CRL's TREC-8 Systems
Cross-Lingual IR, and Q&A

Bill Ogden, Jim Cowie, Eugene Ludovik, Hugo Molina-Salgado,
Sergei Nirenburg, Nigel Sharples, Svetlana Sheremtyeva
Computing Research Laboratory, New Mexico State University
January 1999

Abstract

This paper describes the systems used by CRL in the Cross-lingual IR and Q&A tracks.
The cross-language experiment was unique in that it was run interactively with a mono-lingual user simulating how a true cross-language system might be used. The methods used in the Q&A system are based on language processing technology developed at CRL for machine translation and information extraction.

Cross-Lingual IR

Can Monolingual Users Create Good Multilingual Queries?

Our interest in Interactive and Cross Language Text Retrieval has led to the design of a unique user interface for the cross language task. While many automatic techniques for query term translation and disambiguation have been proposed and tested, little work has involved the evaluation of a cross language system in combination with its user. We and others have proposed designing an interface that allows the user to help disambiguate terms provided by a system by providing "back-translations" of the system selected terms from which a monolingual user can select the appropriate meanings. The MULINEX system (http://mulinex.dfki.de) provides a query assistant feature with just such an interface. For our cross-language track experiment we wanted to see if these types of interfaces would help a monolingual user create good multilingual queries.

In our experiment, a single English speaker, who had little or no experience with German, French, or Italian, generated queries in each of these languages for the cross-language track run. For each topic, the user would read the English title, description, and narrative, and select the English terms from these sections judged to be the best query terms. They were only allowed to select terms that were contained in the original English topic. Then for each of the other target languages, the system showed extended English definitions of potentially relevant cross language query terms and phrases alongside their translations (see Figure 1). Only those terms that actually occur in the target data were presented to reduce the number of alternative terms. The user then selected the English definition that most accurately reflects the intention in the original query. The query terms selected for each language were used to retrieve and rank documents for that language and the results for all languages were merged into the final ranked list.

Retrieval
We conducted our cross-language runs using the Unicode Retrieval System Architecture (URSA) a multilingual retrieval engine that indexes and retrieves text using a common encoding scheme for all languages. Therefore, encoding for all texts were first converted to Unicode. URSA indexing of

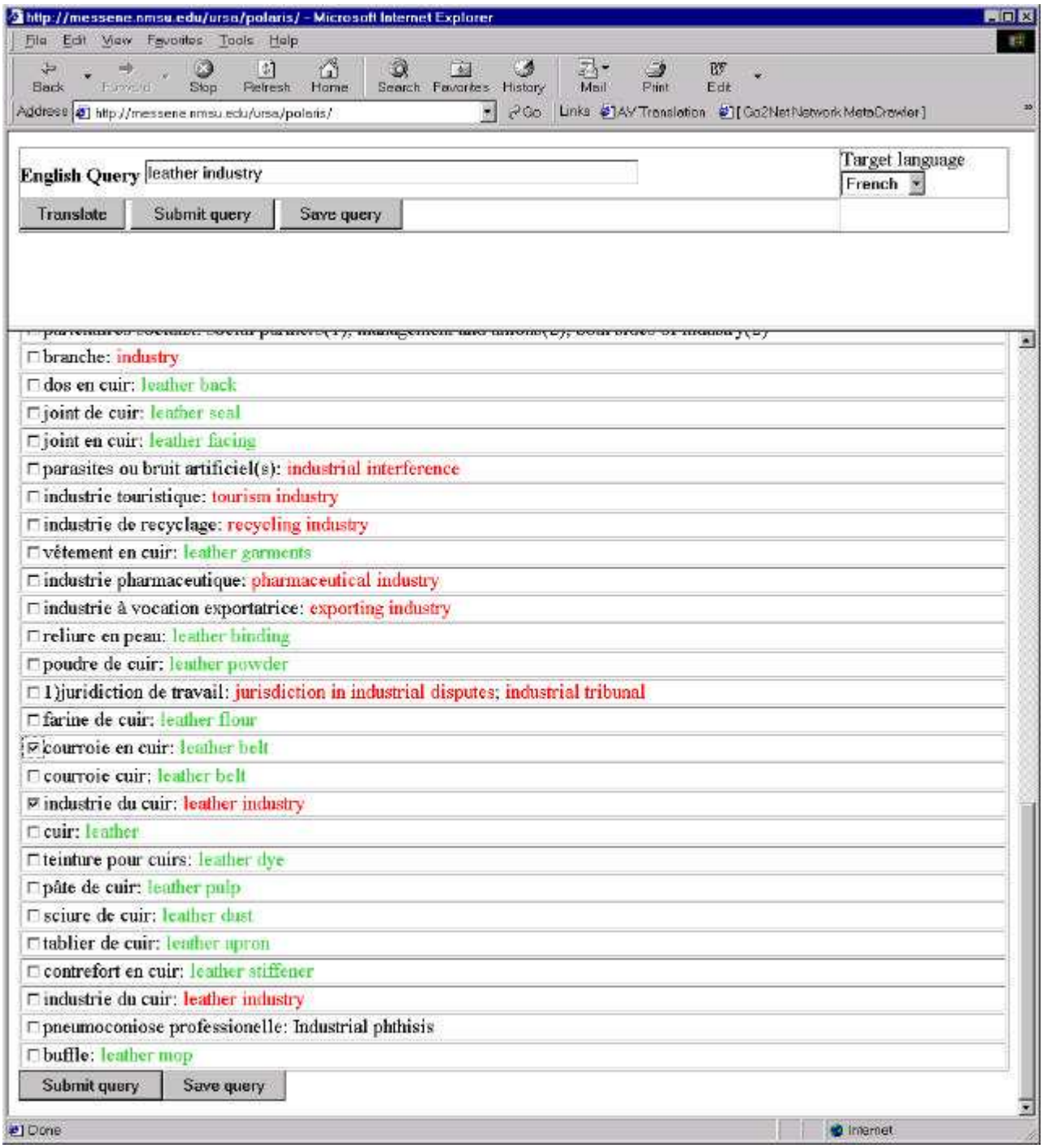

Figure 1. User-assisted cross-language query translation

the text used only simple stemming procedures specific for English, French, Italian and German. No other language specific compound word, phase indexing or other types of language processing was attempted. Consequently, one could expect an overall improvement in the performance of the base lined system, given the right effort. In this experiment we were only concerned with comparing the performance of the system when the cross-language queries were generated by the

system with help from a monolingual English-only speaker and the queries that were hand built by native language speakers and provided by NIST.

This was a preliminary experiment designed to test the feasibility of our approach. As usual the quality of the bilingual dictionaries will have a strong effect on the outcome. Some good query terms just were not present in the bilingual dictionaries used. In addition, our retrieval and ranking software could be better tuned to take advantage of the forms of the dictionary entries and phrases.

Merging

Merging the TREC multi-lingual queries is a constant issue for the cross-language studies. Our query system produces an ordered list (by score) of document ID's and the score for each language. The scores for each language are not comparable therefore the query results can not be merged using the score directly. Our technique was similar to that reported by the IBM group at TREC-7 [7] and involved obtaining a probability estimate that a returned document is relevant which and comparing these estimates between language retrieval systems. We used TREC-7 topics and results with our query system to obtain the performance for each language in terms of a sequence of relevance probabilities based on a precision score ordered by rank. To obtain the relevance probability we compared the results of the query system to the NIST supplied relevance tables (qrels) that specifies whether a document is relevant to a particular query. For each language, we generated a table mapping a rank index (from one to one thousand) to a precision score at that rank. For example, this tells us that a document at rank index 8 has a precision of 0.452, whereas a document at rank index 100 may have a precision of only 0.083. These rank-precision tables are roughly linear when plotted on a graph of precision vs. log (rank), so using linear regression an estimate of the relevance probability can be obtained for a given rank. These probability values are directly comparable between the different query systems, so the results for each query system can

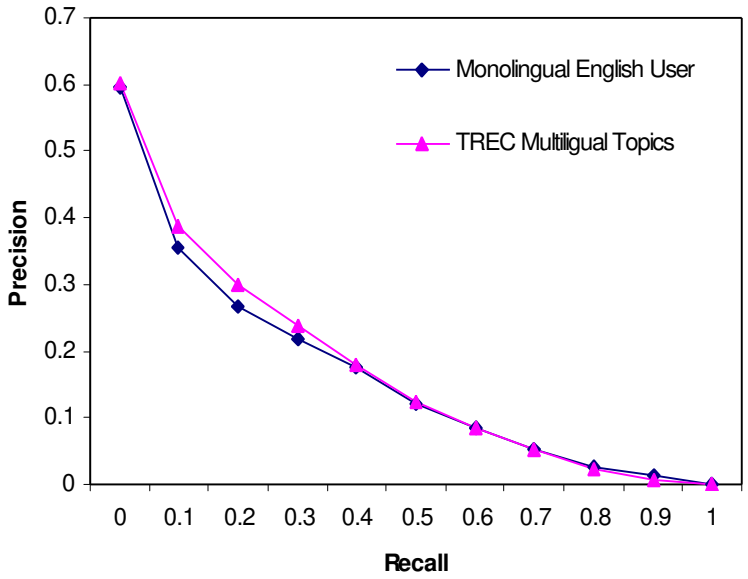

Figure 2. Precision –Recall performance of the cross-language retrieval system comparing the TREC supplied topics to those generated by the monolingual English user with help from the system

be merged using the probability value as a sort key.

Analysis

The primary analysis compares the results obtained by the monolingual user to the results obtained with the hand-translated queries provided by NIST for the cross language topics. As can be seen in Figure 2, the overall Precision/Recall curves for the two conditions are quite similar indicating that the user who knows no Italian, French, or German can use the system to generate queries that are as good as ones generated from the human translated TREC topics for these languages. The combined results shown in Figure 2 contain a large portion of English documents as well as the languages the user does not know. So, a more informative look at the data is shown in Table 1. Here the data show that indeed the English user is not doing as well as the baseline provided by the human translated TREC topics. For Italian and French, the user is doing about 85% of the baseline performance and for German it is worse (70 – 74 percent of the baseline).

	AveP	Recall At P(.20)
Italian – English User	.1770	.3249
Italian – Baseline TREC	.2077	.4027
% baseline	85%	81%
French - English User	.2722	.3980
French - Baseline TREC	.3236	.4682
% baseline	85%	85%
German – English User	.1110	.2297
German – Baseline TREC	.1592	.3103
% baseline	70%	74%

Table 1. Retrieval performance for individual languages comparing English user with TREC monolingual queries.

The fall off in performance Retrieval can have a number of reasons. For example the dictionaries that were used could have been lacking significant query terms or phrases. Indeed, if a query term could not be translated, the present system would not provide any alternative. Therefore, a number of simple improvements can be made to make the system better. With the more sophisticated tuning of the system, it can be expected that monolingual users will indeed be able to query in languages they cannot understand.

Question Answering

Extraction Based Method

CRL's approach to the Q&A problem is based on the Mikrokosmos Ontology [4]. The Ontology is intended to allow the representation of complex meanings. It consists of around 5,000 concepts linked using 200 relationship types. Each concept is linked to other concepts through up to 16 different relationships. The Ontology is being used principally to support machine translation, but recently we have been investigating its use as a control architecture for information extraction [2,3]. In this application a static template is defined by naming slots and defining potential slot fillers using the names of concepts from the Ontology. For example:

```
ELECTION
    {"ELECT", "ELECT"}
    {"PERSON-ELECTED", "HUMAN}
    {"PLACE", "PLACE"}
    {"DATE", "TIME"}
    {"POSITION-ELECTEDTO","SOCIAL-ROLE"}
```

defines an election template. The first element being the slot label, and the second the appropriate concept that must be attached to an element that would fill this slot. Our idea for question answering was to use the question to dynamically define such a template (partially filled with strings from the question), use a Boolean retrieval system to retrieve documents in which the key phrases, or equivalents occur, and extract the missing information -- the answer by carrying out the extraction process.

The amount of effort involved in this task was a total of six man weeks. Wherever possible off-the-shelf components were used. The Boolean retrieval was not completed in time for the evaluation, and the top five documents supplied by the AT&T retrieval engine were used. This had an impact on performance, as our whole method, at present, is dependent on information being localized in a single sentence in the document, which is not guaranteed with a general purpose ad-hoc retrieval.

Methodology

Our complete system consists of three main phases:

- Question Analysis - Recognize question structure and type
- Retrieval - Query building and document structuring
- Answer Generation - Sentence selection and answer selection.

Each of these is described briefly in the sections below.

Question Analysis

The basic processing undergone by the question and by sentences in the retrieved documents is the same. First the document is processed by a part of speech tagger, this marks each word in the sentence with one part of speech. In our current system we use a statistical tagger from MITRE. The text is run independently through the CRL Diderot name recognition system [5]. This recognizes names of organizations, places, people, and a variety of other units of interest (dates, money percentages etc.) The current complete list is shown in the table below. The labels are names of concepts from the Mikrokosmos Ontology.

Table of Elements recognized for the Q&A task

LINEAR-SIZE	ELECTRICITY	POPULATION-DENSITY	NATIONALITY
AREA	ENERGY	TEMPORAL-OBJECT	INHABITANT
VOLUME	VELOCITY	TIME-OBJECT	MATERIAL
LIQUID-VOLUME	ACCELERATION	AGE	EVENT-NAME
MASS	TEMPERATURE	NAME-HUMAN	PRODUCT-TYPE
RATE	COMPUTER-MEMORY	ORGANIZATION	NUMERIC-TYPE
PRESSURE		PLACE	DATE

The results of part of speech tagging and name and concept recognition are merged and the words are grouped into phrases, preference being given to the text units discovered by concept recognition. Verb and noun phrases and prepositional phrases are identified. A simple lexicon based stemming algorithm is then applied to the heads of all phrases and provides the citation forms needed to support lookup in the English to Ontology Lexicon.
Patterns are then applied to recognize noun phrase and verb phrase; phrases recognized by the name and measure recognition phase are not merged into noun phrases. In every case a head noun is identified. The head noun or verb is looked up in an English to Ontology lexicon. At this point we are ready to match the question against a set of skeletal question structures held in a "question lexicon". This allows the many ways that a question can be specified to all be mapped to a request for the same answer. Each entry consists of three parts:

<Type of Answer needed> <Additions to retrieval query> <Question pattern>

Where:

<Type of answer needed> specifies the ontological type of the answer needed

<Additions to retrieval query> specifies ontological concepts that should be mapped to lexical items to be used in the query process

<Question pattern> Is a pattern containing strings, which should be in the question, ontological types,, and Kleene stars, which allow matching any unit of question text. There is an implied "*" at the end of every question pattern. Currently there are some 500 question patterns in the system. Below we show the patterns used to handle questions on temperature.

Temperature Question Patterns

TEMPERATURE	THERMOMETRIC-UNIT	* what * temperature
TEMPERATURE	THERMOMETRIC-UNIT	* how hot
TEMPERATURE	THERMOMETRIC-UNIT	* how cold
TEMPERATURE	THERMOMETRIC-UNIT	* how many degrees
TEMPERATURE	THERMOMETRIC-UNIT	* how high * temperature
TEMPERATURE	THERMOMETRIC-UNIT	* how low * temperature

TEMPERATURE	THERMOMETRIC-UNIT	* what * melting point
TEMPERATURE	THERMOMETRIC-UNIT	* what * boiling point
TEMPERATURE	THERMOMETRIC-UNIT	* what * freezing point
TEMPERATURE	THERMOMETRIC-UNIT	* how many * THERMOMETRIC-UNIT

Temperature is a concept which is an object consisting of a NUMERIC-UNIT and a THERMOMETRIC_UNIT. The second element specifies that lexical entries attached to the concept THERMOMETRIC-UNIT should be included in the queries generated by the retrieval component of the system. The first pattern would recognize "At what temperature does tin melt?'. The last pattern contains a concept in addition to strings, in lower case. This would match questions such as "How many degrees centigrade is the melting point of tin?".
The question recognition system uses dynamic programming to select the closest matching question pattern. Strings are matched with strings in the question, and concepts are matched with the head concepts found for each phrase. If a direct match is not found the concept's parent in the "IS-A" hierarchy will also be tried. This information is then passed both to the retrieval system query builder and to the answer extraction system.

Retrieval

The query building component of the system was not integrated in time for use in this evaluation [1]. Instead the top 5 documents returned by the AT&T system, which were provided for the evaluation, were used. A brief description of the eventual operation of the query builder is given here.
Our goal is to find a text with a single sentence which specifies the answer in the context of all the constraints of the question. However, the constraints may need to be relaxed, and synonyms generated to allow a matching sentence to be found. The query system also expands the answer indicator concepts using the ontological lexicon. The THERMOMETRIC-UNIT will become "centigrade OR fahrenheit OR kelvin OR c OR f OR k". A boolean retrieval system is used and the initial query attempts to find all the phrases in a single sentence. If this fails then a second retrieval is attempted using head words. A third retrieval is attempted where head words are substituted by their synonms. If all the above fail then the synonym query is retried with the constraint that all the terms are in a paragraph.
The benefits of giving all the terms in a question equal weighting, and of only performing stemming and term expansion in response to the initial query failure, are that texts are obtained where all the information specified is found in a close context.

Document Structuring

The retrieved documents undergo the same language processing steps as was carried out on the query. Each sentence is part of speech tagged. Name recognition is run on whole documents, which allows much more accurate performance than processing single sentences. Phrases are recognized, and heads of phrases are looked up in the English to Ontology lexicon. The resulting structure, for each document, is then passed to the question answering phase.

Answer Generation

The structured question is used as a template and matched against each sentence in the document. Each sentence receives a score for each string and each concept in the question which matches a text unit in the sentence. If no text unit matches the concept required then the sentence is rejected, otherwise the answer string is produced accompanied by a score for the number of question slots filled in producing this answer. A high number of slots gives a high score. Once all the documents have been processed all the answers are sorted by score and the top five picked. In this preliminary system the answer selection process only requires the answer concept and does not specifically check that the expected answer object is present. Thus *tall* would be an acceptable answer for a linear size question. For the TREC tasks the answers were expanded on either side up to the maximum allowable number of bytes containing whole words. The initial answers produced by

Sample Question and Answer

The following shows a question, the resulting structure, and the set of answers obtained, all from the same document from the LA Times.

```
% How tall is the Eiffel Tower?

answer-indicator LINEAR-UNIT
np "the Eiffel Tower" "tower"
```

```
LA061789-0071    2.0 CRL  1,000-foot
LA061789-0071    2.0 CRL  short
LA061789-0071    0.95 CRL  76-foot
LA061789-0071    0.95 CRL  90-foot
LA061789-0071    0.95 CRL  Too Tall
```

Performance

Results were submitted for the 250 byte and the 50 byte tasks.

250 Byte Responses

33 - no answer
89 - no correct answer
78 - correct answer in 5 responses
Mean rank - 0.268

50 Byte Responses

33 - no answer
97 - no correct answer
70 - correct answer in 5 responses
Mean rank - 0.22

Future Work

Our top priority at the moment is to get retrieval integrated into the system. More sophisticated algorithms for answer selection are also required. For example not producing as an answer something which was specified in the question; some modicum of syntax will also help in matching

sentences to the question template. The question lexicon needs to be expanded to cover more question types.
A web search version will be built to allow the demonstration of the process on non-TREC data. Other knowledge sources will be incorporated to handle answers that are unlikely to be explicitly specified in documents (What is the capital of France?). The method is not language independent, but the components used part of speech tagging, phrase recognition, name recognition and an ontological lexicon are already available for Spanish and Chinese, so the development of question answering systems for these languages should be possible in a relatively short period of time [6,8].

References

[1] Cowie, J. 1999, Collage: An NLP Toolset to Support Boolean Retrieval, in "Natural Language Information Retrieval" editor: Tomek Strzalkowski, Kluwer Academic Publishers
[2] Cowie, J. and Y. Wilks 1999 *Information Extraction*, in "A Handbook of Natural Language Processing: Techniques and Applications for the Processing of Language as Text", Editors: Robert Dale, Hermann Moisl, and Harold Somers. Marcel Dekker Inc 1999
[3] Cowie, J., E.Ludovik, H. Molina-Salgado. 1998. *Improving Robust Domain Independent Summarization*, proceedings of "Natural Language Processing and Industrial Applications", Moncton, Canada, 1998
[4] Mahesh, K., and S. Nirenburg, 1995, A situated ontology for practical NLP, in Proceedings of the workshop on Basic Ontological Issues in Knowledge Sharing, International Joint Conference on Artificial Intelligences (IJCAI-95), Montreal, Canada.
[5] Cowie, J., Guthrie, L., Wakao, T., Jin, W., Pustejovsky, J., and Waterman, S. 1993.The Diderot Information Extraction System. In Proceedings of the First Conference of the Pacific Association for Computational Linguistics, (PACLING 93), Vancouver, Canada
[6] Cowie, J. 1996. CRLs approach to MET (Multilingual Named Entity Recognition), Proceedings of the Tipster Text II 24 Month Workshop, Morgan Kaufman, May 1996
[7] M. Franz, J. McCarley, S. Roukos, "Ad hoc and Multilingual Information Retrieval at IBM" *Proceedings of the Seventh Text Retrieval Conference (Trec-7).* E.M. Voorhees and D.K. Harmon, Editors, NIST Special Publication, 500-242. 1998.
[8] Sheremetyeva S., J. Cowie, S. Nirenburg and R. Zajac. *A Multilingual Onomasticon as a Multipurpose NLP Resource*. 1998. Proceedings of the "First International Conference on Language Resources and Evaluation", Granada, Spain.