IR-72

Applications of Multilingual Text Retrieval

W. Bruce Croft, John Broglio and Hideo Fujii

Computer Science Department
University of Massachusetts, Amherst
Amherst, MA 01003-4610
Telephone: (413) 545 0463
Email: croft@cs.umass.edu

Abstract

The recent enormous increase in the use of networked information access and on-line databases has led to more databases being available in languages other than English. The Center for Intelligent Information Retrieval (CIIR) at the University of Massachusetts is involved in a variety of industrial, government, and digital library applications which have a need for multilingual text retrieval. Most information retrieval research, however, has been evaluated using English databases and queries, and relatively little is known about how well advanced statistical techniques that incorporate ranking and term weighting perform in different languages. We describe our experience with a range of projects involving text retrieval in Spanish, Japanese and Chinese. The issues covered by these projects include document representation techniques such as morphology and segmentation, query formulation and expansion techniques, relevance feedback, and comparisons of retrieval effectiveness with English databases. The results indicate that advanced statistical techniques are effective in a wide range of languages, and that new languages can be incorporated with only moderate effort.

Introduction

The recent enormous increase in the use of networked information access and on-line databases has led to more databases being available in languages other than English. The Center for Intelligent Information Retrieval (CIIR) at the University of Massachusetts (Croft 1995) is involved in a variety of industrial, government, and digital library applications which have a need for multilingual text retrieval. In cases where the documents are only available in a foreign language, multilingual text retrieval provides access for native speakers and people who have some familiarity with the language and can generate queries and read documents with the aid of on-line dictionaries and thesauri. A multilingual text retrieval system can also be used to identify good documents for translation. Even in cases where the documents are available in both English and the foreign language, there is a role for multilingual text retrieval. The Library of Congress, for example, is working with the CIIR on providing access to the laws and regulations of many countries, both in English and in the native language.

Despite the interest in multilingual retrieval, most information retrieval research has been done using English (Salton and McGill 1983, Harman 1994). Groups in France, Germany and Israel have reported results on some aspects of text retrieval in their native languages, but little has been published on Asian languages such as Japanese, Chinese and Korean, or other important languages such as Arabic.

Much of what has been published about these languages has focused on Boolean retrieval, where documents are retrieved based on containing a combination of words specified in the query using AND, OR and NOT operators. Statistical retrieval, which emphasizes simple, natural language queries and ranked output, has been shown in many experimental studies to produce significantly better retrieval results (Turtle, 1994) and has recently become the predominant approach in commercial systems.

More research is needed, therefore, on the performance of advanced statistical techniques with different languages. Specifically, the major issue is whether techniques that have been shown to be effective on English can be transferred to other languages, or how they have to be modified to achieve the best performance. A related issue is how easily a statistical retrieval system can be modified to incorporate new languages.

The basic information retrieval processes are representing the information need (query formulation), representing documents (indexing the text), comparing these representations (retrieval), and evaluating the retrieved documents (relevance feedback). Whenever a new language is incorporated, each of these processes could be affected and the techniques used may need to be different than those developed for English databases.

In this paper, we describe our experience with multilingual text retrieval, analysis and routing software. The languages used in our current projects are Spanish, Japanese and Chinese. The specific issues we will discuss and the information retrieval processes they are related to are:

1. **Document Representation**
    - a) **Character encoding:** Documents and queries in different languages must be displayed and input using non-ASCII character encodings.
    - b) **Spanish morphology:** Different techniques for removing suffixes from Spanish words in order to relate word variants that arise from inflectional and derivational morphology.
    - c) **Segmentation in Chinese:** In contrast to English, word tokens in these Asian languages are not separated. One of the first steps in indexing text documents, then, is to decide what constitutes an indexing token.
    - d) **Combining representations:** In languages that require segmentation, both character-based and word-based representations can be used to improve retrieval effectiveness.
2. **Representing Information Needs**
    - a) **Query processing in Japanese:** Structure in a user's natural language query can be represented in order to improve effectiveness.
    - b) **Query expansion in Japanese:** Corpus analysis can be used to automatically expand queries with related concepts.
    - c) **Relevance feedback in Chinese:** User feedback on the relevance of retrieved documents can be used to automatically modify the initial query. Character-based languages can require different techniques than those developed for English.
3. **Retrieval**
    - a) **Comparing the effectiveness of Japanese and English retrieval:** A central issue is whether statistical techniques developed for English are as effective with other languages. Comparing retrieval effectiveness with

similar databases in different languages is difficult but provides valuable information.

The next section of the paper provides an overview of the INQUERY text retrieval engine, which is the basis of the CIIR projects. This overview focuses on the important functionality of the system and the query language that is used to represent information needs. The remaining sections address the issues listed above. We conclude by summarizing the major results of our multilingual projects and indicating future research directions.

The INQUERY Retrieval System

The INQUERY retrieval engine used in the CIIR language experiments is based on a probabilistic model of retrieval using a Bayesian net framework (Turtle and Croft 1991). The system has been used in a variety of research projects and applications, and it has been consistently very effective in the government-sponsored TREC evaluations (Harman 1995). The following list gives a brief overview of the major features of INQUERY that are used in the research described here.

Ranked output - INQUERY computes the probability that a document is relevant to a query by combining evidence (such as word counts) from the text of the document and the corpus as a whole. The probability value is then used to rank the documents for presentation to the user. The ranking function used is similar to other advanced statistical systems that incorporate "tf.idf" weights (Salton and McGill, 1983). The effectiveness of this process has been demonstrated by recall/precision evaluations in TREC and other settings (e.g. Rajashekar and Croft 95), and there are significant usability advantages for ranked output.

Ability to handle both simple and complex queries - The INQUERY query language provides a range of operators which are used to specify how evidence in the document text should be combined to estimate relevance. This means that INQUERY's probabilistic framework can be used for simple word-based queries, Boolean queries, phrase-based queries or any combination. The query language is not designed to be used by searchers directly but rather is a target language for the user interface. In terms of the underlying inference net model, the query language operators specify how to combine probabilities from parent concepts. The #AND operator, for example, is a probabilistic version of Boolean AND that combines the parent concept probabilities by multiplying them. This follows from the assumption that the concept represented by the #AND only represents a document when all parent concepts also represent that document.

Examples of operators include:

- averaging and weighted averaging of evidence (#SUM and #WSUM),
- probabilistic Boolean (#AND, #OR, #NOT),
- strict Boolean that preserve probabilities (#BAND, #BANDNOT),
- proximity and phrase (#n, #UWn, #PHRASE),
- synonym (#SYN), and
- passages (#PARSUMn).

The #n proximity operator allows up to n words between the words in the argument, and insists they occur in the order given. The #UWn (unordered window) operator specifies that the words in the argument must occur in a text window of size n, but they can occur in any order. The #WSUM operator allows relative importance weights to be associated with different components of the query.

The #n, #UWn, #PHRASE and #SYN operators calculate probabilities based on the statistics of words in the documents. In a sense, they create new indexing concepts. In the case of #n and #UWn, the number of occurrences of words satisfying the proximity restriction are used to calculate a tf.idf probability, as with simple word-based concepts

New operators can be added to represent different linguistic structures or relationships.

Flexible and efficient indexing - The INQUERY indexing process is designed to be easily extensible in order to incorporate a variety of document structures (e.g. HTML, MARC, etc.), different morphological processing techniques such as stemming (Krovetz 1993), and domain-specific concept recognizers (e.g. marking occurrences of people, companies, locations, and dates).

Tools for query processing and query expansion - Natural language queries (in English) can be transformed into INQUERY queries using tools such as a part-of-speech tagger (Church, 1988) and a stop structure recognizer (removing phrases such as "Give me documents about ..."). Queries can also be automatically expanded using related phrases from the corpus (Jing and Croft 1994).

Support for relevance feedback and routing - User feedback on the relevance of retrieved documents can automatically modify either a query in a typical interactive session or a profile in a routing environment where incoming documents are compared to stored profiles (Salton and McGill 1983, Buckley, Salton and Allan, 1994). The two main steps in the query modification process are term selection and term weighting. Term selection (in English) involves choosing which words and phrases from the relevant documents should be added to the query. Term weighting is used to indicate the relative importance of the concepts in the new query and is based on frequency of occurrence in the identified relevant documents.

Character Encodings

The ASCII character set and the libraries of common computing languages are heavily biased towards English. Other languages using the Roman alphabet have additional characters, usually represented by some byte code greater than 128, which require some adjustment to routines for input, display, indexing and retrieval.

For non-Roman character representation, the issue is much more complicated. There may be several ways to encode non-Roman characters, all of which must be supported. Additionally, there may be different character sets for the same language, as in Serbo-Croatian (Cyrillic and Roman alphabets) or Chinese (Beijing "simplified" characters versus Taiwan "traditional" characters).

While it may be a simple matter to modify indexing and retrieval code to handle different character encodings, some care must be taken to ensure that a user's query is submitted to the system in the same encoding as the indexed database, and that the display present the proper encoding as well. This may be done by forcing the display and input character set to conform to that of the database, or more congenially by translating from one character set to another at runtime to accommodate the user.

Romance languages use accented vowels (e.g. â, è, é, ê), dieresis (ü), cedilla (ç) and enye (ñ). In most ASCII systems these are assigned to codes above decimal 128, and many systems display the letters for these codes without any adjustment. How the user enters the codes will vary from system to system, as will the adjustments in input programs to allow the codes to be recognized.

It is trivial to extend indexing and retrieval code from English to handle these characters, but that does not solve the questions of orthography and stemming. Given the extra effort necessary to key in the accents, a user may prefer to enter characters without accents. It is also common for accents to be entered incorrectly, or to be omitted because of familiarity with systems which do not handle them adequately. A simple method of spelling correction in this case is to reduce all accented characters to their unaccented counterparts.

There are several popular methods of encoding Japanese and Chinese characters. To complicate matters further, just as there are two popular Anglo-Romanizations of Mandarin (the old Wade-Giles and the modern Pinyin), there are two main graphic representations for Mandarin characters, the Taiwan traditional (BIG5) and the Beijing simplified (GB). There are several two-byte computer encodings for Chinese characters, with further variations. Although GB and BIG5 both use two bytes for each Chinese character, even when the graphic representation of a character is the same in both sets, the byte encoding will be different. Further confusion is possible, since a random two-byte code in BIG5 may coincidentally produce an unrelated or meaningless graphic character in a GB display.

Unicode (Unicode, 1994) represents a more rational approach to encoding multiple languages and should have increased impact as more software and database support for this standard emerges.

Spanish Morphology

Stemming is a general term used to describe the process of conflating word variants, usually by removing letters. The typical case in Western European languages is to remove suffixes to reduce all forms of a word to one sequence of letters called a stem (Krovetz 1993).

The purpose of stemming is to enhance recall. In English, inflectional endings such as plurals and verb suffixes must be removed. Beyond that the problem gets more complex.

For example, should "computer", "compute", "computation" and "computerization" all be conflated to the stem "compute"? Underconflation will hurt recall; documents will be missed that should be retrieved. Overconflation will hurt precision and negatively affect the user's reaction to the retrieval process: if the user enters the query "policy conflicts", she will be surprised to retrieve articles about "police conflicts".

One of the most effective (in terms of average recall and precision) stemming algorithms is due to Porter (1980), and consists of a large number of rules applied in steps that remove increasingly larger parts of a word's ending. The result of this process is a word fragment. In the case of the example given above, both "policy" and "police" stem to "polic".

When a new language is incorporated into a statistical retrieval system such as INQUERY, one of the major components of the system that must be changed is the stemmer. To test how difficult this would be, we built a Spanish stemmer on the model of the Porter stemmer for English, and tuned it for Spanish suffix morphology.

In most Romance languages, verb morphology is very productive. Where English may have four forms for a regular verb, with easily recognized suffixes, a language like Spanish can have fifty-five forms. Additionally, if the word stem ends in a consonant (such as c) which has both hard and soft pronunciations, the spelling of the stem will change. Using five megabytes of comprehensive morphological data provided by New Mexico State University, we culled out just the verb forms, including irregular forms, which our automatic stemming did not handle properly. This reduced our data space requirements tenfold while still giving us good performance.

Further research, suggested by the success of Cornell's simple Spanish stemmer (Buckley *et al*, 1995), showed that similar retrieval performance can be achieved with a suffix-trimming stemmer that only removes plurals and gender suffixes. This stemmer was written in less than one day, but other languages such as Arabic would be considerably more difficult.

Another approach to generating a stemmer for a new language is to derive corpus-based relationships between word variants using a measure of statistical dependence based on co-occurrence in text windows (Croft and Xu, 1995). If this measure of dependence for two word variants exceeds a threshold value, those variants are put into the same synonym or equivalence class. In order to reduce the computational requirements, two methods of deriving initial equivalence classes were tested in English. The first used the Porter stemming algorithm on the words in a corpus, and the second was based on the simple assumption that any word starting with the same three characters (trigram) may be related. The somewhat surprising result was that the algorithm that used virtually no linguistic rules but relied almost entirely on statistics performed the best in terms of recall-precision results on the test collections. The same approach is being tested for Spanish.

A drawback of this approach is that words which have no morphological connection can get conflated, if they accidentally co-occur for some reason. In the Spanish corpus, for example, the words desaceleración (deceleration) and desempleo (unemployment) co-occur nearly five percent of the time, and thus are conflated by the trigram stemmer, even though they clearly do not share a common stem. The problem here is that the three characters which they do share are a morphological prefix. In the first pass of the trigram stemmer, for example, one conflation class with 30 members included such unrelated terms as concesión (concession), consejo (council), constitucional (constitutional), constructor (builder), and contratista (contractor).

In cases such as these we must look to characters beyond the prefix to discover if the terms really share a root. Knowledge of common prefixes, such as des- or con-, was used to modify the initial classes. Preliminary results suggest that this method will perform as well as the Porter-style stemmer.

Segmentation in Chinese

In English and Romance languages (at least since the demise of Latin), a space indicates a break between words. In so-called agglutinating languages such as Turkic, and to a lesser extent Finno-Ugric and Germanic languages, some words are delimited by spaces while other words are compound, so that one unbroken lexical representation may contain several individual component words. An English example is "headache", but in other languages, compounding can be much more productive. Unless such compounds are teased apart and indexed as a sequence of individual items, recall may be adversely affected.

In Chinese, the issue is even more challenging: there is no default punctuation to indicate word separation, so a character could be a whole word or only part of a multicharacter word (Wu and Tseng, 1993). In order to identify the words in the text perfectly, it is necessary to understand the meaning of the text. On the other hand, it is possible to attempt to identify words automatically; for example, by consulting a dictionary and choosing the longest possible word formed by the characters seen. This may not be accurate if the characters actually were intended to represent two or more shorter words. In the worst case, the word chosen may be unrelated and both precision and recall will be negatively affected. Less damage is done if the long word happens to be related to the shorter words intended; this is similar to the case where the user enters terms that are more restrictive than actually intended, e.g., televangelist where television preacher is the desired concept. A number of relevant documents will be missed because they do not contain the word.

It is also reasonable simply to ignore the word segmentation problem and index the individual Chinese characters, using a character-by-character best match to select relevant documents for a given query. This can give false matches where the characters from one word in a query are in unrelated words in the documents, but if there are enough words in the query, mistakes will tend to be reduced by the cumulative evidence. Quality can be further enhanced by indexing the document collection on a character basis, and then segmenting the queries, either automatically or by hand. Character-based retrieval is discussed further in the next section.

The CIIR has developed a stochastic word segmentation program which uses a Hidden Markov Model algorithm (Huang *et al*, 1990) to find a best-fit word segmentation for a sequence of characters. One advantage of this approach is that it is an order of magnitude faster (over 100 Mbytes/hr) because dictionary lookup is used only during training, but not at runtime.

The fundamental concept behind a stochastic segmentation algorithm is a transition between states. Given the current state (a character or sequence of characters), the likelihood that the next transition should be a word termination or a word continuation is used to determine segmentation. The stochastic segmenter uses a Chinese lexicon during the training initialization phase to construct a reasonable model with equal probabilities for all transitions. The model is then trained on free text to adjust the probabilities based on actual text examples. This reduces the amount of error due to unusual words in the dictionary. The refined model can then be used to segment new text.

Examples of the segmentation done by the stochastic algorithm are:
上海的外商投资 (foreign investment in Shanghai) is converted to
上海的 (in Shanghai) 外商 (foreign business) 投资 (investment).
中国的汽车工业 (Chinese automobile industry) is converted to
中国 (China) 的 (of, part of Chinese) 汽车 (automobile) 工业 (industry).
The last sentence contains a small error that would not affect retrieval (的 is one of the most common Chinese words). Most segmentation errors occur with proper nouns, which usually end up as a sequence of single characters. This could be addressed with a proper name recognizer.

Combining Representations in Japanese

The Japanese written language uses heterogeneous character classes, where each class has a clear linguistic functional role: Kanji (or Chinese) characters are primarily used to express abstract or complex important concepts ideographically; Words written using Katakana characters are mostly phonetic loan words, especially from English. Hiragana characters are used for inflection or other functional words such as particles, auxiliary verbs, etc. Furthermore, one can often see English alphabetic words in a Japanese text, especially as proper nouns such as "IBM" or "C". In a recent study, we examined character-based and word-based representations for documents in INQUERY (Fujii and Croft, 1993). Further research has indicated that combining these representations in the INQUERY framework improves retrieval performance relative to either representation.

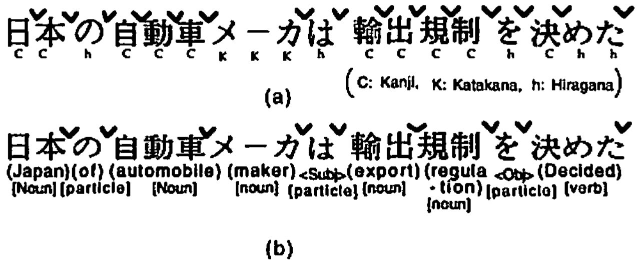

Figure 1. An example of character-based (a) and word-based (b) term boundaries.

The character-based indexing procedure used in the INQUERY experiments was as follows: All Hiragana characters are removed from the text; each Kanji character is individually indexed; sequences of Katakana characters or English characters are extracted as index terms. The word-based indexing technique extracts words (or some normalized forms like word stems), as the index terms. Segmentation was performed using the JUMAN program (Matsumoto *et al*, 1991). Examples of character-based and word-based indexing are shown in Figure 1 (a) and (b).

Figure 2 and 3 show how these indexing methods can be used in the INQUERY framework. In Figure 2, both document words and query words are decomposed into a set of Kanji characters. In Figure 3, each Kanji compound noun is divided into its components. In character-based indexing, the connections between a query word and the query characters are formulated by INQUERY operators. In word-based indexing, the connections between a query compound word and the compound elements are specified using the operators.

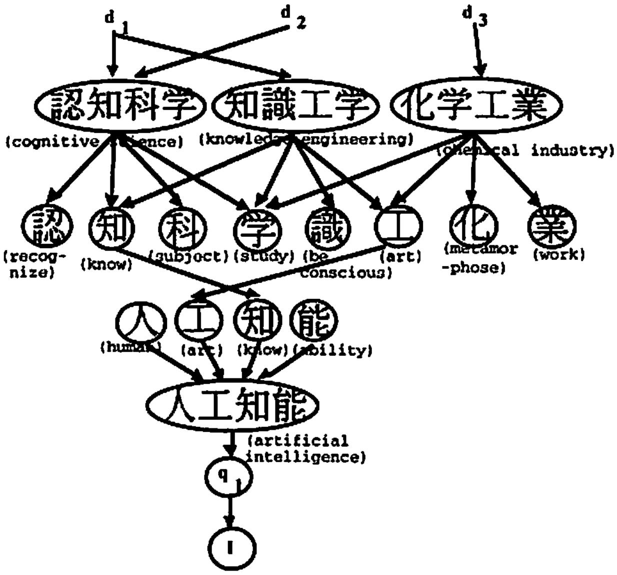

Figure 2. An example of Japanese character-based inference network.

In our earlier experiments (Fujii and Croft, 1993), we found that character-based representations produced similar retrieval effectiveness to word-based representations. Performance was, in fact, slightly higher at higher recall levels which appeared to be due to a low-level thesaurus effect from matching individual characters (suggested by Figures 2 and 3). More recently, we have combined both representations and found that retrieval effectiveness was significantly improved (more than 10% improvement in average precision), especially at lower recall levels. A combined representation means that documents must be indexed both by characters and words, which results in substantially larger index overhead. Given that word-based indexing is substantially slower than character indexing (due to the segmentation process), however, it appears that producing both representations would only be justified in applications that required the best retrieval effectiveness possible.

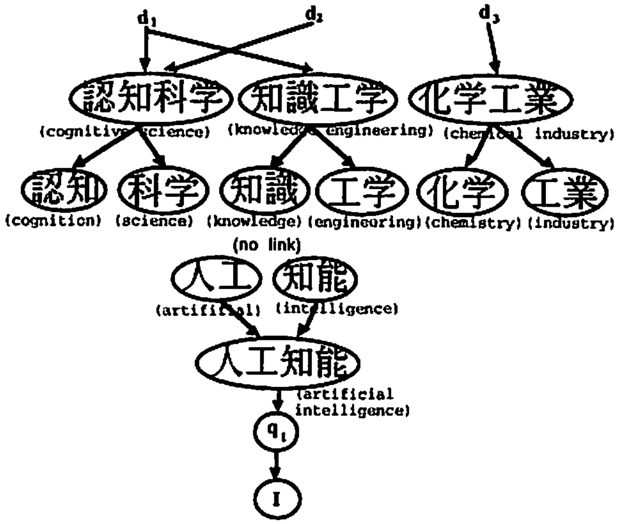

Figure 3. An example of Japanese word-based inference network.

Query Processing in Japanese

Queries in English can be submitted to INQUERY using either natural language or the structured query language or a mixture of the two. Natural language queries are transformed incrementally into complex structured queries in the INQUERY query language by a series of query text processing modules in order to improve retrieval effectiveness (Callan and Croft, 1993). Query text processing must minimally mirror the indexing text processing, but because query texts are much shorter than document collections, it is practical to use more thorough textual analysis.

In the case of Japanese, natural language queries are transformed into structured form according to the phrase structure. For example, a query, "I want to know about the advancement of Japanese companies in southeastern Asia" could be translated into

"#SUM(advancement #PHRASE(Japanese companies) #PHRASE(southeastern Asia))".

We have tested four models for transforming queries, namely NLQ, SHORT, LONG, and JOINED (Fujii and Croft, 1993). The NLQ (=natural language query) model does not assume any structure between words or characters. The SHORT model groups a set of Kanji characters in a word. The LONG model clusters nouns with adjectival modifications (e.g., Tounan [southeast(*n.*)] Ajia [Asia]). The JOINED model puts together LONG phrases which are connected by "-no" [of] in Japanese.

The experimental results using these transformations indicated that the SHORT and LONG versions of the query are effective with character-based indexing, but have no significant impact on the performance of word-based indexing. This is consistent with our results in English.

Query Expansion in Japanese

A central problem in text retrieval is reducing the mismatch between the words used in a query and those in the documents. Automatic query expansion through corpus analysis is a promising solution to this problem. Our approach to query expansion (called PhraseFinder) is based on the assumption that concepts found in similar lexical contexts may also be related semantically (Jing and Croft, 1994). For example, the words "connectionist" and "neural networks" might occur in similar lexical contexts, but rarely in the same documents. The semantic relationship captured is not necessarily synonymy, because PhraseFinder also might relate "connectionist" and "back propagation", which co-occur but have different meanings.

A PhraseFinder database is an INQUERY database of "pseudo-documents". Each pseudo-document represents a concept, usually defined as a noun sequence, that occurs in the document collection. The "text" of the pseudo-document consists of words that occur near the concept in the document collection. For example, a PhraseFinder database may contain an *amnesty program* pseudo-document that is indexed by *1986*, *act*, *control*, *immigrant*, *law*, etc. Although very different in implementation, the approach is similar in spirit to the distributed representation used by the LSI system (Deerwester *et al*, 1990).

The usual document retrieval algorithms are used to retrieve the pseudo-documents that represent concepts. Thus, INQUERY can use any structured query to retrieve a ranked list of concepts. A query is expanded by evaluating it against a PhraseFinder database, selecting the top ranked concepts, weighting them, and adding them to the query.

This approach has recently been tested on a small Japanese newspaper database (Han, Fujii and Croft, 1995). These experiments found many of the same results as English with regard to the PhraseFinder parameters such as the size of the text window used for determining associations and the weighting algorithms. Even though the database contained only 1,100 documents, small but consistent retrieval improvements were obtained with the automatic expansion. We are currently carrying out similar experiments using a much larger database.

As an example of the type of expansion phrases obtained using this technique, the query "Japanese automobile" retrieved the phrases "lead plate steel", "Japanese automobile industry", "production company", "GE", "automobile market", "automobile distribution", "automobile telephone", "MDC Company", "automobile part maker" and "European market".

Relevance Feedback in Chinese

Relevance feedback (Salton and McGill, 1983) is a technique where user judgments on the relevance and non-relevance of retrieved documents are used to modify the original query and improve the retrieval effectiveness. Although this is quite an old technique, much has been learned recently about feedback with full-text documents and larger numbers of training examples in the TREC experiments (Harman, 1995). The two basic steps in this learning process are term selection and term weighting. Term selection involves choosing the words or phrases to be added to the original query, and term weighting assigns relative importance to them.

Both term selection and weighting are based on the frequency of the term in the identified relevant documents and the corpus. In the current English version of INQUERY, for example, words are ranked according to their frequency in the set of relevant documents. The query is expanded using the highest ranked words, with the exception of words that occur in a table of the most frequent words in the corpus. The word "company", for example, may occur frequently in the identified relevant documents but would not be added to the query because it is one of the most common words in a newspaper corpus. The original query words and new words are then weighted using a “tf.idf” formula.

In the case of a Chinese version of INQUERY that uses character-based indexing, the default terms that are added to the query are single characters. People using this system felt that these added characters were not sufficient to focus the search. Given this, there were two obvious alternatives. One was to segment the texts of the identified relevant documents to identify Chinese words that could be added to the query. The alternative technique, which is being used while the segmenter is under development, is to identify frequent bigrams (2 character sequences) in the relevant document texts. The bigrams are added to the query using the INQUERY proximity operators and are weighted using the same formula as English words. The effectiveness of this approach is currently being evaluated, but initial user reaction is positive.

As an example of this relevance feedback process, we used the query
俄罗斯的经济改革 (economic reforms in Russia)
俄罗斯的 (Russian) 经济 (economic) 改革 (reform),
which was converted into the character-based query
#SUM(俄 罗 斯 的 经 济 改 革).

After identifying two relevant documents in the initial retrieved set, the modified query was

```
#WSUM( 5.338472 俄 2.094052 罗 1.607136 斯
 0.075717 的 1.014599 经 1.721905 济 1.908511 改 2.333531 革
!c! Relevance feedback added bigrams
 1.489387 #1( 经 济) 1.908377 #1( 改 革) 1.384988 #1(企 业)
 0.965981 #1( 政 府) 1.819754 #1( 罗 斯) 1.853917 #1( 俄 罗)
 3.100488 #1( 切 尔))
```

The new bigrams included the words for reform (改革) and economy (经济) which, although in the initial query, had not been identified as words. Other new bigrams were industry (企业), government (政府), two bigrams from Russia (俄罗, 罗斯), and a bigram that was part of a Russian government official's name (切尔).

Comparing Retrieval Effectiveness

One of the basic questions in multilingual text retrieval is whether advanced statistical techniques are as effective in other languages as they are in English. Japanese, for example, has many different characteristics to English, such as lexicon (e.g., number of loan words), morphology (e.g., no plural form), syntax (e.g., S-O-V word order), pragmatics (e.g., the paragraph structure is less well defined), and written language system (e.g., Chinese characters and no spaces between words).

Comparing retrieval performance requires comparable test databases and queries. A translated corpus may be ideal for this purpose, but we were forced to use the following procedure:

I) From some selected texts in both languages which contain the same information, measure the sentence length and the ratio. For this task, we used the English and Japanese pamphlets of Smithsonian museums. The result was 1 : 2.5 in characters for Japanese vs. English, i.e., 1 : 1.25 in byte length.

II) Choose a collection of one language, and get the statistics of text length frequencies. We used a Japanese collection of business newspaper articles about "joint ventures", which contains 890 documents.

III) Create a English population of documents of the same subject. We made this from a *Wall Street Journal* database of the corresponding years to the Japanese documents [year: 1987-1991, size: 498 MB, 163,092 documents], giving an INQUERY query, "joint venture".

IV) Using the text length frequencies of Japanese as a probability distribution, choose a set of English articles randomly from the population.

Table 1 shows the summary of our collections and queries in this experiment.

Figure 5 shows the recall-precision curves for the two languages. The Japanese queries performed better than the English at all recall levels, especially at low recall. Our test collection seems to be appropriately organized since the precision at 100% recall is almost the same. One possible explanation for the performance difference is that in Japanese, Kanji words, which are of Chinese origin, have more specific meaning than native Japanese words which are often written in Hiragana, and they are preferably used as a formal expression in a written text. Thus, Japanese lexical semantics is more narrowly specified than in English.

The results certainly indicate that the advanced statistical techniques perform well in Japanese, despite the significant differences in languages.

Table 1. Summary of the test collections.

	< **English** >	< **Japanese** >
Collection		
form:	newspaper articles	newspaper articles
subject:	joint ventures in business	joint ventures in business
source:	Wall-Street Journal 1987-1991	Mostly from Nikkei-Shinbun 1987-91
collection size:	890 articles (1255 kb)	890 articles (972 kb)
article length:	mean: 1192.0b	mean: 945.8b
(*1.25=1188.8)		
	S.D.: 732.3b	S.D.: 580.3b
(*1.25=725.4)		
	max/min: 4922/172b	max/min: 4044/138
Queries		
queries:	25 queries translated from Japanese	25 queries
query size:	5.2 words/query	8.7 char/query

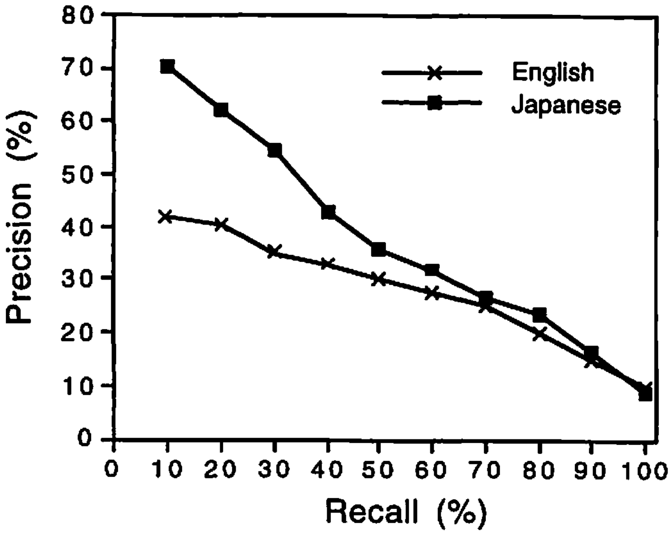

Figure 5. P-R Curve for English and Japanese.
(25 NLQ queries, using #phrase; Word-based in Japanese)

Conclusion

In general, the results presented here show that advanced retrieval techniques developed for English can be used effectively in a range of very different languages. Morphological processing, query expansion, and relevance feedback are all effective in English and can be applied, with some modifications, in languages as different as Spanish, Chinese and Japanese. We have also used similar approaches in a Finnish version of INQUERY. The new techniques developed for specific languages included stemming, segmentation and query processing algorithms. These were easily incorporated into the INQUERY framework.

We are currently investigating extending INQUERY with other languages, such as Arabic, Korean, and Ukrainian. Although we expect each of these languages to have their own unique problems, such as Arab morphology, the basic INQUERY framework should be adequate. One of the major issues is dealing with the multiple character encodings for query entry and document presentation.

Another research direction that we are pursuing is true multilingual search. That is, entering a query in one language and retrieving documents in several other languages. There is beginning to be evidence that this is a less difficult task than translation, and that tools such as multilingual dictionaries and statistical analysis of parallel corpora provide the basis for effective retrieval.

Acknowledgments

The authors wish to acknowledge the work of Paul Booth, Jay Ponte, and Chengfeng Han on Spanish and Chinese databases. This work was supported in part by the NSF Center for Intelligent Information Retrieval at the University of Massachusetts.

References

Buckley, C.; Salton, G.; Allan, J. (1994) The effect of adding relevance information in a relevance feedback environment. *Proceedings of ACM SIGIR 94*, Springer Verlag, 292-300.

Buckley, C.; Salton, G.; Allan, J; Singhal, A. (1995) Automatic query expansion using SMART: TREC3. In D. Harman, editor, *Proceedings of the Third Text Retrieval Conference*, NIST Special Publication 500-225,

Callan, J.; Croft, W.B. (1993) An evaluation of query processing strategies using the TIPSTER collection. *Proceedings of ACM SIGIR 93*, 347-356.

Church, K. (1988) A stochastic parts program and noun phrase parser for unrestricted text. *Proceedings of the 2nd Conference on Applied Natural Language Processing*, 136-143.

Croft, W.B. (1995) NSF Center for Intelligent Information Retrieval. *Communications of the ACM,* 38(4), 42-43.

Croft, W.B.; Xu, J. (1995) Corpus-specific stemming using word form co-occurrence. *Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval*, 147-159.

Deerwester, S.; Dumais, S.; Furnas, G.; Landauer, T.; Harshman, R. (1990) Indexing by latent semantic analysis. Journal of the American Society for Information Science, 41, 391-407.

Fujii, H.; Croft, W.B. (1993) A comparison of indexing techniques for Japanese text retrieval. *Proceedings of ACM SIGIR 93*, 237-246.

Han, C.; Fujii, H.; Croft, W.B. (1995) Automatic query expansion of Japanese text retrieval. Technical Report 95-11, Computer Science Department, University of Massachusetts, Amherst.

Harman, D. (1995) Overview of the Third Text Retrieval Conference (TREC-3). In D. Harman, editor, *Proceedings of the Third Text Retrieval Conference*, NIST Special Publication 500-225, 1-20.

Huang, X.D.; Ariki, Y.; Jack, M.A. (1990*) Hidden Markov Models for Speech Recognition.* Edinburgh University Press, Edinburgh.

Jing, Y.; Croft, W.B. (1994) An association thesaurus for information retrieval. *Proceedings of RIAO 94*, 146-160.

Krovetz, R. (1993) Viewing morphology as an inference process. *Proceedings of ACM SIGIR 93*, 191-202.

Matsumoto, Y.; Kurohashi, S.; Myoki, Y.; et al. (1991) *User's Guide for the JUMAN system - A User-Extensible Morphological Analyzer for Japanese*, Nagao Laboratory, Kyoto University.

Porter, M. (1980) An algorithm for suffix stripping. *Program*, 14(3), 130-137.

Rajashekar, T.B.; Croft, W.B. (1995) Combining automatic and manual index representations in probabilistic retrieval. *Journal of the American Society for Information Science*, 46(4), 272-283.

Salton, G.; McGill, M. (1983) *Introduction to Modern Information Retrieval.* McGraw-Hill, New York.

Turtle, H. (1994) Natural language vs. Boolean query evaluation. *Proceedings of ACM SIGIR 94*, Springer Verlag, 212-220.

Turtle, H.; Croft, W.B. (1991) Evaluation of an inference network-based retrieval model. *ACM Transactions on Information Systems*, 9(3), 187-222.

The Unicode Consortium (1994) The Unicode Standard, Worldwide Character Encoding.

Wu, Z.; Tseng, G. (1993) Chinese text segmentation for text retrieval. *Journal of the American Society for Information Science*, 44(9), 532-542.