Comparative Study of Monolingual and Multilingual Search Models for Use with Asian Languages

JACQUES SAVOY

Université de Neuchâtel, Neuchâtel, Switzerland

Based on the NTCIR-4 test-collection, our first objective is to present an overview of the retrieval effectiveness of nine vector-space and two probabilistic models that perform monolingual searches in the Chinese, Japanese, Korean, and English languages. Our second goal is to analyze the relative merits of the various automated and freely available toolsto translate the English-language topics into Chinese, Japanese, or Korean, and then submit the resultant query in order to retrieve pertinent documents written in one of the three Asian languages. We also demonstrate how bilingual searches could be improved by applying both the combined query translation strategies and data-fusion approaches. Finally, we address basic problems related to multilingual searches, in which queries written in English are used to search documents written in the English, Chinese, Japanese, and Korean languages.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval.

General Terms: Algorithms, Experimentation, Measurement

Additional Key Words and Phrases: Multilingual information retrieval, cross-language information retrieval, natural language processing with Asian languages, results-merging, Chinese language, Japanese language, Korean language, search engines with Asian languages

---

1 MONOLINGUAL IR FOR ASIAN LANGUAGES

During the last few years, interest in Asian languages, particularly in Chinese (C), Japanese (J), and Korean (K) has been increasing. Given the growing number of Internet pages and sites available in these languages, along with an ever-expanding number of online users[1] working with them, a better understanding of the automated procedures used to process them is clearly needed. These Asian languages also represent various external differences that, compared to European languages, present the IR community with very interesting challenges.

While the Latin alphabet consists of only 26 characters (or 33 in the Cyrillic and 28 in the Arabic alphabets), standard Asian languages require quite a larger number of characters (around 13,000 for the Chinese BIG5 encoding system, around 8,200 for Korean, and 8,800 for Japanese). When processing the languages of the far East, the implicit assumption that one byte corresponds to one character is no longer valid. These facts lead to additional challenges for anyone using typical Unix functions like `wc`, `sort`,and `grep`, and generally entail the use of more complex input and output methods [Lunde 1998].

This research was supported by the Swiss National Science Foundation under grant 20-103420/1.
Author's address: Institut interfacultaire d'informatique, Université de Neuchâtel, Pierre-à-Mazel 7, 2000 Neuchâtel, Switzerland. E-mail: Jacques.Savoy@unine.ch

[1] See the Website at http://global-reach.biz/globstats/

A typical Chinese sentence consists of a continuous string of characters (or, more precisely, ideographs[2]) without any delimiting spaces separating them. So finding words within such a continuous string becomes a major problem, one that has to be resolved before tackling various other problems such as linguistic analysis, machine translation, or information retrieval. The Chinese language may be written using one of two main character formats. These are the traditional (usually encoded by using the BIG5 standard) and the simplified formats (using the GB standard system [Lunde 1998]), not to mention considerable orthographic variations encountered when spelling foreign names [Halpern 2002].[3] There are four writing systems in Japanese, namely the *Hiragana* syllabic character set (representing around 37.3% of the total number of characters[4]); the *Katakana* (9.7% of a syllabic character set used mainly to write words of foreign origin such as "computer," foreign names like "MacIntosh," or onomatopoeic words like "buzz"); the *Kanji* (corresponding to Chinese characters and making up 46.3% of characters used); and finally *ASCII* characters (about 6.7%, used to write numbers or company names like "Honda").

In addition to the visual differences between the European and Asian languages, there are morphological differences too [Sproat 1992]. On the basis of morphological information or word structure, the languages studied in our evaluations can be broadly grouped, based on Bloomfield's classification [1933],[5] into three different types: (1) English, Latin, and most other European languages are inflectional, within which certain distinct features are used to create single or fairly unified suffix formats added to a given stem (inflectional suffixes such as "-s" in "runs" or derivational suffixes like "-ment" in "establishment"). (2) Chinese belongs to an isolated language family in which the vast majority of words are invariable, meaning that, in IR, system-stemming procedures would play a less important role. (3) While both the Japanese or Korean languages may be considered members of the agglutinative language family in which various affixes are added to a given stem, they may also belong to a separate class that has neither a clear nor close relationship with any other language.

Given these visual and morphological differences between Indo-European and Asian languages, it is important to verify whether the efficient search models already developed for European languages will perform as well with Asian languages. The first section in this article addresses this question and is organized as follows. Section 1.1 briefly describes the various corpora in our evaluations; Section 1.2 explains the main characteristics of the nine vector-space schemes and the two probabilistic IR models; Section 1.3 presents the indexing strategies used in our experiments; Section 1.4 provides an evaluation of various indexing and search strategies; Section 1.5 evaluates a

---

[2] Also referred to as pictographs or logographs, depending on their etymology.

[3] Each natural language has some of these orthographic variations (such as "color" and "colour" in English). However, the main differences are related to homophones involving proper names. For example, Stephenson, the inventor of the steam engine, and Stevenson, the author, have the same pronunciation, and both names may be written identically in Japanese, Chinese, or Korean languages.

[4] Without counting half-width forms, punctuation, or other graphic or drawing symbols.

[5] We may also classify languages according to their word order (namely the order in which subject, verb, and object appear in a normal sentence), being SVO for the English and Chinese languages, and SOV for the Japanese and Korean languages. However, word order does not usually play an important role in various IR systems, at least not in those used in this article.

pseudo-relevance feedback approach intended to improve retrieval effectiveness; finally, Section 1.6 compares the relative merits of various data-fusion operators.

1.1 Overview of the NTCIR-4 Test-Collection

The corpora in our experiments were put together during the fourth NTCIR[6] evaluation campaign [Kishida et al. 2004a]. Created to promote the study of the information retrieval of Asian languages, this test-collection includes various newspapers written in four different languages. The English collection is taken from the *Mainichi Daily News* (Japan), *Taiwan News*, *China Times English News* (Taiwan), the *Xinhua News Service* (China), the *Korean Times,* and the *Hong Kong Standard*. The Chinese collection contains news extracted from the *United Daily News*, *China Times*, *China Times Express*, *Commercial Times*, *China Daily News,* and *Central and Daily News*. These documents were written in Mandarin using the traditional Chinese character set. The Japanese collection contains articles taken from the *Mainichi* and *Yomiuri* newspapers (Japan), while the Korean corpus was extracted from both the *Hankookilbo* and *Chosunilbo* newspapers (Korea).

Table I compares the various sizes of these corpora, ranking the Japanese collection as the largest, the English corpus as second, the Chinese corpus as third and the Korean collection as the smallest. Table I also compares the mean number of distinct bigrams per document, showing that this value is clearly larger for the Chinese collection (363.4 bigrams/article) when compared to the Korean (236.2 bigrams/article) or the Japanese corpus (114.5 bigrams/article). For the English collection, the mean number of distinct words per document is 96.6.

When analyzing the number of pertinent documents per topic, only *rigid* assessments were considered. Thus in this article only "highly relevant" and "relevant" items are seen as relevant, under the assumption that only highly relevant or relevant items are useful for all topics. In certain circumstances, however, we also assumed that records found to be only somewhat pertinent could be of some value. As a result of this rigid judgment system, the retrieval effectiveness measures depicted in this article show lower performance levels than they would with more relaxed assessments. However, we

Table I. NTCIR-4 CLIR Test-Collection Statistics (Under *Rigid* Evaluation)

	English	*Chinese*	*Japanese*	*Korean*
Size (in MB)	619 MB	490 MB	733 MB	370 MB
# of documents	347,376	381,375	593,636	254,438
Publication year	1998-1999	1998-1999	1998-1999	1998-1999
Encoding	ASCII	BIG5	EUC-JP	EUC-KR
Number of distinct indexing words or bigrams / document				
Mean	96.6	363.4	114.5	236.2
Standard deviation	61.9	219.9	97.0	146.2
Median	82	326	90	209
Maximum	2,052	5,935	5,232	3,762
Minimum	1	1	1	2
Number of topics	58	59	55	57
Number of relevant items	5,866	1,318	7,137	3,131
Mean relevant items / topic	101.138	22.339	129.764	54.930
Standard deviation	130.785	13.502	119.56	40.851

[6] See the Web site http://research.nii.ac.jp/ntcir/.

Median	35.5	19	88	43
Maximum	642 (Q#7)	61 (Q#18)	548 (Q#57)	171 (Q#9)
Minimum	5 (Q#2)	3 (Q#9)	6 (Q#4)	3 (Q#52)

Table II. Examples of Two Topics in the NTCIR-4 Test Collection

```
<TOPIC>
<NUM> 010 </NUM>
<TITLE> Hu Jintao, Visit, Japan, Korea </TITLE>
<DESC> Find articles pertaining to the activities of the Standing Committee member of
    Politburo China and Hu Jintao's visit to Japan or Korea in 1998. </DESC>
<NARR>
    <BACK> Hu Jintao, ranking 5th in the Standing Committee member of Politburo China,
    left Beijing for a visit to Japan and South Korea on April 21st, 1998. It was his first
    diplomatic visit after being elected Vice President of the National Council in China in
    March of 1998. Hu Jintao had a five-day official visit to Japan and visited South Korea
    from the 26th to the 30th. Please query important scheduled activities of Hu Jintao's
    visit. </BACK>
    <REL> Documents about reports of Hu Jintao's visit to Japan or Korea are relevant.
    Comments about this visit from other countries are not relevant. </REL> </NARR>
<CONC> Hu Jintao, Japan, South Korea, Visit, Activities </CONC> </TOPIC>

<TOPIC>
<NUM> 018 </NUM>
<TITLE> Teenager, Social Problem </TITLE>
<DESC> Find articles dealing with a teenage social problem </DESC>
<NARR>
    <BACK> As materialism appears in many aspects of society, many incidents related to
    young teenagers are becoming a major social problem. </BACK>
    <REL> Articles dealing with specific incidents or social problems related to teenagers
    (age 11 to 19) that show a summary or background story of an incident (problem) and
    information on the teenagers are relevant. Articles only addressing general criticisms on
    youth problems are irrelevant. Incidents or social problems where teenagers are
    mentioned but are not the main issue are partially relevant. </REL> </NARR>
<CONC> teenager social problem, youth problem, youth, teenager, human traffic, runaway,
    robbery, suicide, sexual abuse </CONC> </TOPIC>
```

believe that our conclusions would be similar, whether we used *rigid* or *relaxed* assessments.

A comparison of the number of relevant documents per topic, as shown in Table I, indicates that for the Japanese collection the median number of relevant items per topic is 88, while for the Chinese corpus it is only 19. By contrast, the number of relevant articles is greater for the Japanese (7,137) and English (5,866) corpora, when compared to the Korean (3,131) or Chinese (1,318) collections. These divergences may have an impact on some of our merging strategies (see Section 3).

Following the TREC model, the structure of each topic was based on four logical sections: a brief title ("<TITLE>" or T), a one-sentence description ("<DESC>" or D), a narrative part ("<NARR>" or N) specifying both the background context for the topics ("<BACK>"), a relevance assessment criteria ("<REL>"), and finally a concept section ("<CONC>" or C) that provides some related terms (see Table 2 for examples). Rather than limiting them to a narrow subject range, the topics made available were chosen to reflect a variety of information needs (such as "Viagra," "North Korea, Starvation,

Response," "Nanotechnology, Realization, Research Trends" or "Japan, Amendment, Law, Self-Defense Force").

1.2 Search Models

In order to ensure that useful conclusions are obtained when handling new test collections, we considered it important to evaluate retrieval performance under varying conditions. Thus, in order to obtain this broader view, we evaluated a variety of indexing and search models, ranging from very simple binary-indexing schemes to more complex vector-processing schemes.

First, we considered adopting a binary-indexing scheme in which each document (or topic) is represented by a set of key words, without assigning any weights (IR model denoted "document=bnn, query=bnn" or "bnn-bnn"). Binary logical restrictions may often be too restrictive for document- and query-indexing. It is also not always clear whether a document should be indexed by a given term (in this article, a single word or bigram). Given that a more appropriate answer is neither "yes" or "no," but something in between, term-weighting should allow for better differentiation of terms, and thus increase indexing flexibility. In this vein, we may also assume that the frequency with which a term occurs in a document or in a query (denoted tf) can be a useful feature (IR model denoted "nnn-nnn").

As a third weighting feature, we may consider that terms that occur very frequently in the collection do not help us discriminate between relevant and non-relevant items. For this reason we could either count their frequency in the collection or, more precisely, their inverse document frequency (denoted idf), resulting in larger weights for more specific terms and smaller weights for more frequent ones. It is important to note here that this specificity does not depend on a given term's semantic properties, but is derived from a statistical notion, or as Sparck Jones says, "we think of specificity as a function of term use" [Sparck Jones 1972]. For example, the word "computer" may be viewed as very specific in a legal corpus because it rarely appears there, whereas in a computer science collection it is viewed as a broader term, one that may have a variety of meanings.

Moreover, by using cosine normalization, whereby each indexing weight could vary in the range of 0 to 1, we could introduce a technique that usually improves retrieval effectiveness (IR model: "ntc-ntc"); see the Appendix for the exact weighting formulations for the IR models in this article.

There are also other variants that we might create, especially where a given term in a document is viewed as a rare event. Thus, it may be good practice to give more importance to the first occurrence of a term, as compared to its ensuing and repeating occurrences. Therefore, the tf component may be computed as ln(tf) + 1.0 ("ltc", "lnc", or "ltn") or as 0.5+0.5·[tf / max tf in a document] ("atn"). We might also consider that a term's presence in a shorter document represents stronger evidence than it does in a longer document. In order to take document length into account, more complex IR models have been suggested, including the "Lnu" [Buckley et al. 1996] or the "dtu" IR model [Singhal et al. 1999].

In addition to vector-space approaches, we also considered probabilistic IR models, such as the Okapi probabilistic model [Robertson et al. 2000]. We implemented the Prosit model as a second probabilistic approach (or "deviation from randomness") [Amati and van Rijsbergen 2002; Amati et al. 2003]. As shown in Eq. (1), this IR model combines two information measures. The first component measures the informative content (denoted $Inf^1_{ij}(tf)$), based on the observation that in the document D_i we found *tf* occurrences of the term t_j. The second one measures the risk (denoted $1\text{-}Prob^2_{ij}(tf)$) in

accepting the term t_j as a good descriptor, knowing that in document D_i there are *tf* occurrences of term t_j.

For the first information factor, $Prob^1_{ij}(tf)$ is the probability of observing, by pure chance, tf occurrences of the term t_j in document D_i. If this probability is high, term t_j may correspond to a noncontent-bearing word in the context of the entire collection [Harter 1975]. In the English language these words generally correspond to determinants like "the," prepositions like "with," or verb forms like "is" or "have," and are considered of little or no use in describing a document's semantic content. There are also various nouns that may often appear in numerous documents within a particular corpus, such as "computer" and "algorithm," particularly when the articles in which they are found are extracted from computer science literature. On the other hand, if $Prob^1_{ij}(tf)$ is small (or if $-\log_2[Prob^1(tf)]$ is high), the term t_j would provide important information regarding the content of the document D_i. As defined in Eq. (2), in our implementation $Prob^1(tf)$ is expressed as a geometric distribution, where $p = 1/(1+\lambda)$. Other stochastic distributions have been suggested in Amati and van Rijsbergen [2002].

The term $Prob^2_{ij}(tf)$ represents the probability of having *tf*+1 occurrences of the term t_j, since *tf* occurrences of this term have already been found in document D_i. This probability can be evaluated using the Laplace law of succession, as $Prob^2_{ij}(tf) = (tf + 1) / (tf+2) \approx tf / (tf +1)$ [Dodge 2003; p. 227]. This approximation does not take document length into account, and in our experiments we have included it as shown in Eq. (3).

$$w_{ij} = Inf^1_{ij}(tf) \cdot Inf^2_{ij}(tf) = -\log_2[Prob^1(tf)] \cdot (1 - Prob^2_{ij}(tf)) \quad (1)$$

$$Prob^1(tf) = [1/(1+\lambda_j)] \cdot [\lambda_j / (1+\lambda_j)]^{tf} \quad \text{with } \lambda_j = tc_j / n \quad (2)$$

$$Prob^2_{ij}(tf) = tfn_{ij} / (tfn_{ij} + 1) \quad \text{with } tfn_{ij} = tf_{ij} \cdot \log_2[1 + ((c \cdot mean\ dl) / l_i)] \quad (3)$$

where w_{ij} represents the indexing weight attached to term t_j in document D_i; tc_j indicates the number of occurrences of term t_j in the collection; *n* is the number of documents in the corpus; *mean dl* is the mean length of a document; and l_i the length of document D_i.

1.3 Indexing

In the previous section, we described how each indexing unit was weighted so that it reflects its importance in describing the semantic content of a document or a request. This section will explain how such indexing units are extracted from documents and topic formulations.

For the English collection, we used words as indexing units and based the indexing process on the SMART stop-word list (571 terms) and stemmers (in this case, Lovins [1968] stemming algorithm). When indexing Indo-European languages, it is natural to consider words as indexing units. For several European languages this approach has usually produced the best retrieval results, as demonstrated by various CLEF evaluation campaigns [Peters et al. 2004; 2005]. In this case, delimiting words within a sentence is a relatively easy task (with some problems, for example, "IBM360" or "test-suite" can be viewed as being made-up of either one or two words). For various languages there is a list of high-frequency or stop-list words that are usually found irrelevant when describing the semantic content of documents or queries.

Moreover, in order to conflate word variants into the same stem or root, we also need to adapt a stemming algorithm for each European language. To achieve this goal, we defined a light stemming procedure by removing only inflectional suffixes used to

indicate number (singular vs. plural), gender (feminine, masculine, or neutral), or case (nominative, genitive, ablative, locative, etc.) of a given noun or adjective.[7] Based on the CLEF 2001 test-collections, Savoy [2002] demonstrated that we can obtain mean average precision improvements of 10% (English), 15% (Italian), 18% (Spanish), 21% (German), and 24% (French) when applying a light stemmer, compared to a system that uses no stemming (T queries). With TDN queries, these improvements are less significant, ranging from 4% (English) to 10% (Spanish and German) to 14% (French and Italian).

More sophisticated stemming strategies also suggest removing certain derivational suffixes (e.g., "-ize," "-ably," "-ship" in the English language). The difference in retrieval effectiveness between light and more complex stemming approaches is usually small. In the French language, for example, Savoy [2002] shows that improvements of 5% (T queries) to 2% (TDN queries) are possible when using an extended stemming procedure.

For the Finnish language, however, it seems that the design and development of an effective stemming procedure requires a more complex morphological analysis, one based on a dictionary. For example, Tomlinson [2004] found a statistically significant difference of around 13% in favor of a dictionary-based stemmer, compared to a derivational stemmer ("Snowball" in this case). For the same language, and based on another set of queries, Moulinier and Williams [2005] confirmed this finding. The real stemming problem for Finnish is that stems are often modified when suffixes are added. For example, "*matto*" (carpet in the nominative singular form) becomes "*maton*" (in the genitive singular form, with "-n" as suffix) or "*mattoja*" (in the partitive plural form, with "-a" as suffix). Once we remove the corresponding suffixes, we are left with three distinct stems, namely "*matto*", "*mato*" and "*matoj*". Of course, irregularities such as these also occur in other languages--they usually help to make the spoken language flow better, e.g., "submit" and "submission" in English. In Finnish, however, these irregularities are more common, and thus render the conflation of various word forms into the same stem more problematic.

Finally, most European languages manifest other morphological characteristics, with compound word constructions being the most important (e.g., handgun, worldwide). Braschler and Ripplinger [2004] show that decompounding German words would significantly improve retrieval performance, resulting in improvements from 16% to 34% for T queries and 9% to 28% for TDN requests.

In order to develop a language-independent indexing strategy, McNamee and Mayfield [2004] suggest using an overlapping *n*-gram approach to define the indexing units. In this scheme, each sentence is decomposed into sequences of *n* characters. For example, when analyzing the phrase "the white house", the following 4-grams are extracted {"the_", "he_w", "e_wh", "_whi", "whit", "hite", … "hous", "ouse"}. With this type of indexing approach, stop-word lists and stemmers adapted for the corresponding language are not required, since during indexing the *n*-grams that appear in all documents (e.g., "with", "have", or very frequent suffixes like "-ment") will be assigned null, or at least insignificant, weights. According to McNamee and Mayfield [2004], and based on eight European languages, the most effective *n*-gram decomposition seems to be between 4-grams and 5-grams.

As explained previously, in the Chinese and Japanese languages words are not clearly delimited. We therefore indexed documents written in Asian languages by using an overlapping bigram approach, an indexing scheme found to be effective for various

[7] Such stop-word lists and stemmers are available for various European languages at http://www.unine.ch/clef/ or at http://snowball.tartarus.org/.

Chinese collections [Kwok 1999; Luk and Kwok 2002], or during the NTCIR-3 evaluation campaign [Chen and Gey 2003]. There are also other factors involved in our choice of an indexing tool. When considering the Korean language for example, Lee et al. [1999] found that more than 80% of Korean nouns were composed of one or two *Hangul* characters; Sproat [1992] reported a similar finding for Chinese. When analyzing the mean length of continuous characters in the Japanese corpus, we found its value to be 2.3 for *Kanji* characters, with more than 70% of continuous *Kanji* sequences composed of one or two characters. When studying the mean length of continuous *Hiragana* characters, we calculated an average value of 2.1, and for sequences composed of only *Katakana* characters, with a mean value of 3.96.

In our experiments we adopted an overlapping bigram approach. In this case, the "ABCD EFG" sequence generates the following bigrams {"AB," "BC," "CD," "EF" and "FG"}. In order to stop bigram generation, in our work we generated these overlapping bigrams for Asian characters only, using spaces and other punctuation marks (as collected for each language from their respective encodings). Moreover, we did not split any words written in ASCII characters. In our experiments the most frequent bigrams were removed before indexing. For the Chinese language, for example, we defined and removed a list of the 215 most frequent bigrams; for Japanese, 105 bigrams; and for Korean, 80 bigrams. For Chinese, we also evaluated the unigram (or character) indexing approach.

Finally, as suggested by Fujii and Croft [1993] and [Chen and Gey 2003], before generating bigrams for the Japanese documents, we removed all *Hirakana* characters, given that they are mainly used to write grammatical words (e.g., *doing*, *do*, *in*, *of*), and the inflectional endings of verbs, adjectives, and nouns.

For Asian languages there are, of course, other indexing strategies that might be used. In this vein, various authors suggest indexing Chinese documents by using words generated by a segmentation procedure (e.g., one based on the longest matching principle [Nie and Ren 1999; Foo and Li 2004]). But Nie and Ren [1999] indicated that retrieval performance based on word indexing does not really depend on an accurate word segmentation procedure; this was confirmed by Foo and Li [2004]. Nie and Ren [1999] also stated that segmenting a Chinese sentence affects retrieval performance; and recognizing a greater number of 2-character words usually contributes to enhanced retrieval. These authors did not, however, find a direct relationship between segmentation accuracy and retrieval effectiveness. Moreover, manual segmentation does not always produce better performance when compared to character-based segmentation.

For the Japanese language, Chen and Gey [2003], using the NTCIR-3 test-collection and D topics, obtained a mean average precision value of 0.2802 when combining overlapping bigrams and characters, versus 0.2758 for a word-based indexing strategy (words were segmented with the *Chasen* morphological analyzer [Matsumoto et al. 1999]). This difference in performance is small (1.6%), and seems to indicate that both indexing schemes result in similar retrieval effectiveness.

For Korean, Lee and Ahn [1996] also suggested using *n*-gram representation. In fact, even though word boundaries are marked by spaces, this language also uses numerous suffixes and even prefixes. Compound constructions are also used very frequently; a morphological analyzer can be used to separate compound words into simple nouns. Murata et al. [2003] obtained effective retrieval results using this linguistic approach. However, Lee et al. [1999] showed that *n*-gram indexing could provide similar and sometimes better retrieval effectiveness when compared to word-based indexing applied in conjunction with a decompounding scheme.

1.4 Evaluating IR Systems

Having described the various IR models, it would be useful to know how these search strategies will behave when used with the Asian test-collections. In order to measure retrieval performance we have adopted noninterpolated mean average precision (MAP), as computed by TREC-EVAL. To determine whether or not any given search strategy might be better than another, we based our statistical validation on the bootstrap approach [Savoy 1997]. Thus, in the tables in this article, statistically significant differences are shown underlined (two-sided nonparametric bootstrap test, significance level fixed at 5%). We evaluated the various IR schemes under three topic formulations: first, the queries were built using only the title (T) section; second, using the descriptive (D) section; and third, using all topic logical sections (TDNC).

The mean average precision determined by the 11 search models is shown in Table III for the English and Korean collections, with the best performance under a given condition shown in boldface type (these values were used as a baseline for our statistical tests in Tables III, IV and V). For the Japanese, Table IV depicts the performance achieved when generating bigrams from both *Kanji* and *Katakana* characters (left side), where in this case a bigram may be composed of one *Kanji* and one *Katakana* character.

Table III. MAP for Various IR Models, English and Korean Monolingual Search

	Mean Average Precision					
	English (*word,* 58 *queries*)			*Korean* (*bigram*, 57 *queries*)		
Model	T	D	TDNC	T	D	TDNC
Prosit	0.2977	0.2871	**0.3803**	0.3882	0.3010	0.4630
Okapi-npn	**0.3132**	0.2992	0.3674	0.4033	0.3475	**0.4987**
Lnu-ltc	0.3069	**0.3139**	0.3524	**0.4193**	**0.4001**	0.4857
dtu-dtn	0.2945	0.2945	0.3126	0.3830	0.3773	0.4397
atn-ntc	0.2808	0.2720	0.3417	0.3604	0.3233	0.4202
ltn-ntc	0.2766	0.2908	0.3271	0.3768	0.3494	0.4224
ntc-ntc	0.1975	0.2171	0.2559	0.3245	0.3406	0.4133
ltc-ltc	0.1959	0.2106	0.2798	0.31.3	0.3205	0.4342
lnc-ltc	0.2295	0.2421	0.3235	0.3231	0.3233	0.4616
bnn-bnn	0.1562	0.1262	0.0840	0.1944	0.0725	0.0148
nnn-nnn	0.1084	0.1013	0.1178	0.1853	0.1523	0.1711

Table IV. MAP for Various IR Models, Japanese Monolingual (55 Queries)

	Mean Average Precision					
	Bigram for Kanji/ Katakana			*Bigram for Kanji only*		
Model	T	D	TDNC	T	D	TDNC
Prosit	0.2637	0.2573	0.3442	0.2734	0.2517	0.3381
Okapi-npn	**0.2873**	**0.2821**	**0.3523**	**0.2972**	**0.2762**	**0.3510**
Lnu-ltc	0.2701	0.2740	0.3448	0.2806	0.2718	0.3397
dtu-dtn	0.2622	0.2640	0.3221	0.2739	0.2670	0.3161
atn-ntc	0.2424	0.2405	0.3303	0.2543	0.2423	0.3191
ltn-ntc	0.2735	0.2678	0.3265	0.2894	0.2730	0.3249
ntc-ntc	0.2104	0.2087	0.2682	0.2166	0.2101	0.2697
ltc-ltc	0.1868	0.1849	0.2596	0.1926	0.1881	0.2548
lnc-ltc	0.1830	0.1835	0.2698	0.1838	0.1809	0.2633
bnn-bnn	0.1743	0.1741	0.1501	0.1703	0.1105	0.0917
nnn-nnn	0.1202	0.1099	0.1348	0.1184	0.0876	0.0931

Table V. MAP for Various IR Models, Chinese Monolingual (59 Queries)

	Mean Average Precision					
	Character (*or Unigram*)			*Bigram*		
Model	T	D	TDNC	T	D	TDNC
Prosit	0.1452	0.0850	0.1486	0.1658	0.1467	0.2221
Okapi-npn	0.1667	0.1198	**0.2179**	0.1755	0.1576	0.2278
Lnu-ltc	**0.1834**	**0.1484**	0.2080	**0.1794**	**0.1609**	**0.2426**
dtu-dtn	0.1325	0.1103	0.1540	0.1527	0.1526	0.2239
atn-ntc	0.1334	0.0944	0.1699	0.1602	0.1461	0.2113
ltn-ntc	0.1191	0.0896	0.1371	0.1666	0.1556	0.2050
ntc-ntc	0.1186	0.1136	0.1741	0.1542	0.1507	0.1998
ltc-ltc	0.1002	0.0914	0.1905	0.1441	0.1430	0.2141
lnc-ltc	0.1396	0.1263	0.2356	0.1469	0.1438	0.2230
bnn-bnn	0.0431	0.0112	0.0022	0.0877	0.0781	0.0667
nnn-nnn	0.0251	0.0132	0.0069	0.0796	0.0687	0.0440

As a variant, we generated bigrams for *Kanji* characters only, with each continuous *Katakana* character sequence considered as a single indexing unit or term. Table V shows the performance for the Chinese corpus, using the unigram (or character) and bigram indexing schemes.

For Korean (right side of Table III), Japanese (Table IV), and Chinese (Table V), the best retrieval models seemed to be the Okapi or the "Lnu-ltc" search models. Surprisingly, this data shows that the best retrieval scheme for short queries was not always the same as that for long topics. For example, for long query formulations (TDNC) and for the Korean collection, the Okapi was the best search model while for short queries (T or D) the vector-space "Lnu-ltc" approach provided better performance. Based on our statistical testing, these differences in performance were not always significant (e.g., for the Japanese corpus, differences between the Okapi and "Lnu-ltc" models were only significant for T queries).

From a general perspective, it is interesting to note that when using a word-based indexing (English collection, Table III) or the *n*-gram scheme for Asian languages

Table VI. MAP for Various IR Models Using the CLEF 2003 Test-Collection (Monolingual Search, TD Queries)

	Mean Average Precision					
Model	*French word* 52 *queries*	*Spanish word* 57 *queries*	*German word* 56 *queries*	*Dutch word* 56 *queries*	*Finnish 5-gram* 45 *queries*	*Russian word* 28 *queries*
Prosit	**0.5201**	0.4723	0.4553	0.4863	**0.4903**	0.3489
Okapi-npn	0.5164	**0.4885**	**0.4693**	**0.4873**	0.4897	0.3458
Lnu-ltc	0.4826	0.4579	0.4544	0.4508	0.4603	**0.3630**
dtu-dtn	0.4658	0.4503	0.4395	0.4378	0.4354	0.3295
atn-ntc	0.4548	0.4404	0.3932	0.4352	0.4856	0.3322
ltn-ntc	0.3901	0.4240	0.3264	0.3951	0.4294	0.3089
ntc-ntc	0.3274	0.2708	0.3264	0.3036	0.3563	0.3014
ltc-ltc	0.3441	0.2974	0.3602	0.3241	0.3772	0.2874
lnc-ltc	0.3798	0.3353	0.3593	0.3315	0.3721	0.2447
bnn-bnn	0.2401	0.2648	0.2331	0.2680	0.2006	0.1523
nnn-nnn	0.1227	0.1984	0.1085	0.1064	0.1483	0.1141

(Tables III to V), the same top IR models always prove to be top performers: Okapi, Prosit, "Lnu-ltc," and "dtu-dtn". Thus, using *n*-grams or words to describe the semantic content of a document (or a request) does not result in any real performance differences among search models. This main conclusion was corroborated by other studies that compared *n*-gram and word-based indexing strategies when analyzing various European languages [McNamee and Mayfield 2004; Savoy 2002]. Table VI depicts the mean average precision obtained by using the CLEF 2003 test-collection [Peters et al. 2004] for various European languages belonging to different language groups such as the Latin family (French and Spanish), the German family (German and Dutch, evaluations included a decompounding stage), the Slavic group (Russian), and the Uralic language family (Finnish) [Savoy 2004c]. As shown in this table, the best performing IR models usually incorporate either the Okapi or the Prosit approach, showing that the performance differences between these two and the "Lnu-ltc" and "dtu-dtn" vector-space models are not always statistically significant.

As described in Section 1.2, the best IR models share three important common aspects: first, when considering the occurrence frequency of a given indexing unit in a document (tf component), the models tend to attribute more weight to the first occurrence of the term than to later occurrences; second, the idf component is also included, e.g., when weighting the search term in the Okapi or "Lnu-ltc" models (or the Prosit approach in the computation of λ_j in Eq. (2); third, and contrary to other IR models, these four best-performing search strategies take document length into account by favoring short documents (usually more focused on a narrow subject).

For the English collection, when analyzing the result language-by-language (left side of Table III), the best retrieval scheme seems to be query-dependant, and the best retrieval performance for T queries is the Okapi model, "Lnu-ltc" for D queries and Prosit for TDNC. Moreover, differences in performance for these three search models were always statistically significant. While, for English, either the Okapi or the Prosit models provide the best retrieval performance [Savoy 2004c; 2005], the good performance by the "Lnu-ltc" model using D queries must be viewed as an outlier.

For the Korean corpus (right side of Table III) and T queries, the binary indexing scheme ("bnn-bnn") resulted in a surprisingly high retrieval performance compared to the D or TDNC query formulations (0.1944, 0.0725 and 0.0148, respectively).

For the Japanese collection (Table IV), it is not clear whether bigrams should have been generated for both *Kanji* and *Katakana* characters (left side) or only for *Kanji* characters (right side of Table IV). When using title-only queries, the Okapi model provides the best mean average precision of 0.2972 (bigrams on *Kanji* only) compared to 0.2873 when generating bigrams on both *Kanji* and *Katakana*. This difference is rather small, and is even smaller in the opposite direction for long queries (0.3510 vs. 0.3523). Based on these results we cannot infer that for the Japanese language one indexing procedure is always significantly better than another.

When comparing character and bigram representations for the Chinese collection, it seems that longer queries (TDNC) tend to perform better with bigram indexing. For T or D query constructions, the difference between character and bigram indexing usually favors the bigram approach (the performance of the "Lnu-ltc" model when using T queries must be viewed as an exception). The question that arises is the following: How can we improve the retrieval effectiveness of these retrieval models? To answer this question, we suggest incorporating a blind query expansion stage during the search process (Section 1.5) and then applying a fusion strategy (Section 1.6).

1.5 Blind Query Expansion

It is known that once a ranked list of retrieved items has been computed, we can automatically expand the original query by including terms that appear frequently in the top retrieved documents. Called blind query expansion or pseudo-relevance feedback, this technique is performed before presenting the final result list to the user. In this study, we adopted Rocchio's approach [Buckley et al. 1996] with $\alpha = 0.75$, $\beta = 0.75$, whereby the system is allowed to add *m* terms extracted from the *k* best-ranked documents from the original search, as depicted in the following formula,

$$Q' = \alpha \cdot Q + \beta \cdot \frac{1}{|k|} \cdot \sum_{i=1}^{k} D_i \qquad (4)$$

in which Q' indicates the expanded query composed of the previous query Q and of *m* terms extracted from the *k* best-ranked documents D_i, assumed to be relevant to the query Q. Of course, other relevance feedback strategies have been proposed; for example, Robertson [1990] suggested making a clear distinction between the term-selection procedure and the term-weighting scheme. In a similar vein, Carpineto et al. [2001] suggested using a theoretic information measure, in this case the Kullback-Leibler divergence, for both selecting and weighting terms.

To evaluate this proposition, we used the Okapi and the Prosit probabilistic models. Table VII summarizes the best results for the English and Korean language collections; Table VIII lists the best retrieval results for the Japanese corpus (and with our two indexing strategies), as does Table IX for the Chinese collection (character or bigram indexing). In these tables, the rows labeled "Prosit" or "Okapi-npn" (baseline) indicate mean average precision before applying the blind query expansion procedure. The rows starting with "#doc. / #terms" indicate the number of top-ranked documents and number of terms used to enlarge the original query, and thus obtain the best retrieval effectiveness. Finally, the rows labeled "& Q expansion" depict the mean average

Table VII. MAP with Blind Query Expansion (English and Korean Monolingual)

	Mean Average Precision					
	English (*word*, 58 *queries*)			*Korean* (*bigram*, 57 *queries*)		
Model	T	D	TDNC	T	D	TDNC
Prosit	0.2977	0.2871	0.3803	0.3882	0.3010	0.4630
#doc. / #terms	10 / 125	10 / 75	5 / 40	5 / 20	3 / 30	10 / 75
& Q expansion	**0.3731**	**0.3513**	**0.3997**	0.4875	0.4257	0.5126
Okapi-npn	0.3132	0.2992	0.3674	0.4033	0.3475	0.4987
#doc. / #terms	10 / 20	10 / 10	10 / 20	10 /60	5 / 40	10 / 50
& Q expansion	0.3594	0.3181	0.3727	**0.4960**	**0.4441**	**0.5154**

Table VIII. MAP with Blind Query Expansion, Japanese Monolingual (55 Queries)

	Mean Average Precision					
	Bigram for Kanji/ Katakana			*Bigram for Kanji only*		
Model	T	D	TDNC	T	D	TDNC
Prosit	0.2637	0.2573	0.3442	0.2734	0.2517	0.3381
#doc. / #terms	10 / 300	10 / 100	10 /125	10 / 100	10 / 100	10 / 100
& Q expansion	**0.3396**	**0.3394**	**0.3724**	0.3495	**0.3218**	**0.3678**
Okapi-npn	0.2873	0.2821	0.3523	0.2972	0.2762	0.3510
#doc. / #terms	10 / 15	5 / 100	5 / 75	10 / 15	10 / 30	5 / 20
& Q expansion	0.3259	0.3331	0.3640	**0.3514**	0.3200	0.3561

Table IX. MAP with Blind Query Expansion, Chinese Mmonolingual (59 Queries)

	Mean Average Precision					
	Character (*or unigram*)			*Bigram*		
Model	T	D	TDNC	T	D	TDNC
Prosit	0.1452	0.0850	0.1486	0.1658	0.1467	0.2221
#doc. / #terms	10 / 125	10 / 75	3 / 10	10 / 175	10 / 100	5 / 20
& Q expansion	0.1659	0.1132	0.1624	**0.2140**	**0.1987**	**0.2507**
Okapi-npn	0.1667	0.1198	0.2179	0.1755	0.1576	0.2278
#doc. / #terms	10 / 10	10 / 10	10 / 60	5 / 125	5 / 100	5 / 60
& Q expansion	**0.1884**	**0.1407**	**0.2213**	0.2004	0.1805	0.2331

precision achieved after applying the blind query expansion (using the parameter setting specified in the previous row).

From the data in Tables VII to IX, we could infer that the blind query expansion technique improved mean average precision, and this improvement is usually statisticallysignificant (values underlined in the tables). When comparing both probabilistic models, this strategy seems to perform better with the Prosit than with the Okapi model. For some unknown reason, it seems that we must include more terms with the Prosit model than with the Okapi approach. In addition, the percentage enhancement is greater for short topics than for longer ones; for example, in the Japanese collection (bigram for both *Kanji* and *Katakana*) using the Prosit model and T topics, blind query expansion improved mean performance from 0.2637 to 0.3396 (+28.8% in relative effectiveness), compared from 0.3442 to 0.3724 (+8.5%) for TDNC topics.

Knowing that such query expansion may decrease the retrieval effectiveness for some queries, several variants are proposed. For example, Grunfeld et al. [2003] suggest using the Web to find additional search terms, while Luk and Wong [2004] suggest various term-weighting schemes, depending on the term's occurrence in the collection.

1.6 Data Fusion

As an additional strategy to enhance retrieval effectiveness, we considered adopting a data-fusion approach that combines two or more result lists provided by different search models.

By adopting this strategy we assume that different indexing and search models will retrieve different pertinent and nonrelevant items; hence combining the different search models would improve retrieval effectiveness. More precisely, when combining different indexing schemes, we expect to improve recall, due to the fact that different document representations might retrieve different pertinent items [Vogt and Cottrell 1999]. On the other hand, when combining different search schemes, we assume that the various IR strategies are more likely to rank the same relevant items higher on the list than they would the same nonrelevant documents (viewed as outliers). Thus combining them could improve retrieval effectiveness by ranking pertinent documents higher and ranking non-relevant items lower. In this study, we hope to enhance retrieval performance by making use of the second characteristic; while for Chinese, our assumption is that character and bigram indexing schemes are distinct and independent sources of evidence regarding the content of documents. Due to the first effect described above, we expect to improve recall for the Chinese language only.

As a first data-fusion strategy, we considered the round-robin approach (denoted "RR"), whereby we took, in turn, one document from all individual lists and removed duplicates, keeping the most highly ranked instances. Various other data-fusion operators have been suggested [Fox and Shaw 1994], but the simple linear combination

(denoted "SumRSV") seems to, usually, provide the best performance [Savoy 2004a; Fox and Shaw 1994]. In this case, for any given set of result lists, the combined operator is defined as SumRSV = SUM ($\alpha_i \cdot RSV_k$), in which RSV_k denotes the retrieval status value (or document score) of document D_k in the *i*th result list. Finally, the value of α_i (set to 1 for all result lists in our experiments) may be used to reflect differences in retrieval performance among the various IR models.

Given that document scores cannot, usually, be compared directly, as a third data-fusion strategy we normalized document scores within each collection by dividing them by the maximum score (i.e., the document score of the retrieved record in the first position). As a variant of this normalized score-merging scheme (denoted "NormRSV"), we might normalize the document RSV_k scores within the *i*th result list, according to

$$NormRSV_k = ((RSV_k - Min^i) / (Max^i - Min^i)) \quad (5)$$

where Min^i (Max^i) denotes the minimal (maximal) RSV value in the *i*th result list.

As a new data-fusion strategy, we suggested merging the retrieved documents according to the Z-score computed for each result list. For the *i*th result list within this scheme we needed to compute the average of the RSV_k (denoted $Mean^i$) and the standard deviation (denoted $Stdev^i$). Based on these values, we then normalized the document score for each document D_k provided by the *i*th result list, as computed using the following formula:

$$\text{Z-score } RSV_k = \alpha_i \, . \, [((RSV_k - Mean^i) / Stdev^i) + \delta^i], \text{and } \delta^i = ((Mean^i - Min^i) / Stdev^i) \quad (6)$$

where the value of δ^i is used to generate only positive values, and α_i (usually fixed at 1) is used to reflect the relative retrieval performance of the *i*th retrieval model. When the coefficients α_i are not all fixed at 1, the data-fusion operator is denoted "Z-scoreW". The use of the Z-score was also suggested for the topic-detection and tracking contexts [Leek et al. 2002].

Table X shows the mean average precision (MAP) obtained from the Chinese, Japanese, and Korean collections for each of the T, D, and TDNC queries. The top part of Table X shows the individual performances of various retrieval models in our data-fusion experiments. For example, for the T queries in Japanese, we combined the Prosit and Okapi probabilistic models with the "Lnu-ltc" and "ltn-ntc" vector-space schemes. The Chinese language data-fusion experiments also included the Okapi and "Lnu-ltc" models based on character indexing. The round-robin ("RR") scheme shown in this table was intended to serve as a baseline for our statistical testing.

From this data we can see that combining two or more IR models sometimes improves retrieval effectiveness. Moreover, linear combinations ("SumRSV") usually result in good performance, and the Z-score scheme tends to produce the best performance. As shown in Table X under the heading "Z-scoreW", we attached a weight of 2 to the Prosit model, 1.5 to the Okapi, and 1 to other IR models.

But combining separate result lists did not always enhance performance, as shown in the Korean collection using TDNC queries. In this case, none of the data-fusion operators performed significantly better than the round-robin scheme, while the best single retrieval model (Okapi in this case) was shown to have the best mean average precision (0.5141). However, it is difficult to predict exactly which data-fusion operator will produce the best results. The Z-score or the weighted Z-score schemes did seem to

Table X. MAP with Various Data-Fusion Schemes

	Mean Average Precision								
	Chinese (*bigram*/*character*) 59 *queries*			*Japanese* (55 *queries*) *bigram for Kanji/ Katakana*			*Korean* (*bigram*) 57 *queries*		
Model	T	D	TDNC	T	D	TDNC	T	D	TDNC
#doc/#term	5 / 30	10 / 100	10 / 60	10 / 200	10 / 75	10 / 350	10 / 100		3 / 30
Prosit	0.2007	0.1987	**0.2450**	0.3388	0.3390	0.3688	0.4868		0.4657
#doc/#term	5 / 100	10 / 100		5 / 10	10 / 150	10 / 150	3 / 30	5 / 20	10 / 40
Okapi-npn	0.1987	0.1758		0.3181	0.3324	0.3624	0.4654	0.4335	**0.5141**
#doc/#term	3 / 75	5 / 125		10 / 350	5 / 75	10 / 200	10 / 300		
Lnu/ltc	0.1824	0.1711		0.2879	0.2884	0. 3545	0.4500		
#doc/#term	10 / 40	5 / 60		10 / 350			5 / 15	5 / 10	
ltn-ntc	0.1780	0.1898		0.2786			0.4303	0.3946	
#doc/#term	10 / 10	3 / 10							
Okapi-npn	0.1884	0.1394	<- character indexing						
#doc/#term	3 / 75	3 / 60							
Lnu-ltc	0.1926	0.1592	<- character indexing						
RR	0.1903	0.1778		0.3283	0.3385	0.3679	0.4737	0.4260	0.5047
SumRSV	0.2103	0.1947		0.3455	0.3420	0.3739	0.5044	0.4391	0.5030
NormRSV	0.2120			0.3486	0.3444	0.3746	**0.5084**	0.4431	0.5045
Z-score	**0.2135**	0.1996		0.3498	0.3458	**0.3755**	0.5074	0.4442	0.5023
Z-scoreW	0.2120	**0.2011**		**0.3513**	**0.3484**	0.3728	0.5078	**0.4471**	0.5058

produce good results when handling different languages and query formulations. Our experiments also indicate that combining short queries resulted in better improvement than combining longer topics.

2 BILINGUAL IR

In order to retrieve information written in one of our far-east languages based on a topic description written in English, we made use of freely available resources that automatically provide translations into the Chinese, Japanese, or Korean languages. In this study we chose four different machine-translation (MT) systems and two machine-readable bilingual dictionaries (MRDs) to translate the topics, namely BabelFish, FreeTranslation, InterTran, WorldLingo, EvDict, Babylon, available at the following locations:

BabelFish	babel.altavista.com/translate
FreeTranslation	www.freetranslation.com
InterTran	www.tranexp.com:2000/InterTran
WorldLingo	www.worldlingo.com
EvDict	www.samlight.com/ev/
Babylon	www.babylon.com

When translating a topic into another language, we could also consider parallel and/or comparable corpora. Such an approach is based on document-level alignments where, in order to find terms statistically related to the target language, documents in various languages are paired according to their similarity [Braschler and Schäuble 2000]. Comparable corpora were not readily available however, so as a partial solution Nie et al. [1999] suggested using their PTMiner system to extract parallel corpora from the Web. Using these Web page collections, sentences from two pages written in two different languages were aligned using a length-based alignment algorithm [Gale and Church 1993] and the system then computed the probabilities of translating one term into

another. When using this type of statistical translation model, however, source quality (e.g., Web sites) and available corpora size are of prime importance [Nie and Simard 2001]. Cultural, thematic, and time differences could also play a role in the effectiveness of these approaches [Kwok et al. 2001].

In the absence of an explicit translation tool, Buckley et al. [1998] suggest that words in one language can be viewed as misspelled forms from another language (for example, English topics are viewed as misspelled French expressions). Following this example, Gey [2004] assumes that Chinese topics can be converted into their Japanese equivalents (after carrying out character set conversion), and hopefully some of the resulting search terms would in fact be the appropriate Japanese words. Based on the NTCIR-4 test-collection and using TDNC queries, this author obtained a MAP of 0.0893 when searching the Japanese collection for queries written in Chinese. This retrieval performance represents 25.6% of the corresponding monolingual MAP. Of course, such bilingual searches would only work when dealing with related languages, such as Italian and French or, in our context, Chinese and Japanese.

When using the Babylon bilingual dictionary, we submitted search key words word-by-word. In response to each submitted word, the Babylon system provides not only one but several translation terms (in an unspecified order). In our experiments, we decided to pick the first available translation (labeled "Babylon 1"), the first two (labeled "Babylon 2"), or the first three (labeled "Babylon 3").

Table XI shows the mean average precision obtained when translating English topics employing our two MRDs, the four MT systems, and the Okapi model. The first row ("Okapi-npn") also contains the retrieval performances of manually translated topics that will be used as a baseline. The symbol "n/a" in Table XI represents missing entries, indicating that the translation devices were not able to provide a translation for each language.

Based on the T queries and the best single query translation resource, the performance level was only 45.2% of a monolingual search for the Chinese language (0.0795 vs. 0.1755); 67.9% for Japanese (0.1952 vs. 0.2873); and 46% for Korean (0.1855 vs. 0.4033). Hence we can see that machine translation systems resulted in generally poor performance levels. Moreover, the differences in mean average precision were always statistically significant and favored manual topic translation approaches. When compared to our previous work with European languages [Savoy 2004c], the differences are clearly larger. For example, during the CLEF 2003 evaluation campaign

Table XI. MAP for Various Query Translation Approaches (Okapi Model)

	Mean Average Precision								
	Chinese (*bigram*) 59 *queries*			*Japanese* (55 *queries*) *bigram for Kanji/ Katakana*			*Korean* (*bigram*) 57 *queries*		
Model	T	D	TDNC	T	D	TDNC	T	D	TDNC
Okapi-npn	0.1755	0.1576	0.2278	0.2873	0.2821	0.3523	0.4033	0.3475	0.4987
Babylon 1	0.0458	0.0459	0.0643	0.0946	0.1255	0.1858	0.1015	0.0628	0.0706
Babylon 2	0.0441	0.0434	0.0607	0.0899	0.1202	0.1766	0.0948	0.0625	0.0660
Babylon 3	0.0473	0.0412	0.0651	0.0911	0.1172	0.1651	0.0925	0.0611	0.0627
EvDict	0.0465	0.0532	0.0753	n/a	n/a	n/a	n/a	n/a	n/a
WorldLing	0.0794	0.0702	0.1109	0.1951	**0.1972**	0.2385	0.1847	0.1745	0.2694
BabelFish	0.0795	0.0749	0.1111	0.1952	**0.1972**	0.2390	**0.1855**	**0.1768**	**0.2739**
InterTrans	n/a	n/a	n/a	0.0906	0.0888	0.1396	n/a	n/a	n/a
FreeTrans	0.0665	0.0643	0.0967	n/a	n/a	n/a	n/a	n/a	n/a
Combined	WorldLingo / EvDict			WorldLingo / Babylon 1			WorldLingo / BabelFish		
with Okapi	**0.0854**	**0.0813**	**0.1213**	**0.2174**	0.1951	**0.2550**	0.1848	**0.1768**	0.2706
withProsit	0.0817	0.0728	0.1133	0.1973	0.1897	0.2508	0.1721	0.1475	0.2409

and using the FreeTranslation MT system, we obtained 82.7% of the performance level achieved by a monolingual search for the French language (0.4270 vs. 0.5164); 80.6% for Spanish (0.3997 vs. 0.4885); and 77.4% for Italian (0.3777 vs. 0.4880) (evaluation based on English queries). Using the same MT system (FreeTranslation in this case), this comparison reveals that automatic translation from English to other Indo-European languages seems more effective than translating from English into Asian languages.

Moreover, the evaluation performances depicted in Table XI show that machine translation software tends to produce better query translation than dictionary-based approaches (namely, "Babylon" or "EvDict"). Thus automatic query translation operating within a context (topic formulation in this case) may reduce translation ambiguity. A query-by-query analysis reveals, however, that in these experiments the main underlying translation problem is related to the presence of proper nouns (e.g. "Carter", "Torrijos"), geographical terms (e.g., "South Korean") or other proper names (e.g., "Viagra"). By inspecting the Korean queries we found that, in the automatically translated queries, proper nouns were usually not translated and were written in the Latin alphabet (with some exceptions, e.g., "Michael"). Even though the machine usually returned a translation when terms were translated into Japanese, the suggested translation usually differed from the term used by humans (e.g., "South Korean", while "Apple Computer" seemed to be translated correctly). Moreover, there was no correlation between the performance of translated queries in Japanese and Korean. For example, the machine-based translated Query #7 ("Carter-Torrijos Treaty") performs reasonably well in Korean (0.9188 (bilingual search) vs. 0.9733 (monolingual)), its performance on the Japanese corpus was better than the monolingual run (0.6847 (bilingual search) vs. 0.3651 (monolingual)). This analysis seems to indicate that we need to consider introducing a supplementary stage during which the Web can be used to provide translations or at least useful related key words when handling English proper nouns. Kwok et al. [2004] were able to improve the English to Korean search by 18% when using such a technique. Chen and Gey [2003] suggested a similar approach for cases when untranslated English words (mainly proper nouns) are found. These terms were submitted to Yahoo!Chinese (or Yahoo!Japan) and the first 200 entries were then downloaded and segmented into words. After this step, from each line containing the specific English word, they extracted the five Chinese words immediately to the left and to the right of the English word and included them in the translated topic (assigning a weight $1/k$, with $k = 1$ to 5, to represent the distance between the Chinese and English words).

Looking at the results language-by-language, it seems that the BabelFish MT system tends to produce the best translations of topics in Japanese and Korean, and both BabelFish and WorldLingo MT for Chinese. In order to improve retrieval performance, we developed three possible strategies. First, we concatenated the output of two translation tools into a single query. For Chinese, we combined the translations given by WorldLingo with those of "EvDict"; for Japanese we concatenated the translations provided by WorldLingo with those of "Babylon 1"; and for Korean, we combined WorldLingo and BabelFish. As shown in the last two rows of Table XI, the combined translation strategy seems to enhance retrieval effectiveness for Chinese and Japanese, but not for Korean.

In a second attempt to improve performance, we applied a blind query expansion to the combined translated topics. As shown in Table XII, this technique clearly enhanced retrieval effectiveness when we used the Okapi or Prosit probabilistic models. As for monolingual IR (see Tables VII to IX) and for the Chinese and Japanese collections, the results achieved by the Prosit system after pseudo-relevance feedback were usually better

than those obtained by the Okapi search model. Surprisingly, for T queries in the Japanese corpus, the Okapi combined with blind query expansion achieved a performance level of 0.2733 (or 95.1% of the monolingual performance, however without blind query expansion). When compared to other bilingual runs, blind query feedback seems to be a very attractive strategy for enhancing retrieval effectiveness.

As a third strategy for enhancing retrieval effectiveness, we might consider adopting a data-fusion approach that combines two or more result lists provided by different search models (as shown with the monolingual search; see Section 1.6).

As an additional strategy, it would be useful to know or predict when a given translation is good or when a given search might produce a proper response. In this vein, Kishida et al. [2004b] suggest using a linear regression model to predict the average precision of the current query, based on both manual evaluations of translation quality for the current query and the underlying topic difficulty. Using the 55 queries written in Japanese, together with their machine-based translations from Korean, Chinese, and English, these authors found that the 64% variability in average performance was due to both translation quality and intrinsic query difficulties. In a related paper, however, Cronen-Townsend et al. [2002] showed that, in monolingual IR, a query's idf average value might adequately predict its retrieval effectiveness or intrinsic difficulty. Based on such findings, it may be worthwhile to combine various translations on a per-query basis or to select the most appropriate parameters when expanding the original query, also on a per-query basis.

3. MULTILINGUAL INFORMATION RETRIEVAL

In this section we will investigate situations in which users write a topic in English in order to retrieve relevant documents in English, Chinese, and Japanese (CJE) or in English, Chinese, Japanese, and Korean (CJKE). To deal with this multilanguage hurdle, we based our approach on bilingual IR systems, as described in the previous section. Thus, the various collections were indexed separately, and once the original requests were received, they were translated into different languages and submitted to the various collections or search engines. As a response, a ranked list of retrieved items was returned from each collection. From these lists we needed to produce a unique ranked result list, using the merging strategy described further on in this section. Moreover, in our multilingual experiments, only one search engine would be available, which is a common situation in digital libraries or in other office environments. We wanted to compare our various merging strategies via a good general search engine, so we selected the Prosit model. Based on the same test-collection, Savoy [2004b] evaluated various multilingual merging strategies by using a variety of search engines.

Table XII. MAP for Blind Query Expansion on Translated Queries (Okapi or Prosit)

	Mean Average Precision								
	Chinese (*bigram*) 59 *queries*			*Japanese* (55 *queries*) *bigram for Kanji/ Katakana*			*Korean* (*bigram*) 57 *queries*		
Model	T	D	TDNC	T	D	TDNC	T	D	TDNC
Okapi-npn	0.0854	0.0813	0.1213	0.2174	0.1951	0.2550	0.1848	0.1768	0.2706
#doc/#term	5 / 60	5 / 60	5 / 75	10 / 75	5 / 75	5 / 75	5 / 75	10 / 200	5 / 60
& Q exp	0.1039	0.1003	0.1290	**0.2733**	0.2185	0.2669	**0.2397**	**0.2139**	0.2882
Prosit	0.0817	0.0728	0.1133	0.1973	0.1897	0.2508	0.1721	0.1475	0.2409
#doc/#term	5 / 40	10 / 125	5 / 60	10 / 200	10 / 100	10 / 200	10 / 125	10 / 125	10 / 100
& Q exp.	**0.1213**	**0.1057**	**0.1644**	0.2556	**0.2600**	**0.3065**	0.2326	0.2098	**0.2968**

Other search strategies have, of course, been suggested for handling multilingual collections. For example, as an alternative to the query translation approach, we might translate all documents into a single common language [Braschler and Peters 2004; Chen and Gey 2004]. In such a case we might form a huge unique collection with all available documents, and, since the search would be performed by comparing to a single collection, no merging procedure is required. As shown in the CLEF 2003 evaluation campaign, such an indexing and search strategy usually provides very good retrieval effectiveness. The document translation approach does, however, require more computational effort; if we allow users to write their queries in *k* languages, we need to translate each document into *k-1* other languages.

We adopted a query translation strategy and then, after performing a search on each language, we merged the different result lists. The top part of Table XIII illustrates a merging problem in which a query has been sent to three collections. In response, three result lists were received, and so we had to merge the retrieved items in order to form a unique ranked list, one that reflects the degree of pertinence of each item within the request.

As a first merging approach, we considered the round-robin method [Voorhees et al. 1995], whereby we took one document in turn from all individual lists. In this case, we might assume that each collection (or language in this study) contains approximately the same number of pertinent items and that the distribution of relevant documents is similar across the result lists. Under these hypotheses, the rank of the retrieved documents would be the key feature in generating the final unique result list presented to the user.

As a second approach, and in order to account for the document score computed for each retrieved item (denoted RSV_k for document D_k), we might formulate the hypothesis that each collection could be searched by the same, or a very similar, search engine, and hence that the similarity values would be directly comparable. Such a strategy, called raw-score merging, produced a final list sorted by document score, as computed by each collection. Since we used the same retrieval model (Prosit) for searching within all collections separately, we could expect resulting document scores to be more comparable, and thus the document score could be used to sort the retrieved items. However, the document scores were not always comparable, and this merging strategy favors documents with a high retrieval status value from the Japanese or Korean corpus, as illustrated in Table XIII.

Table XIII. Example of Three Merging Strategies

Japanese Collection			*Chinese Collection*			*Korean Collection*		
rank	document	RSV	rank	document	RSV	rank	document	RSV
1	JP015	90	1	ZH167	0.75	1	KR785	60
2	JP256	88	2	ZH572	0.45	2	KR178	54
3	JP678	50	3	ZH719	0.39	3	KR710	51
4	JP961	45	4	ZH739	0.38	4	KR389	30
5	JP178	44	5	ZH078	0.35	5	KR781	29
…	…	…	…	…	…	…	…	…
Round-robin			Raw-score			MaxRSV		
1	JP015		1	JP015	90	1	JP015	1.00
2	ZH167		2	JP256	88	2	ZH167	1.00
3	KR785		3	KR785	60	3	KR785	1.00
4	JP256		4	KR178	54	4	JP256	0.98
5	ZH572		5	KR710	51	5	KR178	0.90
6	KR178		6	JP678	50	6	KR710	0.85
7	JP678		7	JP961	45	7	ZH572	0.60

As demonstrated by Dumais [1994], however, collection-dependent statistics represented by document or query weights may vary widely among collections; this phenomenon may therefore invalidate the raw-score merging hypothesis, even when the same search engine is used. Thus, as a third scheme, we could normalize the RSV_k by using the retrieved record document score listed in the first position ("MaxRSV") or by using eq. (5) ("NormRSV"). Under these merging strategies, we assume that document scores computed by search engines working with different corpora are not comparable. Therefore, these document scores must be normalized before they can be used as keys to sort the retrieved items. As depicted in Table XIII, such a merging strategy would account for the difference between a given document score and the document score for the first retrieved item provided by the same collection. In our example, the difference between the first and the third item in the Korean collection is relatively small compared to the difference between the first and the second document in the Chinese collection. Therefore, the third document of the Korean corpus "KR710" must appear before the second document extracted form the Chinese corpus "ZH572".

As a fifth merging scheme, we suggest a biased round-robin approach, which extracts not just one document per collection per round, but one document from both the English and Chinese collections and two from the Japanese and Korean. A merging strategy such as this exploits the fact that the Japanese and Korean corpora possess more articles than the English or the Chinese collections (see Table 1). So we may assume that the Japanese or Korean corpus will contain more pertinent information than the English or Chinese collection.

As a sixth merging approach, we could use our Z-score model (see Section 1.6 and Eq. (6)) to define a comparable document score across the collections. This merging strategy would exploit the fact that the top-ranked retrieved and pertinent items usually provide much greater RSV values than do the others, and such documents must be presented to the user. Manmatha et al. [2001] propose a similar idea when modeling the document score distribution in the form of a mixture model. On the other hand, when the document scores from a given result list are all more or less the same, we must consider that such a distribution contains a very large number of irrelevant documents.

In this merging strategy, we may also consider that each collection may have different numbers of pertinent items or that each collection is searched by IR models having different mean retrieval performances. To reflect this bias when using a given collection or search engine, we could multiply each normalized document score by a corresponding weight. Using this idea, under the label "Z-scoreW" we assigned a weight of 1.2 to the Japanese and Korean result lists and 1.0 to the English and Chinese runs. In this study, we increased the weight attached to the Japanese and Korean languages because these collections contained more documents, and hopefully more relevant documents.

Finally, we could use logistic regression to predict the probability of a binary variable outcome, according to a set of explanatory variables [Le Calvé and Savoy 2000]. In our current case, we predicted the probability that document D_k would be relevant, given both the logarithm of its rank (indicated by $\ln(rank_k)$) and the original document score RSV_k as in Eq. (7). Based on these estimated relevance probabilities (computed independently for each language using S+ software), we sorted the records retrieved from separate collections in order to obtain a single ranked list.

$$\text{Prob}\left[D_k \text{ is rel} \mid \text{rank}_k, \text{rsv}_k\right] = \frac{e^{\alpha+\beta_1\cdot\ln(\text{rank}_k)+\beta_2\cdot\text{rsv}_k}}{1+e^{\alpha+\beta_1\cdot\ln(\text{rank}_k)+\beta_2\cdot\text{rsv}_k}} \quad (7)$$

But in order to estimate the underlying parameters, this approach requires that a training set be developed. In our evaluations we did this by using the leaving-one-out approach

to produce an unbiased estimate of the real performance. In this case, the training set was made up of all queries except one; this last request was used to compute the average precision for this single query. Finally, we iterated them over the query samples, generating 60 different training sets (composed of 59 queries) and 60 query evaluations from which a mean average precision could be computed.

Table XIV shows the retrieval effectiveness of the various merging strategies when English queries are translated automatically. The top part of this table shows the mean average precision obtained independently for each language, using the Prosit search model along with query expansion (number of top-ranked documents / number of additional search terms). The middle part depicts the mean average precision when searching the Chinese, Japanese, and English collections (CJE), while the bottom part also includes the Korean language (CJKE). In this table and for both multilingual environments, the round-robin merging strategy serves as a baseline upon which statistical tests can be performed. Finally in Table XV, we evaluated multilingual runs using manually translated topics. As depicted in Tables XIV and XV, we could then estimate retrieval effectiveness due to automatic query translation strategies. Moreover, the experiments shown in Table XV may be used to confirm the findings in Table XIV.

The data in Table XIV indicates that only a few runs produced retrieval effectiveness that might be viewed as statistically superior to that of the round-robin baseline. When considering manually translated queries as shown in Table XV, more merging strategies resulted in significantly better performance than the round-robin scheme did.

Table XIV. MAP of Various Merging Strategies for the CJE and CJKE Collections with Automatic Query Translation

	Mean Average Precision		
	T	D	TDNC
English (on 58 queries)	Prosit 10/125	Prosit 10/75	Prosit 5/40
	0.3731	0.3513	0.3997
Chinese (on 59 queries)	Prosit 5/40	Prosit 10/125	Prosit 5/60
Lingo & Ed	0.1213	0.1057	0.1644
Japanese (on 55 queries)	Prosit 10/200	Prosit 10/100	Prosit 10/200
Lingo & Babylon 1	0.2556	0.2600	0.3065
Korean (on 57 queries)	Prosit 10/125	Prosit 10/125	Prosit 10/100
Lingo & BabelFish	0.2326	0.2098	0.2968
Merging strategy on CJE			
Round-robin (baseline)	0.1591	0.1554	0.2040
Raw-score	0.1573	0.1467	0.1914
MaxRSV	0.1671	0.1614	0.2072
NormRSV (Eq. 5)	0.1660	0.1646	0.2129
Biased round-robin (J=2)	0.1657	0.1632	0.2116
Z-score (Eq. 6)	0.1625	0.1613	0.2096
Z-scoreW (Eq. 6) (J=1.2)	0.1673	0.1662	0.2156
Logistic regression	**0.1978**	**0.1917**	**0.2363**
Merging strategy on CJKE			
Round-robin (baseline)	0.1394	0.1343	0.1870
Raw-score	0.1381	0.1292	0.1740
MaxRSV	0.1354	0.1296	0.1718
NormRSV (Eq. 5)	0.1407	0.1379	0.1871
Biased round-robin (J=K=2)	0.1412	0.1358	0.1885
Z-score (Eq. 6)	0.1406	0.1397	0.1941
Z-scoreW (Eq. 6) (J=K=1.2)	0.1430	0.1421	0.1970
Logistic regression	**0.1676**	**0.1630**	**0.2187**

As a first approach, both simple and normalized merging schemes ("MaxRSV" and "NormRSV") provide reasonable performance levels, with the "NormRSV" merging scheme being slightly better. In our experiments, while the retrieval effectiveness of the raw-score approach was not very good, decreases in performance were usually not statistically significant compared to the round-robin scheme (except for manually translated queries and CJKE search, as shown in the bottom part of Table XV). Our biased round-robin scheme seems to perform better when compared to the simple round-robin version, yet it is difficult *a priori* to know whether any given corpus will

Table XV. MAP of Various Merging Strategies Applied to the CJE and CJKE Collections with Manual Query Translation

	Mean Average Precision		
	T	D	TDNC
English (on 58 queries)	Prosit 10/125 0.3731	Prosit 10/75 0.3513	Prosit 5/40 0.3997
Chinese (on 59 queries)	Prosit 10/175 0.2140	Prosit 10/100 0.1987	Prosit 5/20 0.2507
Japanese (on 55 queries)	Prosit 10/300 0.3396	Prosit 10/100 0.3394	Prosit 10/125 0.3724
Korean (on 57 queries)	Prosit 5/20 0.4875	Prosit 3/30 0.4257	Prosit 10/75 0.5126
Merging strategy on CJE			
Round-robin (baseline)	0.2230	0.2139	0.2505
Raw-score	0.2035	0.1981	0.2364
MaxRSV	0.2222	0.2180	0.2541
NormRSV (Eq. 5)	0.2281	0.2195	0.2560
Biased round-robin (J=2)	0.2345	0.2260	0.2624
Z-score (Eq. 6)	0.2293	0.2243	0.2620
Z-scoreW (Eq. 6) (J=1.2)	0.2351	0.2320	0.2716
Logistic regression	**0.2505**	**0.2396**	**0.2827**
Merging strategy on CJKE			
Round-robin (baseline)	0.2305	0.2157	0.2636
Raw-score	0.1913	0.1879	0.2430
MaxRSV	0.2210	0.2038	0.2645
NormRSV (Eq. 5)	0.2305	0.2139	0.2674
Biased round-robin (J=K=2)	0.2393	0.2234	0.2734
Z-score (Eq. 6)	0.2395	0.2273	0.2770
Z-scoreW (Eq. 6) (J=K=1.2)	0.2462	0.2361	0.2866
Logistic regression	**0.2549**	**0.2422**	**0.2981**

Table XVI. Inverted File and Search Statistics (NTCIR-4 Test-Collection)

	English	*Chinese*	*Japanese*	*Korean*
# postings	524,788	2,704,517	804,801	320,431
Inverted file size	385 MB	1,187 MB	650 MB	530 MB
Building time	454.5 sec.	1,116.2 sec.	578.7 sec.	446.1 sec.
T queries				
Mean query size	4.25 wd/query	5.8 bi/query	6.35 bi/query	5.58 bi/query
Search time per query	0.23 sec.	0.183 sec.	0.287 sec.	0.187 sec.
TDNC queries				
Mean query size	34.25 wd/query	116.4 bi/query	28.7 bi/query	101.4 bi/query
Search time per query	0.433 sec.	0.452 sec.	0.492 sec.	0.56 sec.

actually contain more relevant items than another. In this study we assume that the number of documents in a given collection is correlated with the number of relevant items contained in this corpus. Both the Z-score and the weighted Z-score (with $\alpha = 1$ for the English and Chinese corpora, and 1.2 for both the Japanese and Korean languages) usually achieved better performance levels than the round-robin approach (performance differences were not, however, always statistically significant, at least in Table XIV). For all multilingual searches, our logistic merging scheme produced the best mean average precision, and was always statistically superior to the round-robin approach. As a second-best approach, Tables XIV and XV indicate that our weighted Z-score merging scheme always produced the second-best retrieval performance. Translated queries were relatively significant. For the CJE multilingual retrieval and T queries, the best automatic run had a mean average precision rate of 0.1978 compared to 0.2505 (or a 21% difference in relative performance). When compared with the CJKE multilingual search and T queries, the difference was greater (0.1676 vs. 0.2549, or 34.2%).

In addition to retrieval effectiveness, it would be worthwhile to obtain an overview of computational efforts required to build and search these test collections. The top part of Table XVI lists the size of each collection in terms of storage space requirements and number of documents. The "# postings" row indicates the number of terms (words for the English corpus, bigrams for the three Asian languages) in the inverted file. The next row shows the inverted file size and the following row depicts the time (user CPU time + system CPU time) needed to build this inverted file. The other rows show the average query size and search time (in seconds) required for both short (T) and long (TDNC) queries (measured without blind query expansion).

To implement and evaluate the various search models, we used an Intel Pentium III/600 (memory: 1 GB, swap: 2 GB, disk: 6 x 35 GB).

CONCLUSION

Successful access to multilingual document collections requires an effective monolingual indexing and search system, a combined query translation approach, and a simple but efficient merging strategy [Braschler and Peters 2004; Chen and Gey 2004; Savoy 2004a]. Using this blueprint, derived during the latest CLEF evaluation campaigns, we effectively applied it to the three far-east Asian languages. Thus, as a result of our evaluations when indexing Asian languages based on bigrams, the “Lnu-ltc” vector space or the Okapi probabilistic IR models (see Tables III to V) achieve the best retrieval performance levels. Blind-query expansion has proven to be a worthwhile approach, especially when processing short queries and using the Prosit IR model (see Tables VII though IX). In order to further improve retrieval effectiveness, a data-fusion approach could be considered, although this technique would require additional computational resources (see Table X).

Based on our analysis of bilingual search performances, our results conflicted with those for certain European languages [Savoy 2004a; 2004c], especially given the number and questionable quality of freely available translation resources. Thus, when compared with corresponding monolingual searches in which we translated user information from English into Chinese, Japanese, or Korean languages, overall retrieval effectiveness decreases more than 30% for the Japanese, and more than 50% for the Chinese and Korean languages (see Table XI). To improve this poor performance, we could concatenate two (or more) translations (see the last two rows of Table XI), employ a blind query expansion approach (see Table XII), and a data-fusion approach.

When evaluating various merging strategies using different query sizes, it appears that when merging ranked lists of retrieved items provided by separate collections, good retrieval effectiveness is obtained with the Z-score merging procedure (around 5% better that the round-robin approach). When a representative query sample is available, however, the logistic merging scheme always produces the best retrieval effectiveness (between 10% (CJKE, manual query translation, T queries) to 24% (CJE, automatic query translation, T queries) better that the round-robin approach).

ACKNOWLEDGMENTS

The author would like to thank the task NTCIR-4 CLIR organizers for their efforts in developing the Asian test-collections and the anonymous referees for their useful suggestions and comments. The author would also like to thank C. Buckley from SabIR for giving us the opportunity to use the SMART system, together with Samir Abdou and Pierre-Yves Berger for their help in translating the English topics. This research was supported in part by the Swiss National Science Foundation under Grant #200020-103420.

APPENDIX

In Table A.1, w_{ij} represents the indexing weight assigned to term t_j in document D_i. To define this value, we use n to indicate the number of documents in the collection and nt_i the number of distinct indexing units (bigrams or terms) included in the representation of D_i. We assigned values to the constant b as follows: 0.5 for both the Chinese and Japanese corpora; 0.55 for the English; and 0.75 for the Korean. While we fixed the constant k_1 at 1.2, *avdl* at 500, *pivot* at 100, and the *slope* at 0.1. For the Prosit model, we assigned $c = 2$ for the Japanese and Korean corpus; $c = 1$ for the English; and $c = 1.5$ for the Chinese. These values were chosen because they usually result in better retrieval performance levels. Finally, the value "*mean dl"* was fixed at 151 for the English, 480 for the Chinese, 144 for the Japanese, and 295 for the Korean corpus.

Table A.1. Weighting Schemes

bnn	$w_{ij} = 1$	npn	$w_{ij} = tf_{ij} \cdot \ln[(n-df_j)/ df_j]$	ltn	$w_{ij} = (\ln(tf_{ij}) + 1) \cdot idf_j$
nnn	$w_{ij} = tf_{ij}$	lnc	$w_{ij} = \frac{\ln(tf_{ij}) + 1}{\sqrt{\sum_{k=1}^{t} (\ln(tf_{ik}) + 1)^2}}$	ntc	$w_{ij} = \frac{tf_{ij} \cdot idf_j}{\sqrt{\sum_{k=1}^{t} (tf_{ik} \cdot idf_k)^2}}$
atn	$w_{ij} = idf_j \cdot [0.5 + 0.5 \cdot tf_{ij} / maxtf_i]$			dtn	$w_{ij} = [\ln(\ln(tf_{ij}) + 1) + 1] \cdot idf_j$
Lnu	$w_{ij} = \frac{\left(1 + \ln(tf_{ij})\right) / \left(\ln(\text{mean tf}) + 1\right)}{(1 - \text{slope}) \cdot \text{pivot} + \text{slope} \cdot nt_i}$			ltc	$w_{ij} = \frac{\left(\ln(tf_{ij}) + 1\right) \cdot idf_j}{\sqrt{\sum_{k=1}^{t} \left(\left(\ln(tf_{ik}) + 1\right) \cdot idf_k\right)^2}}$
Okapi	$w_{ij} = \left((k_1 + 1) \cdot tf_{ij}\right) / \left(K + tf_{ij}\right)$			dtu	$w_{ij} = \frac{\left(\ln\left(\ln(tf_{ij}) + 1\right) + 1\right) \cdot idf_j}{(1 - \text{slope}) \cdot \text{pivot} + \left(\text{slope} \cdot nt_i\right)}$

REFERENCES

AMATI, G. AND VAN RIJSBERGEN, C. J. 2002. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans Information Systems 20*, 4 (2002), 357-389.

AMATI, G., CARPINETO, C., AND ROMANO, G. 2003. Italian monolingual information retrieval with PROSIT. In *Advances in Cross-Language Information Retrieva.l* C. Peters et al. (eds.), Lecture Notes in Computer Science, 2785, Springer-Verlag, Berlin, 257-264.

BLOOMFIELD, L. 1933. *Language*. Holt, Rinehart and Winston, New York.

BUCKLEY, C., SINGHAL, A., MITRA, M., AND SALTON, G. 1996. New retrieval approaches using SMART. In *Proceedings of the TREC-4 Conference* (Gaithersburg, MD, Nov. 1995). D.K. Harman (ed), NIST Special Publication 500-236, 25-48.

BUCKLEY, C., MITRA, M., WALTZ, J., AND CARDIE, C. 1998. Using clustering and superconcepts within SMART. In *Proceedings of the TREC-6 Conference* (Gaithersburg, MD, Nov.1997). E.M. Voorhees and D.K. Harman (eds), NIST Special Publication 500-240, 107-124.

BRASCHLER, M. AND SCHÄUBLE, P. 2000. Using corpus-based approaches in a system for multilingual information retrieval. *IR Journal 3*, 3 (2000), 273-284.

BRASCHLER, M. AND PETERS, C. 2004. Cross-language evaluation forum: Objectives, results and achievements. *IR Journal 7,* 1-2 (2004), 7-31.

BRASCHLER, M. AND RIPPLINGER, B. 2004. How effective is stemming and decompounding for German text retrieval? *IR Journal 7*, 3-4 (2004), 291-316.

CARPINETO, C., DE MORI, R., ROMANO, G.. AND BIGI, B. 2001. An information-theoretic approach to automatic query expansion. *ACM Trans. Information Systems 19*, 1 (2001), 1-27.

CHEN, A. AND GEY, F.C. 2003. Experiments on cross-language and patent retrieval at NTCIR-3 workshop. In *Proceedings of the NTCIR-3 Conference* (Tokyo). N. Kando (ed.), 2003.

CHEN, A. AND GEY, F.C. 2004. Multilingual information retrieval using machine translation, relevance feedback, and decompounding. *IR Journal 7*, 1-2 (2004), 149-182.

CRONEN-TOWNSEND, S., ZHOU, Y., AND CROFT, W.B. 2002. Predicting query performance. In *Proceedings of the 25th International Conference on Research and Development in Information Retrieval*. K. Jarvelin et al. (eds.). ACM, New York, 299-306.

DUMAIS, S.T. 1994. Latent semantic indexing (LSI) and TREC-2. In *Proceedings of the TREC-2 Conference* (Gaithersburg, MD, Sept. 1993). D.K. Harman (ed.), NIST Special Publication 500-215, 105-115.

DODGE, Y. (ED.) 2003. *The Oxford dictionary of Statistical Terms*. Oxford University Press, Oxford, UK.

FOO, S. AND LI, H. 2004. Chinese word segmentation and its effect on information retrieval. *Information Process. Manage. 40*, 1 (2004), 161-190.

FOX, E.A. AND SHAW, J.A. 1994. Combination of multiple searches. In *Proceedings of the TREC-2 Conference* (Gaithersburg, MD, Sept. 1993). D.K. Harman (ed.), NIST Special Publication 500-215, 243-249.

FUJII, H. AND CROFT, W.B. 1993. A comparison of indexing techniques for Japanese text retrieval. In *Proceedings of the 16th International Conference on Research and Development in Information Retrieval*. ACM, New York, 237-246.

GALE, W. A. AND CHURCH, K. W. 1993. A program for aligning sentences in bilingual corpora. *Computational Linguistic 19*, 1 (1993), 75-102.

GEY, F. 2004. Chinese and Korean topic search of Japanese news collections. In *Working Notes of NTCIR-4*, N. Kando (ed.), Tokyo, June 2004, 214-218.

GRUNFELD, L., KWOK, K. L., DINSTL, N., AND DENG, P. 2004. TREC2003 robust, HARD and QA track experiments using PIRCS. In *Proceedings of the TREC-12 Conference* (Gaithersburg, MD, Nov. 2003). E.M. Voorhees and D.K. Harman (eds), NIST Special Publication 500-255, 510-521.

HALPERN, J. 2002. Lexicon-based orthographic disambiguation in CJK intelligent information retrieval. In *Proceedings of COLING-02: The 3rd Workshop on Asian Language Resources and International Standardization.*

HARTER, S. P. 1975. A probabilistic approach to automatic keyword indexing. Part I: On the distribution of specialty words in a technical literature. *J. American Association for Information Science 26* (1975), 197-216.

KISHIDA, K., CHEN, K. H., LEE, S., KURIYAMA, K., KANDO, N., CHEN, H. H., MYAENG, S. H., AND EGUCHI, K. 2004a. Overview of CLIR task at the Fourth NTCIR Workshop. In *Working Notes of NTCIR-4*. N. Kando (ed.), Tokyo, June 2004, 1-59.

KISHIDA, K., KURIYAMA, K., KANDO, N., AND EGUCHI, K. 2004b. Prediction of performance on cross-lingual information retrieval by regression models. In *Working Notes of NTCIR-4*. N. Kando (ed.), Tokyo, June 2004, 219-224.

KWOK, K. L., GRUNFELD, L., DINSTL, N., AND CHAN, M. 2001. TREC-9 Cross-language, Web and question-answering track experiments using PIRCS. In *Proceedings of the TREC-9 Conference* (Gaithersburg, MD, Nov, 2000). E.M. Voorhees and D.K. Harman (eds). NIST Special Publication 500-249, 417-426.

KWOK, K. L. (1999). Employing multiple representations for Chinese information retrieval. *J. American Society for Information Science 50*, 8 (1999), 709-723.

KWOK, K. L., DINSTL, N., AND CHOI, S. 2004. NTCIR-4 Chinese, English, Korean cross-language retrieval experiments using PIRCS. In *Working Notes of NTCIR-4*. N. Kando (ed.), Tokyo, June 2004, 186-192.

LE CALVÉ, A. AND SAVOY, J. 2000. Database merging strategy based on logistic regression. *Information Process. Manage. 36*, 3 (2000), 341-359.

LEE, J. H. AND AHN, J. S. 1996. Using n-grams for Korean text retrieval. In *Proceedings of the 19th International Conference on the ACM-SIGIR'96*). H. P. Frei et al. (eds.). ACM Press, New York, 216-224.

LEE, J. J., CHO, H.Y., AND PARK, H. R. 1999. N-gram-based indexing for Korean text retrieval. *Information Process. Manage. 35*, 4 (1999), 427-441.

LEEK, T., SCHWARTZ, R., AND SRINIVASA, S. 2002. Probabilistic approaches to topic detection and tracking. In *Topic Detection and Tracking: Event-based Information Organization*. J. Allan (ed.). Kluwer, Boston, MA, 67-83.

LOVINS, J. B. 1982. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics 11*, 1 (1982), 22-31.

LUK, R.W.P. AND KWOK, K. L. 2002. A comparison of Chinese document indexing strategies and retrieval models. *ACM Trans. Asian Language Information Process. 1*, 3 (2002), 225-268.

LUK, R.W.P. AND WONG, K. F. 2004. Pseudo-relevance feedback and title re-ranking for Chinese information retrieval. In *Working Notes of NTCIR-4*. N. Kando (ed.).Tokyo, June 2004, 206-213.

LUNDE, K. 1998. *CJKV Information Processing. Chinese, Japanese, Korean & Vietnamese Computing*. O'Reilly, New York.

MANMATHA, R., RATH, T., AND FENG, F. 2001. Modeling score distributions for combining the outputs of search engines. In *Proceedings of the 24th International Conference on the ACM-SIGIR'2001*. D. H. Kraft et al. (eds). ACM, New York, 267-275.

MATSUMOTO, Y., KITAUCHI, A., YAMASHITA, T., HIRANO, Y., MATSUDA, H., AND ASAHARA, M. 1999. Japanese morphological analysis system ChaSen. Tech. Rep. NAIST-IS-TR99009, NAIST. http://chasen.aist-nara.ac.jp/

MCNAMEE, P. AND MAYFIELD, J. 2004. JHU/APL experiments in tokenization and non-word translation. In *Comparative Evaluation of Multilingual Information Access Systems*. C. Peters et al. (eds.). Lecture Notes in Computer Science 3237. Springer-Verlag, Berlin, 85-97.

MOULINIER, I.. AND WILLIAMS, K. 2005. Report on Thomson legal and regulatory experiments at CLEF 2004. In *Advances in Cross-Language Information Retrieval*. C. Peters et al. (eds.). Lecture Notes in Computer Science 3491 Springer-Verlag, Berlin, 110-122.

MURATA, M., MA, Q., AND ISAHARA, H. 2003. Applying multiple characteristics and techniques to obtain high levels of performance in information retrieval. In *Proceedings of the NTCIR-3 Conference* (Tokyo). N. Kando (ed.).

NIE, J. Y. AND REN, F. 1999. Chinese information retrieval: using characters or words? *Information Process. Manage. 35*, 4 (1999), 443-462.

NIE, J. Y., SIMARD, M., ISABELLE, P., AND DURAND, R. 1999. Cross-language information retrieval based on parallel texts and automatic mining of parallel texts from the Web. In *Proceedings of the 22nd International Conference of the ACM-SIGIR'99*. M. Hearst et al. (eds). ACM, New York, 74-81.

NIE, J. Y. AND SIMARD, M. 2001. Using statistical translation models for bilingual IR. In *Evaluation of Cross-language Information Retrieval Systems*. C. Peters et al. (eds.). Springer-Verlag, Berlin, 137-150.

PETERS, C., BRASCHLER, M., GONZALO, J., KLUCK, M. (eds.). 2004. *Advances in Cross-Language Information Retrieval*. Lecture Notes in Computer Science 2785, Springer-Verlag, Berlin.

PETERS, C., CLOUGH, P., GONZALO, J., JONES, G., KLUCK, M., MAGNINI, B. (eds.). 2005. *Advances in Cross-Language Information Retrieval*. Lecture Notes in Computer Science 3491, Springer-Verlag, Berlin.

ROBERTSON, S. E. 1990. On term selection for query expansion. *J. Documentation 46*, 4 (1990), 359-364.

ROBERTSON, S. E., WALKER, S., AND BEAULIEU, M. 2000. Experimentation as a way of life. *Information Process. Manage. 36*, 1(2000), 95-108.

SAVOY, J. 1997. Statistical inference in retrieval effectiveness evaluation. *Information Process. Manage.33*, 4 (1997), 495-512.

SAVOY, J. 2002. Recherche d'information dans des corpus plurilingues. *Ingénierie des systèmes d'informations 7*, 1-2 (2002), 63-93.

SAVOY, J. 2004a. Combining multiple strategies for effective monolingual and cross-lingual retrieval. *IR Journal 7*, 1-2 (2004), 121-148.

SAVOY, J. 2004b. Report on CLIR task for the NTCIR-4 evaluation campaign. In *Working Notes of the NTCIR-4*. N. Kando (ed.). Tokyo, June 2004, 178-185.

SAVOY, J. 2004c. Report on CLEF-2003 monolingual tracks: Fusion of probabilistic models for effective monolingual retrieval.. In *Comparative Evaluation of Multilingual Information Access Systems*. C. Peters et al. (eds.), Lecture Notes in Computer Science 3237, Springer-Verlag, Berlin, 322-336.

SAVOY, J. 2005. Data fusion for effective European monolingual information retrieval. In *Advances in Cross-Language Information Retrieval*. C. Peters et al. (eds.). Lecture Notes in Computer Science , Springer-Verlag, Berlin, 233-244.

SINGHAL, A., CHOI, J., HINDLE, D., LEWIS, D. D., AND PEREIRA, F. 1999. AT&T at TREC-7. In *Proceedings of the TREC-7 Conference* (Gaithersburg, MD, Nov. 1998). E.M. Voorhees and D.K. Harman (eds.). NIST Special Publication 500-242, 239-251.

SPARCK JONES, K. 1972. A statistical interpretation of term specificity and its application in retrieval. *J. Documentation 28*, 1 (1972), 11-21.

SPROAT, R. 1992. *Morphology and Computation*. The MIT Press, Cambridge, MA.

TOMLINSON, S. 2004. Lexical and algorithmic stemming compared for 9 European languages with Hummingbird SearchServer™ at CLEF 2003. In *Comparative Evaluation of Multilingual Information Access Systems*. C. Peters et al. (eds.). Lecture Notes in Computer Science 3237, Springer-Verlag, Berlin, 286-300.

VOGT, C. C. AND COTTRELL, G. W. 1999. Fusion via a linear combination of scores. *IR Journal 1*, 3 (1999), 151-173.

VOORHEES, E. M., GUPTA, N. K., AND JOHNSON-LAIRD, B. 1995. The collection fusion problem. In *Proceedings of the TREC-3 Conference* (Gaithersburg, MD, Nov. 1994). D. K. Harman (eds). NIST Special Publication 500-225, 95-104.

Received July 2004; revised November 2004; accepted December 2004.