in order to re-submit a slightly more general query. Unfortunately we had no such strategy and this was the main reason why so many NIL results were returned (117 / 200 queries, i.e. 58.5% overall).

For named entity recognition essentially the same system was used as for TREC. This performs quite well and its main limitation is missing entity types which could readily be added. The main addition here was the entity general_name. In just two cases its use for processing queries of unknown type resulted in correct answers. For Query 106 'Of which political party Rudolf Scharping is member?' Run 1 returned the correct answer 'Social Democratic'. Run 2 returned the answer 'Kohl'. Run 2 returned the correct answer to Query 160 'To give the name of a German philosopher': 'Habermas'. Run 1 returned the answer 'Western'. Finally, in at least one case where the answer was wrong it was not nonsensical – Query 8 'Which largest is the exporting country of pétrole gross?' (system's answer: 'Kuwait'). The last component of the system is answer selection. The two strategies used (highest-scoring and most frequent) were the same as in TREC and did not perform well. No answer patterns such as Hovy (2002) are used and so the selection is just being made on non-structural correspondences between terms in a query and candidate answers such as co-occurrence measures. We now present our findings in terms of the specific issues of cross-lingual QA.

- Translation of queries. Our approach is heavily reliant on obtaining a good translation. The results from Google were adequate but there were many errors which affected subsequent processing. A better strategy has to be worked out, perhaps involving a more detailed grammatical analysis of the query based on a specialised knowledge of query forms (rather than texts in general) followed by submission of individual query constituents for translation. In addition, several engines could be used and the results combined to overcome repeated errors of specific types committed by particular systems.

- Translation of names within queries. Some names were recognised by Google (e.g. 'États-Unis') and hence translated correctly (e.g. 'United States'). However, many were left untranslated (e.g. Tchétchénie). The document collection refers to names in their American English form which can be radically different from the French term. For example the equivalent of Tchétchénie is Chechnya though it can appear in other variants. Generating the correct translation(s) for the whole range of names which might apear in a factoid question is a major task which we have not yet looked at in detail. Without accurate translations we will not answer questions mentioning such names correctly.

- Diacriticals. Our approach to these was to leave them in place for translation but to remove them entirely before question classification. This unsophisticated approach must surely lose important information. The processing of texts using diacriticals raises issues which we have not previously encountered when working with English documents. The most obvious of these is the need to match a word containing the appropriate accents with its equivalent without, while at the same time giving priority to exact matches. The present cross-language system was essentially a monolingual English one with a French front end but in the context of monolingual French-French QA attention would have to be paid to query formulation unless the retrieval engine used explicitly supported a morphologically sophisticated model of multilingual documents.

- Search queries returning no results. Intentionally narrow search expressions submitted to DTSearch could result in no matched document portions. This happened more often than we anticipated, the main reason being incorrectly translated (or not translated) parts of the input query. The answer is better translation together with query relaxation.

- General names. Our experiment in recognising general names was not very successful. Part of the problem is accurate verification of candidates without knowing the question type. A better strategy has to be worked out for this as there will always be unknown questions.

8. References

Clarke, C. L. A., Cormack, G. V., Kemkes, G., Laszlo, M., Lynam, T. R., Terra, E. L., & Tilker P.L. (2003). Statistical Selection of Exact Answers (MultiText Experiments for TREC 2002). In E. M. Voorhees and L. P. Buckland (Eds) *Proceedings of the Eleventh Text REtrieval Conference (TREC 2002), Gaithersburg, Maryland, November 19-22, 2002*. NIST Special Publication 500-251. Gaithersburg, MD: Department of Commerce, National Institute of Standards and Technology.

DTSearch (2000). www.dtsearch.com .

Google Translation (2003). www.google.ie/language_tools?hl=en .

Hovy, E. et al. (2002). *A Typology of over 140 Question-Answer Types.* Accessed 2002. www.isi.edu/natural-language/projects/webclopedia/Taxonomy/taxonomy_toplevel.html

O'Gorman, A. (2003). Open Domain Question Answering in French. Undergraduate Final Year Project, University of Limerick, Ireland.

Sutcliffe, R. F. E. (2003). Question Answering using the DLT System at TREC 2002. In E. M. Voorhees and L. P. Buckland (Eds) *Proceedings of the Eleventh Text REtrieval Conference (TREC 2002), Gaithersburg, Maryland, November 19-22, 2002*. NIST Special Publication 500-251. Gaithersburg, MD: Department of Commerce, National Institute of Standards and Technology.