Hindi CLIR in Thirty Days

LEAH S. LARKEY, MARGARET E. CONNELL, AND NASREEN ABDULJALEEL
University of Massachusetts, Amherst

As participants in the TIDES Surprise Language exercise, researchers at the University of Massachusetts helped collect Hindi-English resources and developed a cross-language information retrieval system. Components included normalization, stop-word removal, transliteration, structured query translation, and language modeling using a probabilistic dictionary derived from a parallel corpus. Existing technology was successfully applied to Hindi. The biggest stumbling blocks were collection of parallel English and Hindi text and dealing with numerous proprietary encodings.

Categories and Subject Descriptors: H.3.3 [**Information Systems**]: Information Storage and Retrieval– *Information search and retrieval* H.3.1 [**Information Systems**]: Information Storage and Retrieval - *Content analysis and indexing: Dictionaries, indexing methods, Linguistic processing*
General Terms: Design, Experimentation, Languages
Additional Key Words and Phrases: Hindi, cross-language, cross-lingual information retrieval, evaluation

1 INTRODUCTION

The participation of the University of Massachusetts in the retrieval task of the TIDES Surprise language exercise was undertaken with a limited staff of two nearly full-time researchers, one Hindi-speaking graduate student, and a fraction of another. In cooperation with the roughly 15 other groups who participated in the exercise, we provided resources to the Surprise language community, developed a high-quality Hindi cross-language information retrieval (CLIR) system that would support participation in the final CLIR evaluation, and completed the work in one month's time.

Given these time and resource constraints, the initial paucity of resources, and the cooperative aspect of the project, we proceeded by sharing resources and by adapting an English-Arabic CLIR system [Larkey et al. 2003] to Hindi. We concentrated our efforts on adapting what we expected to be the most important components of the system and on testing these components. In what follows we describe the operation of the search system. Then, we focus on normalization, stop-word removal, stemming, transliteration, and dictionaries. We describe how we developed Hindi versions of these components and evaluated their effectiveness. Finally, we present results on the Surprise language

ACM Transactions on Asian Language Information Processing, 2003, 2(2), pp. 130-142.
This research was supported in part by the Center for Intelligent Information Retrieval and in part by SPAWARSYSCEN-SD grant numbers N66001-99-1-8912 and N66001-02-1-8903. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor. Authors' address: University of Massachusetts, Department of Computer Science, 140 Governors Drive, Amherst, MA 01003-4610. Email: {larkey|connell|nasreen}@cs.umass.edu.
Permission to make digital/hard copy of part of this work for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication, and its date of appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. © 2003 ACM 1073-0516/01/0300-0034 $5.00

evaluation queries. We found that approaches that work well for other languages also work for Hindi, with a few surprises.

2 CROSS-LANGUAGE INFORMATION RETRIEVAL

Cross-language (or cross-lingual) information retrieval (CLIR) refers to the retrieval and ranking of documents in one language in response to a query issued in a different language. Retrieving documents that the user might be unable to read may seem strange, but such a facility can allow us to select a small number of documents for manual translation, or to search for documents in many languages using a single query. We present an overview here of the cross-language search system that we use for Arabic and for Hindi. Detailed explanations of Hindi-specific normalization, stop-words, stemming, transliteration, and dictionaries follow later.

Retrieval Models

We used two different search engines, based on two different probabilistic retrieval models: (1) a *tf·idf* engine emulating INQUERY [Callan et al. 1995] and (2) cross-lingual language modeling (LM) [Berger and Lafferty 1999; Xu et al. 2001]. Both are widely used in information retrieval. The details of these models, their underlying assumptions, and a comparison of their strengths and weaknesses with respect to different resources can be found in Larkey and Connell [2003].

We review here the cross-language aspects of the two models. For INQUERY, the query is translated using the *structured query translation* method [Ballesteros and Croft 1998; Pirkola 1998]. The translated query contains all the dictionary translations for each query word, treating the alternatives for each word as a synonym set. Mathematically, this is equivalent to replacing all the occurrences of each member of the synonym set with one representative member in order to count the occurrences. Figure 1 shows a simple example of a structured query translation into Hindi for the query fragment *Indian president*.

Indian		भारतीय
	→	
president		#syn (अधिपति अध्यक्ष मुखिया राष्ट्रपति सभापति सरदार)

Fig. 1. Structured query translation example.

In cross-lingual language modeling, the system estimates, for each document, the probability that the query was generated by the document, as follows:

$$P(Q_e \mid D_h) = \prod_{e \in Q_e} \left(\lambda \sum_{h \in Hindi} P(h \mid D_h) P(e \mid h) + (1-\lambda) P(e \mid GE) \right)$$

where *e* is an English word in query Q_e; $P(h|D_h)$ is the probability of drawing the Hindi word *h* from the document D_h; $P(e|h)$ is the conditional probability of choosing a particular English translation *e* for the Hindi word *h*; and $P(e|GE)$ is the probability of drawing word *e* from a background model of general English. $P(e|h)$ is also called a *translation probability*, and requires a bilingual lexicon with probability estimates.

Previous work [Xu, et al. 2001; Larkey and Connell 2003] has shown that language modeling can produce superior results when a probabilistic dictionary based on parallel corpora is available; but structured query translation works as well or better when good probability estimates are not available. We have also found it easier to incorporate additional components like transliteration (discussed below) into structured query translation. Combining the outputs of the two engines tends to produce more effective retrieval. Ranked lists are merged by averaging scores that have first been linearly transformed to fall within a range between 0 and 1.

Along with the corpus to be searched, both approaches to cross-lingual retrieval require a bilingual lexicon or dictionary, that is, a list of Hindi translations for each English word, including the Hindi-to-English translation probabilities required by the language model. Terms in the Hindi corpus are indexed, sometimes normalizing and/or stemming the index terms, and stop-words (described below) are often excluded from indexing. Hindi terms in the dictionary receive the same preprocessing (normalization, stemming, stop-word removal) as the corpus.

An English corpus is used for estimating background probabilities in language modeling. Such a corpus should be as large as possible, and should contain text as similar to the target Hindi text as possible (e.g., news articles), covering as similar a time span as possible. We used a large corpus that we had used successfully in our Arabic work – 616,650 Associated Press articles from the years 1994-1998, from the Linguistic Data Consortium's North American News Supplement [LDC 1998]. Lower-case (not stemmed) terms in the corpus were indexed, excluding stop-words. This corpus was also used for English query expansion under both retrieval models.

Query Expansion

One of the most reliable findings in formal information retrieval evaluations is that expansion techniques improve retrieval, as measured by an increase in mean average precision [Oard and Gey 2003; Peters et al. 2002; NTCIR Workshop 2001]. In one common approach, *pseudo-relevance feedback* (PRF), we take the top-ranked documents from a first retrieval pass and add terms from the top documents into the query.

The Retrieval Process

In all the experiments reported below, the retrieval process proceeds as follows:

1. Remove English stop-words and convert the query to lower-case.
2. Expand the query by searching the English corpus and adding the top-ranked 5 terms from the top-ranked 10 documents to the query via pseudo-relevance feedback.
3. Conduct an INQUERY search, first translating the query via structured query translation (SQT). This structured Hindi query can be used to search the Hindi corpus, or can be expanded by adding terms from a first-pass search of the corpus and then performing a second search using the expanded query.
4. Conduct a language model search. Query expansion can be performed in a first pass, with a final search based on the expanded query.
5. Combine the ranked lists from the INQUERY search and the LM search to obtain the final ranking of retrieved documents.

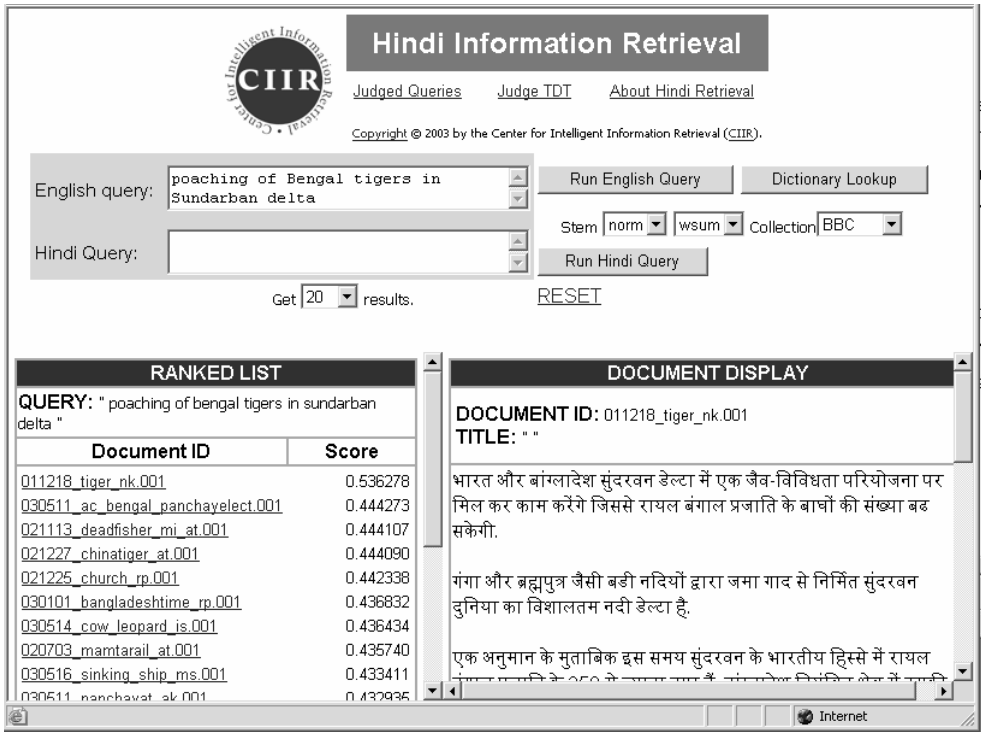
In most respects this is a fairly standard CLIR system. However, it is unusual to use two search engines and to combine their results.

3 TEST DATA

For the development phase of the Surprise language exercise, researchers at the University of Maryland contributed a set of 2927 BBC documents in Hindi and a set of 29 known-item queries (intended to retrieve exactly one particular document). A Hindi-speaking graduate student at the University of Massachusetts judged an additional 605 documents for these 29 queries, finding a few additional relevant documents. The final test set had 46 relevant documents for 29 queries. This query set was too small for significance testing, and presented a large risk of over-fitting if parameters were tuned on this data. However, it was sufficient for detecting large differences in effectiveness among techniques. Another limitation of this data set was that the corpus was too small to use for Hindi query expansion. On experiments using this test data only English queries were expanded. We relied on the data to make decisions about components of our English-Hindi CLIR system, and found that large differences on the test set generally predicted a difference in the final evaluation set.

4 THE WEB INTERFACE

In order to develop and debug our Hindi system, we modified a web-based retrieval system originally developed for Arabic. This interface allows a user to enter a query in English or in Hindi, to search a choice of Hindi text collections (monolingual search for a Hindi query, cross-lingual search for an English query), or to examine dictionary entries for the words in the query. As additional resources (corpora, transliteration models, stemmers) became available during the month, we added them to the interface. A screen shot of the interface is shown in **Fig. 2**.



Copyright © 2003 by the Center for Intelligent Information Retrieval (CIIR).

English query: poaching of Bengal tigers in Sundarban delta

Hindi Query:

Stem norm wsum Collection BBC

Get 20 results.

RANKED LIST

QUERY: " poaching of bengal tigers in sundarban delta "

Document ID	Score
011218_tiger_nk.001	0.536278
030511_ac_bengal_panchayelect.001	0.444273
021113_deadfisher_mi_at.001	0.444107
021227_chinatiger_at.001	0.444090
021225_church_rp.001	0.442338
030101_bangladeshtime_rp.001	0.436832
030514_cow_leopard_is.001	0.436434
020703_mamtarail_at.001	0.435740
030516_sinking_ship_ms.001	0.433411
030511_panchayat_ak.001	0.432935

DOCUMENT DISPLAY

DOCUMENT ID: 011218_tiger_nk.001
TITLE: " "

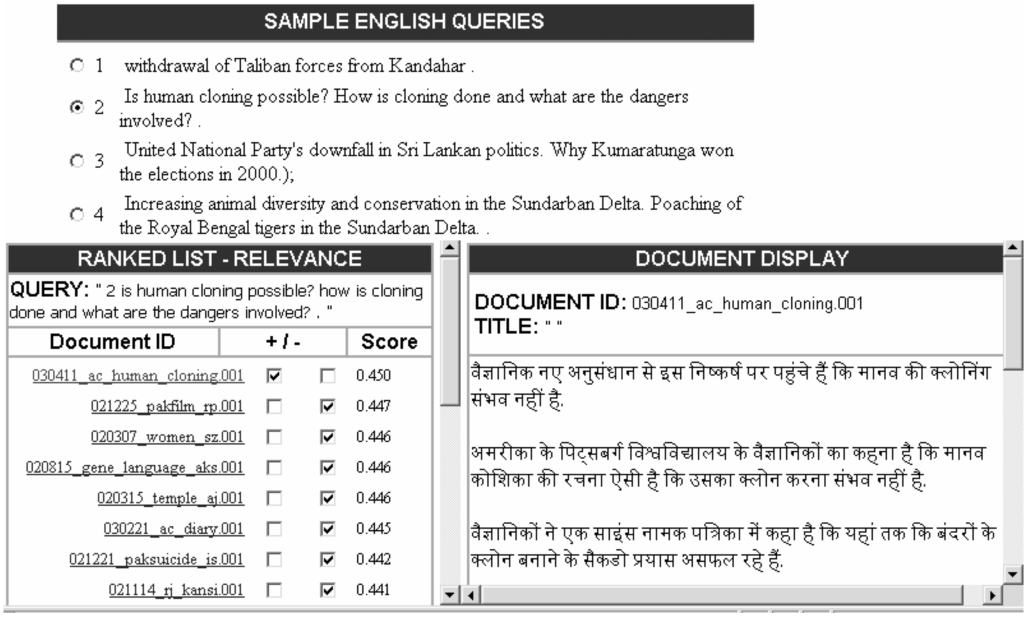
भारत और बांग्लादेश सुंदरवन डेल्टा में एक जैव-विविधता परियोजना पर मिल कर काम करेंगे जिससे रायल बंगाल प्रजाति के बाघों की संख्या बढ सकेगी.

गंगा और ब्रह्मपुत्र जैसी बडी नदियों द्वारा जमा गाद से निर्मित सुंदरवन दुनिया का विशालतम नदी डेल्टा है.

एक अनुमान के मुताबिक इस समय सुंदरवन के भारतीय हिस्से में रायल

Fig. 2. Screen shot of a web-based search system.

The web system also includes a facility for collecting and displaying relevance judgments, shown in **Fig. 3**. We used this system to judge the 605 documents for the 29 test queries mentioned in Section 3. In addition, this interface allowed us a direct view of the rankings of documents relevant to the test queries under different conditions. A version of this system was also used to collect judgments for topic-detection and tracking task [Allan et al. 2003].



SAMPLE ENGLISH QUERIES

○ 1 withdrawal of Taliban forces from Kandahar .

● 2 Is human cloning possible? How is cloning done and what are the dangers involved? .

○ 3 United National Party's downfall in Sri Lankan politics. Why Kumaratunga won the elections in 2000.);

○ 4 Increasing animal diversity and conservation in the Sundarban Delta. Poaching of the Royal Bengal tigers in the Sundarban Delta. .

RANKED LIST - RELEVANCE

QUERY: " 2 is human cloning possible? how is cloning done and what are the dangers involved? . "

Document ID	+ / -		Score
030411_ac_human_cloning.001	☑	□	0.450
021225_pakfilm_rp.001	□	☑	0.447
020307_women_sz.001	□	☑	0.446
020815_gene_language_aks.001	□	☑	0.446
020315_temple_aj.001	□	☑	0.446
030221_ac_diary.001	□	☑	0.445
021221_paksuicide_is.001	□	☑	0.442
021114_rj_kansi.001	□	☑	0.441

DOCUMENT DISPLAY

DOCUMENT ID: 030411_ac_human_cloning.001
TITLE: " "

वैज्ञानिक नए अनुसंधान से इस निष्कर्ष पर पहुंचे हैं कि मानव की क्लोनिंग संभव नहीं है.

अमरीका के पिट्सबर्ग विश्वविद्यालय के वैज्ञानिकों का कहना है कि मानव कोशिका की रचना ऐसी है कि उसका क्लोन करना संभव नहीं है.

वैज्ञानिकों ने एक साइंस नामक पत्रिका में कहा है कि यहां तक कि बंदरों के क्लोन बनाने के सैकडो प्रयास असफल रहे हैं.

Fig. 3. Screen shot of the judgment interface.

5 ENCODING, NORMALIZING, AND REMOVING STOP-WORDS

In early discussions among the groups comprising the Surprise language community, it was decided that Unicode UTF-8 would be the official exchange format for all Hindi text. Unicode is a standard multibyte encoding of characters, which is designed to cover all the symbols of all human alphabets, and some other notation systems as well [Unicode 2003]. It became clear to the participants in the Surprise language project that, due to many factors such as the composite nature of Hindi characters, there was a great deal of variation in how Hindi could be represented in Unicode. For example, the character ओ can be represented in Unicode as one value (Unicode 0913), as a sequence of two Unicode values (0905=अ , 094B=ो), or as a sequence of three Unicode values in either of two orders, (0905=अ , 093E=ा, 0947=े) or (0905=अ , 0947=े, 093E=ा). The Surprise language group's encoding and normalization subcommittee defined a standard form, including a canonical ordering for character components, and distributed a script to convert text into this standard form; we refer to this form as *LDCNorm.*

On top of LDCNorm, we performed some additional normalization to compensate for spelling and punctuation differences that could make queries, dictionaries, and corpora incompatible. Our Hindi expert felt that these distinctions are often not maintained in normal text. The normalizations include the following: replace Hindi end-of-sentence character with "."; replace Hindi numerals with Arabic numerals; remove internal word space (Unicode 200D); normalize three nasality markers (chandra bindu, n+virama, and anusvara), changing all three to anusvara; replace chandra, a rare vowel sometimes used in foreign words, with a common similar vowel; remove all nukta and (remaining) virama (diacritics); and replace certain vowel sequences with a composite vowel. We call this *UMassNorm*, to distinguish it from LDC's normalization.

We produced a stop-word list for Hindi, according to the following common strategy: From the 1000 most frequent words in the BBC corpus, our Hindi speaker manually selected words that qualified as closed-class words, i.e., prepositions, pronouns, conjunctions, particles, common adverbs, auxiliary verbs, and inflections of other very common verbs like *say*. Later, when a collection of documents from Naidunia news was posted on the LDC processed resources page, we manually added some words from its top 1000 most frequent words. The final list had 253 words, later expanded to 275 by the addition of variants. Removing stop-words from queries is usually effective in IR for two reasons: first, these words are relatively content-free; and second, stop-words tend to have a large number of dictionary translations. This becomes a special problem when the dictionary was made by aligning a parallel corpus. Syntax differences between the two languages can result in incorrect alignments for stop-words, adding noise to CLIR if they are used as translations.

The first set of experiments was performed using a small bilingual lexicon, described in Section 7. Table I shows the average precision for the 29 queries, comparing different forms of preprocessing on the queries and indexed terms. In the column labeled *Baseline,* no normalization or stop-word removal was performed on the corpus, queries, or dictionary. In the column labeled *UMassNorm,* Hindi text was normalized as described above. In the column labeled *UMassNorm+Stop*, both normalization and stop-word removal were performed on Hindi text.

Table I. Mean Average Precision On 29 Test Queries (structured query translation)

	Baseline	**UMassNorm**	**UMassNorm+Stop**
No query expansion	.2587	.2948	.3308
English query expansion	.2730	.3329	.3861

This experiment demonstrates that the normalization compensated successfully for at least some differences in spelling conventions, diacritics, etc., which might occur among the queries, corpus, and dictionary. It also demonstrates that the stop-word list was effective. We offered our normalization script and our stop-word list to the Surprise language community. The list was used by many other groups; but the normalizer was not, probably due to confusion about what it was and its relation to the normalization effort that occupied other participants throughout much of the month.

6 TRANSLITERATION

Out-of-vocabulary words are always a problem for dictionary-based CLIR. In typical evaluations, around 50% of out-of-vocabulary words are names [Davis and Ogden 1998]. When the query and document languages have different alphabets, (for example, English queries and Hindi, Arabic, or Japanese documents), transliteration (rendering the English word in the characters of the document language) can produce a correct Hindi spelling for out-of-vocabulary English names or technical terms. Our previous work on Arabic showed that a transliteration engine can be trained automatically from a few hundred name pairs to generate Arabic spellings, and that retrieval effectiveness can be improved by transliterating out-of-vocabulary words, even without knowing whether the words are names. (The technique is described in detail in AbdulJaleel and Larkey [2003].)

To summarize briefly, the transliteration model is a generative statistical model that produces a string of Hindi characters from a string of English characters by replacing English characters or n-grams with Hindi character n-grams. Each English n-gram can have more than one possible Hindi replacement, with an associated probability. Some examples from the English-Hindi model are: P(द | *d*)=0.780; P(ड | *d*)=0.220; P(ध | *dh*)= 0.913; P(ढ | *dh*)=0.087.

The model was trained from lists of a few hundred proper name pairs in English and Hindi via two alignment stages: the first was to select n-grams for the model, and the second to determine the translation probabilities for the n-grams**.** We used GIZA++ for the alignments [Och and Ney 2000]. GIZA++ was designed for word alignment of sentence-aligned parallel corpora, and we used it to do character-level alignment of word pairs, treating n-grams as words.

The transliteration model was used to generate Hindi translations and associated scores for any English query words not found in the dictionary during query translation. The top 20 scoring transliterations for each such word were added to the translated query, and treated as synonyms under structured query translation. In language model conditions, the transliterations were treated as if they came from the dictionary, but all were assigned probabilities P(*e*|*h)*=0.3, a rough estimate that was not tuned or explored empirically.

Table II. Mean Average Precision On 29 Queries With and Without Transliteration (small dictionary)

	UMassNormStop	**Translit**
No query expansion	.3308	.4030
English query expansion	.3861	.4399

Table II shows the effect of adding transliteration to retrieval. Transliteration increased retrieval effectiveness on this query set using this dictionary because it generated Hindi spellings for the 184 of the 343 query words not found in the small dictionary. When we later ran this experiment using the IBM probabilistic dictionary, transliteration made absolutely no difference, since the IBM dictionary covered almost all the query words. Nevertheless, we used transliteration in our final submission, to handle any query words that had no translations.

7 DICTIONARIES

We used two different dictionaries for our development work, one nonprobabilistic and one probabilistic dictionary. The nonprobabilistic dictionary, *Small*, was derived from two sources: a master lexicon provided early in the month by the LDC, and a small list of around 400 place names in both English and Hindi. The place names included country names, Indian city names, and Indian state names provided by ISI (Information Sciences Institute of the University of Southern California).

IBM later posted a bilingual lexicon, resulting from the alignment of parallel English and Hindi news text, with alignment counts. After normalization, cleanup, and removal of English and Hindi stop-words, we made a probabilistic dictionary from this lexicon, as follows: English or Hindi words that contained more than one token after normalization were removed. Word pairs that became identical as a result of normalization were merged, summing their counts. Finally, counts were converted to P(*e*|*h*) probabilities by dividing the count for a Hindi-English pair by the sum of the counts for all the pairs with the same Hindi term. Pairs with probability below .01 were removed. This dictionary was used for development and for our final submission.

A third bilingual lexicon was contributed by ISI, but never posted on the LDC processed resources page. We discovered this lexicon only after the Surprise language exercise was over. We include it here because other researchers in the exercise used it; it was made into a probabilistic dictionary in the same way as the IBM dictionary. **Table III** shows the sizes of these three dictionaries.

Table III. Number of Dictionary Entries for the Three Dictionaries

Dictionary	Number of pairs	English words	Hindi words
Small	69,195	21,842	33,251
IBM	181,110	50,141	77,517
ISI	512,248	65,366	97,275

As Table III shows, the IBM dictionary was far larger than the small dictionary, and the ISI dictionary was far larger than the IBM dictionary. A comparison of performance based on the three dictionaries can be seen in Table IV. Because the IBM and ISI dictionaries contained good probability estimates, we expected language modeling (LM) to be effective, so at this point we added it to our experiments. We continued to test structured query translation (SQT) as well. In these experiments, stop-words were removed, text was normalized, and unknown words were transliterated.

Table IV. Mean Average Precision On 29 Test Queries (comparing dictionaries)

	No Query Expansion			**English Expansion**		
	SQT	**LM**	**Combo**	**SQT**	**LM**	**Combo**
Small	.4156	.4179	.4319	.4528	.4385	.4725
IBM	.6080	.6208	.6276	.6379	.6486	.6723
ISI	.6847	.6815	.6681	.6592	.6869	.6530

The columns labeled *Combo* show the performance on the combination of both retrieval engines. Not surprisingly, the IBM and ISI dictionaries produced better results than the small dictionary, not only because of the presence of probabilities, but because it had far better coverage, particularly of names. The comparison between IBM and ISI shows mixed results. It appears that IBM performed better than ISI on unexpanded queries, but not on expanded queries. On the basis of the comparison between the small and IBM dictionaries, IBM was used for the final submission. The results above led us to expect that the effectiveness of the IBM and ISI dictionaries would be comparable.

8 STEMMING

Stemmers are widely used in information retrieval. Their use can improve retrieval effectiveness, particularly for highly inflected languages, generally more for monolingual retrieval, and for cross-lingual retrieval based on nonprobabilistic dictionaries than for cross-lingual retrieval based on probabilistic dictionaries [Larkey and 2003]. Previous stemming research has shown that even in a highly inflected language like Arabic, a very light stemmer (one that removes a few common affixes) allows as good or better retrieval than a more complicated morphological analyzer [Aljlayl and Frieder 2002; Larkey et al. 2002]. Our Hindi speaker made a list of 27 common suffixes, shown in **Fig. 4**, which indicate gender, number, tense, and nominalization. Our light stemmer removed all of these suffixes, the longest suffix first.

ूँ, ूं, ें, ें, ों, ोँ, यों, यौँँ, िए, ता, ती, ते, ना, नी, ने, के, ा, े, ी, ो, ेंगे, ूँगा, ूँगी, ेगा, ेगी, ियाँ, याँ

Fig. 4. Suffixes removed by the UMass light stemmer.

At the same time, BBN contributed a stemmer based on Ramanathan and Rao [2003], which removed 65 inflectional suffixes.

We compared the effectiveness of these two stemmers with unstemmed retrieval. These experiments were carried out on the same test queries, using the IBM probabilistic dictionary, both retrieval engines, and their combination. The results, seen in **Table V**, show that both forms of stemming seem to make retrieval less effective. On the basis of these results, we decided not to use any stemming in subsequent work; we will return to the stemming question later.

Table V. Mean Average Precision On 29 Test Queries With and Without Stemming

	No Query Expansion			**English Expansion**		
	SQT	**LM**	**Combo**	**SQT**	**LM**	**Combo**
LDCNorm	.5787	.5756	.5840	.6347	.5937	.6404
UMassNorm	.6080	.6208	.6276	.6379	.6486	.6723
UMass stemming	.5335	.5527	.5343	.5713	.5324	.5415
BBN stemming	.5544	.5571	.5557	.5595	.5339	.5391

9 SURPRISE LANGUAGE EVALUATION

The Surprise language evaluation consisted of 15 English topics, Hindi translations of these topics, and a collection of 41,697 Hindi news articles from several different sources. Topics consisted of title, description, narrative, and search terms. We searched the corpus with queries comprised of title, description, and search term (TDS) portions of the topic, and with title, description, narrative, and search term portions (TDNS), and ran both monolingual and cross-lingual searches. Cross-lingual searches were carried out using the same search system described above, but included query expansion on both the English and Hindi side. For monolingual Hindi searches, queries were normalized but not stemmed, and stop-words were removed. Monolingual Hindi queries were expanded via pseudo-relevance feedback (PRF).

Table VI shows mean average precision on 8 runs based on the combination of the two retrieval engines. The table also shows the number of relevant documents returned in the top 5, 10, and 20 ranks, to allow comparison with research from other authors reported in this issue. Since 15 queries are too few for significance testing, no such tests were carried out.

Table VI. Retrieval Effectiveness On Surprise Language Evaluation

	Expansion	**Topic Sections Included**	**Mean Average Precision**	**Num. Relevant Returned**		
				Top 5	**Top 10**	**Top 20**
Monolingual retrieval	None	TDS	.4540	3.734	6.867	10.934
		TDNS	.4661	3.400	6.933	11.666
	Hindi	TDS	.4696	3.800	6.800	11.466
		TDNS	.4738	3.600	7.067	11.866
Cross-lingual retrieval	English	TDS	.3722	3.334	5.667	10.266
		TDNS	.4067	3.600	6.267	10.600
	English+Hindi	TDS	.4047	3.400	6.200	10.666
		TDNS	.4298	3.667	6.533	11.000

These results demonstrate effective retrieval, and confirm the general pattern that expanded queries are more effective than unexpanded ones.

10 STEMMING AND MONOLINGUAL VS CROSSLINGUAL RETRIEVAL

In previous work we found that stemming can have a larger effect on monolingual retrieval than on cross-language retrieval. We did not have Hindi versions of the 29 test queries, so we could not perform monolingual experiments during development. However, the 15 evaluation topics included Hindi fields, so we were able to test stemming with these monolingual queries.

Table VII shows the effect of stemming on the 15 evaluation topics, using the same combination retrieval system described in Section 9 and TDS (title, description, and search terms) queries. The four columns show mean average precision using the same versions of normalization and stemming discussed in Section 8.

Table VII. Mean Average Precision On 15 Evaluation Queries (comparing normalization and two stemmers)

	Expansion	**LDCNorm**	**UMassNorm**	**UMass**	**BBN**
Monolingual	None	.4133	.4540	.4710	.4632
	Hindi	.4280	.4696	.4731	.4629
Cross-lingual	English only	.3467	.3722	.3645	.3580
	English + Hindi	.3856	.4047	.3851	.3811

The first two columns show that UMass normalization (UMassNorm) produced more effective retrieval than the baseline (LDCNorm), confirming the test query results in **Table I**. The data support weaker conclusions about stemming. Both stemmers (UMass and BBN) appear to improve performance slightly over UMassNorm, but only on unexpanded monolingual queries. The cross-language case is clearer, and confirms the results seen in the 29 test queries: i.e., stemming does not improve retrieval effectiveness. The lack of effect from stemming could be due to any of several factors. It is possible that the use of a dictionary derived from a parallel corpus reduced or eliminated the need for stemming, because it contained most of the likely inflections for Hindi words. Another possibility is that there is no great need for stemming in a language like Hindi which is not highly inflected. Finally, these particular stemmers may be poor.

In contrast, Chen and Gey [2003] tested a statistical stemmer for Hindi and found that stemming improved cross-lingual retrieval but not monolingual retrieval. Perhaps their automatically-built stemmer is more effective than either of the two hand-crafted ones. However, it is difficult to compare their work directly to ours, because the two systems differed in almost every phase of processing, from the portions of the topic used for the query to the retrieval engines and bilingual lexicons. We believe that the most important difference is the UMass normalization, which performs some of the same kinds of conflation that a statistical stemmer does. However, this normalizing conflation worked best without any additional stemming (i.e., affix stripping). This issue should be pursued in future work.

11 DICTIONARY COMPARISON ON EVALUATION QUERIES

We repeated the comparison between the two probabilistic dictionaries discussed in Section 7. The results can be seen in **Table VIII**, and are consistent with those on the test data in Section 7. There is no obvious difference in performance between the two dictionaries.

Table VIII. Comparing IBM and ISI Dictionaries On Cross-lingual Retrieval

	Expansion	**Average Precision**	**Num. Relevant Returned**		
			Top 5	**Top 10**	**Top 20**
IBM	English	.3722	3.3335	5.667	10.266
	English+Hindi	.4047	3.4	6.2	10.666
ISI	English	.4173	3.2	6.067	10.134
	English+Hindi	.4285	3.584	6.267	10.134

12 COMMUNICATION AND COLLABORATION ISSUES

This project was very much a collaboration among roughly 16 institutions. Conference calls were held three to five times per week, hundreds of emails were posted, many containing resources as attachments, and resources were placed on an LDC web page where participants could download them. This process worked fairly well in the exchange of resources to avoid duplication of effort and in keeping participants apprised of what issues were causing the most serious problems.

The two problems of encoding and obtaining parallel news text turned out to be the biggest stumbling blocks, not only for our site, but for all the participating groups. The final version of the tools to standardize Unicode text was not available until the last few days of the evaluation. We found ourselves reprocessing text resources numerous times as new versions of the official normalization software became available.

We provided a stop-word list (that other groups used as well), a simple normalizer (it improved our performance, but other groups did not use it), and relevance judgments. We also provided experimental results showing that the IBM dictionary worked far better than the small dictionary, and that stemming did not seem to improve performance.

On the down side, the volume of communication, particularly email, was overwhelming. Groups tended to release untested resources that other groups spent time determining were not useful. Conversely, some useful resources never reached the LDC processed resources page, and important information could become swamped in the hundreds of messages that were less relevant to our work.

On the whole, however, the collaboration was successful. Resource-sharing made it possible for different sites to complete the exercise. The most important resources for cross-language information retrieval that we obtained from other groups were the bilingual lexicons derived from parallel corpora prepared by the statistical machine translation groups, and the normalization software.

13 CONCLUSIONS

The CLIR component of the Surprise language exercise was a success. Hindi presented some language-specific obstacles (proprietary encodings of much of the web text, lack of availability of parallel news text, and variability in Unicode encoding), which made it difficult to pull resources together. Our work was heavily dependent upon the work of the groups that gathered the parallel text and cracked the encodings.

We were able to do a good job at cross-language information retrieval using existing language-independent technology and at adapting language-specific components with minimal customization. Although we relied on the expertise of a Hindi speaker in developing our normalizer and stemmer, each only required approximately a day of her time. The generative transliteration model was developed automatically, and the stop-word list was built automatically with a small amount of manual editing.

Testing a small set of queries against a small corpus of 2927 documents was a surprisingly effective way to predict what techniques would work on an independent set of data. We found in the test data, and confirmed in the evaluation data, that although two different stemmers did not improve cross-lingual retrieval, careful normalization, removal of stop-words, query expansion, transliteration of out-of-vocabulary words, and combination of evidence contributed to effective Hindi retrieval.

ACKNOWLEDGMENTS

We would like to thank Hema Raghavan for providing relevance judgments. participating

REFERENCES

ABDULJALEEL, N. AND LARKEY, L. 2003. Statistical transliteration for English-Arabic cross language information retrieval. In *CIKM 2003: Proceedings of the Twelfth International Conference on Information and Knowledge Management* (New Orleans, LA, Nov. 2003). O. Frieder et al. eds. ACM, New York, 139-146.

ALJLAYL M. AND FRIEDER, O. 2002. On Arabic search: Improving the retrieval effectiveness via a light stemmer approach. In *CIKM 2002: Proceedings of the Eleventh International Conference on Information and Knowledge Management* (McLean, VA, Nov. 2002). K Kalpakis. et al. eds. ACM, New York, 340-347.

ALLAN, J., LAVRENKO, V., AND CONNELL, M.E. 2003. A month to topic detection and tracking in Hindi. This issue. **[ED: CHECK!]**

BALLESTROS, L. AND CROFT, W.B. 1998. Resolving ambiguity for cross-language retrieval. In *Proceedings of the21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Melbourne, Aug. 1998), W.B. Croft et al. eds. ACM, New York, 64-71.

BERGER, A. AND LAFFERTY, J. 1999. Information retrieval as statistical translation. In *Proceedings of SIGIR '99: 22nd International Conference on Research and Development in Information Retrieval* (Berkeley, CA, Aug. 1999), M. Hearst et al. eds. ACM, New York, 222-229.

CALLAN, J.P., CRIFT, W.B. AND BROGLIO, J. 1995. TREC and TIPSTER experiments with INQUERY. *Inf. Process. Manage. 31* (1995), 327-343.

CHEN, A. AND GEY, F.C. 2003. Generating statistical Hindi stemmers from parallel texts. This issue. **[ED: CHECK!]**

DAVIS, M.W. AND OGDEN, W.C. 1998. Free resources and advanced alignment for cross-language text retrieval. In *Proceedings of the Sixth Text Retrieval Conference: TREC-6* (Gaithersburg, MD, Nov. 1997), E. M. Voorhees et al. eds. NIST Special Publication 500-240, 385-394.

LARKEY, L.S., ALLAN, J., CONNELL, M.E., BOLIVAR, A. AND WADE, C. 2003. UMass at TREC 2002: Cross language and novelty tracks. In *The Eleventh Text REtrieval Conference: TREC 2002* (Gaithersburg, MD, Nov. 2002), E.M. Voorhees et al. eds. NIST Special Publication 500-251, 721-732.

LARKEY, L.S. AND CONNELL, M.E. 2003. Structured queries, Language modeling, and relevance modeling in cross-language information retrieval. *Inf. Process. Manage.* To appear

.

LARKEY, L.S., BALLESTROS, L., AND CONNELL, M.E. 2002. Improving stemming for Arabic information retrieval: Light stemming and co-occurrence analysis. In *SIGIR 2002: Proceedings of the Twenty-Fifth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Tampere, Finland, Aug. 2002), M. Beaulieu et al. eds. ACM, New York, 275-282.

LDC. 1998. Linguistic Data Consortium North American News Text Supplement, LDC98T30. http://www.ldc.upenn.edu/Catalog/

NTCIR WORKSHOP 2. 2001. *Proceedings of the Second NTCIR Workshop on Research in Chinese and Japanese Text Retrieval and Text Summarization* (Tokyo, March 2001). http:/research.nii.ac.jp/ntcir/workshop/OnlineProceedings2.

OARD, D.W. AND GEY, F.C. 2003. The TREC-2002 Arabic/English CLIR track, In *The Eleventh Text REtrieval Conference: TREC 2002* (Gaithersburg, MD, Nov. 2002), E.M.Voorhees et al. eds. NIST Special Publication 500-251, 17-26.

OCH, F.J. AND NEY, H. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics* (Hong Kong, Oct. 2000), 440-447.

PETERS, C., BRASCHLER, M., GONZALO, J., AND KLUCK, M. EDS. 2002. *Evaluation of Cross-Language Information Retrieval Systems: Second Workshop of the Cross-Language Evaluation Forum, CLEF 2001*: (Darmstadt, Germany, Sept. 2001). Revised papers. Lecture Notes in Computer Science, Vol. 2406, Springer, New York.

PIRKOLA, A. 1998. The effects of query structure and dictionary setups in dictionary-based cross-language information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Melbourne, Aug.1998), W.B. Croft et al. eds. ACM, New York, 55-63.

RAMANATHAN, A. AND RAO, D.D. 2003. A lightweight stemmer for Hindi. Presented at *EACL 2003: 10th Conference of the European Chapter of the Association for Computational Linguistics, Workshop on Computational Linguistics for South Asian Languages* (Budapest, April 2003.). http://computing.open.ac.uk/Sites/EACLSouthAsia/papers.htm

UNICODE, 2003. What is Unicode? http://www.unicode.org/standard/WhatIsUnicode.html.

XU, J., WEISCHEDEL, R. AND NGUYEN, C. 2001. Evaluating a probabilistic model for cross-lingual information retrieval. In *SIGIR 2001: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (New Orleans, LA, Sept. 2001), W.B. Croft et al. eds. ACM, New York, 105-110.

Received **August 2003**; revised **November 2003**; accepted **October 2003**