The HAIRCUT Information Retrieval System

James Mayfield and Paul McNamee

The complexity of human language makes accessing multilingual information difficult. Most cross-language retrieval systems attempt to address linguistic variation through language-specific techniques and resources. In contrast, APL has developed a multilingual information retrieval system, the Hopkins Automated Information Retriever for Combing Unstructured Text (HAIRCUT), which incorporates five language-neutral techniques: character *n*-gram tokenization, a proprietary term similarity measure, a language model document similarity metric, pre-translation query expansion, and exploitation of parallel corpora. Through extensive empirical evaluation on multiple internationally developed test sets we have demonstrated that the knowledge-light, language-neutral approach used in HAIRCUT can achieve state-of-the-art retrieval performance. In this article we discuss the key techniques used by HAIRCUT and report on experiments verifying the efficacy of these methods.

INTRODUCTION

The popularity of Web search engines and the reasonably high quality of the results they provide have created the sense that the document search problem is largely solved. Yet many domains are not well served by the technologies that underlie today's search engines.

Consider the case of an intelligence analyst who must search for documents in languages other than English. Often, it would be preferable to allow the analyst to enter the query in his or her native language rather than in the language being targeted. The retrieval system should support such queries for several reasons. First, the ability to understand a language is easier to acquire than the ability to generate it. Thus, the analyst with limited ability to read a language may prefer to issue a query in his or her native language. Second, retrieved documents must ultimately be translated, either by the analyst or by a translation service. Given limited translation resources, it makes sense to ensure that documents are relevant to the analyst's interests before translating them. Third, when dealing with a collection, or corpus, that contains many languages, it helps to allow the analyst to pose the query just once and apply that query to each of the target languages. *Cross-language information retrieval* (CLIR), the retrieval of documents in one language that are relevant to a query expressed in another language, is thus an important tool in the intelligence analyst's arsenal.

Current approaches to CLIR tend to be language-specific to handle what are viewed as language-specific problems. These difficulties include the following:

- English tends to put morphological variation at the end of a word (e.g., "ing," "ed," "ence"). In contrast, languages like Arabic and Finnish have *infix morphology,* in which the letters in the middle of a word can change when the word is used in different ways.
- It is easy to tell which character sequences are words in English—just look for delimiting spaces. In languages like Chinese and Japanese, though, no spaces are used. Thismakesitchallengingforamachinetoidentifywordboundaries.
- In English, if it's a word then it's in the dictionary. In languages like Dutch and German, however, it's perfectly OK to glue two words together to form a new, never-before-seen word that appears in no dictionary.

Problems such as these cause most people who attempt retrieval tasks in languages other than English to focus on language-specific techniques. To obtain high-quality retrieval accuracy, many techniques incorporate language-specific resources, both for processing text in a single language and for CLIR. For example, information retrieval systems typically use stopword lists, phrase lists, stemmers, decompounders, lexicons, thesauri, part-of-speech taggers, or other linguistic tools and resources to facilitate retrieval.

Obtaining and integrating such resources is time-consuming and may be costly if commercial toolkits are used. Given that a hot spot might appear anywhere in the world and require analysis of texts in languages for which such resources may not have been developed, it makes sense to examine whether language-neutral techniques that support retrieval over any language can be developed.

This article demonstrates that language-neutral techniques for CLIR are indeed feasible. First, we present five language-neutral techniques that form the core of our approach. Then we describe the APL-developed state-of-the-art Hopkins Automated Information Retriever for Combing Unstructured Text (HAIRCUT) retrieval system and show how these techniques have made HAIRCUT one of the top cross-language retrieval systems in the world. Finally, we report on experiments using the Cross-Language Evaluation Forum (CLEF)[1] and Text REtrieval Conference (TREC)[2] test collections to evaluate retrieval accuracy across a wide spectrum of human languages. These results demonstrate conclusively that accurate cross-language retrieval is possible without language-specific resources.

LANGUAGE-NEUTRAL HUMAN LANGUAGE TECHNOLOGIES

In this section, we describe five human language technologies that are language-neutral; that is, they seem to work well across a wide variety of human languages with little or no language-specific tuning.

Character N-gram Tokenization

Any text retrieval system must characterize each document it indexes by a set of indexing terms that captures a portion of that document's content. A user's query is characterized in the same way, and a *similarity metric* is then used to compare the query characterization to each of the document characterizations. The similarity metric returns a score for each document, and the documents are presented to the user ordered from best score to worst.

Most retrieval systems use *words*, or some variant thereof, as indexing terms. An alternative to words is *character n-grams*, sequences of *n* characters in a row. For example, if we select $n = 5$, then the first few terms for the text "Four score and seven" are *_four, four_, our_s, ur_sc*, and so on.

The use of character *n*-grams in language modeling dates back at least to Claude Shannon. Although Shannon is widely known for his juggling machinery,[3] he also created the area of information theory. In his seminal paper,[4] Shannon described a sequence of character n-gram and word *n*-gram approximations to English.

Most information retrieval systems must maintain a list of all known indexing terms called a *dictionary*. The application of *n*-grams to information retrieval was derived from the desire to decrease dictionary size. While the number of words that may be found in a collection is in theory infinite as the collection grows, the number of *n*-grams is bounded by $|alphabet|^n$. For a small *n*, this number is quite tractable; when $n = 3$ for example, for the English alphabet of 26 letters plus space, at most 19,683 3-grams may be found. Thus, if memory constraints are severe, short-character *n*-grams offer an attractive representation for the retrieval system's dictionary (which, unlike the portion of the index that states which documents contain each term, is typically kept in memory).

With this goal in mind, numerous studies examined the efficiency of short, word-internal character *n*-grams. As early as 1974, de Heer[5] explored the use of "*n*-polygrams" as an alternative to words. He termed the collection of *n*-grams derived from a word the *syntactic trace* of that word. Subsequent work gradually increased *n*-gram length, studied varying lengths of *n*-grams to homogenize term frequency, and increased test collection size.[6–11]

In the 1990s, a shift occurred in how *n*-grams were viewed within information retrieval. Technical changes included an increase in *n* and a shift to word-spanning *n*-grams. Qualitatively, these changes reflected a new view of *n*-grams as indexing terms in their own right, rather than simply indirect representations of words. These changes were hinted at in Cavnar,[12] and firmly established by Damashek.[13]

Reaction to Damashek's work by Harman et al.[14] noted that Damashek's system did not perform up to the level of most other systems that participated in the TREC-3 evaluation of information retrieval systems.

However, in addition to using *n*-grams as indexing terms, Damashek's system also used a novel similarity metric. The effects of these two technologies were not separated, so one cannot safely conclude from these results that *n*-grams are fundamentally inferior to words as indexing terms, even for the TREC-3 test set. In fact, throughout the early history of *n*-grams as indexing terms, little distinction was made between the impact of *n*-grams and the impact of the particular similarity metric in use. This was understandable when the stated purpose for using *n*-grams was memory efficiency, but it makes little sense when trying to understand how *n*-grams affect retrieval accuracy.

Overlapping sequences of characters have been used for many applications other than document retrieval, including language identification,[15] spelling error detection,[16] keyword highlighting,[17] and restoration of diacritical marks.[18] *N*-grams have been recognized for their ability to retrieve documents that have been degraded as a result of optical character recognition errors.[19] However, the largest application of character *n*-grams in information retrieval is probably in retrieval of Asian-language documents.[20–22] As noted above, written languages such as Chinese and Japanese do not include word separator characters. Therefore, a word-based approach to indexing demands a segmenter that can identify word boundaries. Not only is such a segmenter language-specific (requiring new training for each language to be segmented), its errors can also degrade the quality of the index. *N*-grams, in contrast, do not treat word separators as special in any way, and so proceed blissfully onward, regardless of whether separators are present.

As we shall see, *n*-grams work well as indexing terms. This seems counterintuitive, though; why should *ur_sc*, for example, be a good indexing term for the text "Four score and seven?" *N*-grams seem to draw their power from a number of sources. First, they make up in quantity for what they lack in quality. Every character in a text (save the final $n - 1$) begins an *n*-gram, but only a fraction of those characters begin words. Second, *n*-grams naturally conflate related words. For example "juggling," "juggler," and "juggled" all contain the 4-grams *_jug*, *jugg*, and *uggl*, allowing any one of them in a query to match any of the others in a document in three places. Of course, there may also be spurious matches (e.g., to a query about "muggles"); however, spurious matches tend to match fewer *n*-grams than do related words, and those that do match tend not to reinforce each other as do indexing terms that are associated with a single topic.

A third advantage of *n*-grams, at least for larger values of *n*, is that they can capture some information about phrases in the text. Word-based systems typically use a *bag-of-words* approach in which the order of the words in the text is not preserved. Because *n*-grams can span the space between words, they preserve a small amount of information about how those words are related. For example, the presence of the 5-gram *te_ho* provides a small amount of evidence that a document also containing "white" and "house" is probably about the White House and not, say, about singer Barry White playing to a packed house.

> *We believe that the techniques described here can help intelligence analysts handle future crises—whatever the language requirements.*

Finally, *n*-grams are tremendously useful when the data being indexed contain errors. For example, it is common for documents converted to electronic form using optical character recognition to contain a variety of single-character errors. Therefore, the word "*LIBERTY*" might be interpreted by an optical character recognition system as "*UBERTY*" or "*LIBEATY*." Such a word is useless as an indexing term in a word-based system. However, in an *n*-gram–based system, the *n*-grams immediately preceding and immediately following an error are likely to be correct, allowing successful retrieval even in the presence of a large number of errors.

Thus, *n*-grams provide distinct advantages as indexing terms in monolingual retrieval. These advantages, together with their ability to handle wide variations in human languages with equanimity, make *n*-grams a fine choice for a retrieval system that operates in a multilingual setting.

Affinity Sets

A common task when processing human language is to automatically identify associations among the terms of a text. For example, synonyms of query terms or terms that are strongly correlated with them can be added to a user's query to form a new query that produces better retrieval results than the initial query. This technique, which is variously called *pseudo-relevance feedback* or *blind relevance feedback*, is widely used by research systems to improve retrieval accuracy. Term associations are also useful to automatically identify different senses of a word, to find statistical translations of words or phrases, and to locate the most important sections of a document.

In 1983, Salton[23] proposed the use of a term-term matrix to capture relationships between pairs of terms. In the simplest case, each entry in the matrix is the sum over all documents of the inner product of the two terms' term counts:

$$\mathrm{Sim}(t_i, t_j) = \sum_k f(t_i, d_k) \cdot f(t_j, d_k) .$$

Here, t_i and t_j are terms, d_k is a document, and $f(t, d)$ is the frequency of term t in document d. Cosine was also considered as an alternative to inner product.

Qiu and Frei[24] suggested a refinement to this approach to reduce the contribution of common words. They weighted terms by inverse document frequency, a metric that measures the rarity of a term based on the number of documents containing that term, n_i, and the total number of documents, N. They called their approach a *similarity thesaurus*:

$$w_{ij} = f(t_i, d_j) \cdot \log\left(\frac{N}{n_i}\right),$$

$$\mathrm{Sim}(t_i, t_j) = \sum_k w_{ik} \cdot w_{jk} .$$

Church and Hanks[25] examined an asymmetric measure that looked at the words most likely to follow a given word within a certain window. Their approach relied on the mutual information statistic,

$$P(t_i) = \frac{n_i}{N},$$

$$P(t_i, t_j) = \frac{n_{ij}}{N},$$

$$\mathrm{Sim}(t_i, t_j) = \log\left(\frac{P(t_i, t_j)}{P(t_i) \cdot P(t_j)}\right),$$

where n_{ij} is the number of documents containing both t_i and t_j. Other information theoretic measures, such as the Dice coefficient and the chi-squared statistic, have been examined in various studies.

Each of these approaches defines a score between two terms; however, it is sometimes desirable to identify words that are related to a complete phrase or concept. While the above techniques are useful for single input terms, they are undefined for phrases and other multiterm inputs. To address this shortcoming, we devised a proprietary approach to identifying related terms that we call *affinity* sets. We call the elements of an affinity set *affines* of the input word or phrase. To calculate an affinity set, we first perform document retrieval to find a small set of documents that are likely to be relevant to the given input word, phrase, or passage. We include in the affinity set those terms that occur frequently in the retrieved documents but relatively infrequently in the collection as a whole. Examples of output from the algorithm are shown in Table 1.

As seen from the table, the terms suggested by the affinity set algorithm are suitable for automated query expansion. That is, affines of a query can be appended to it to create a new query that covers more relevant terms. While a human-compiled thesaurus could also be used for query expansion, such a resource is difficult to obtain electronically and would fail to account for novel word uses (for example, "surf" as in "surfing the Web," is unlikely to be a synonym for "browse" in Roget's thesaurus). Relevance feedback requires that two separate retrievals are performed, one using the original query terms and one using an expanded set of terms. Consequently, this technique is employed primarily in applications where accuracy is more important than processing speed. Salton[23] claims that automated relevance feedback can have between a 30 and 60% effect on retrieval performance.

In addition to using affines for relevance feedback, we have also applied the method to textual data mining to identify the similarities and differences between two concepts. For example, consider the following paragraph, which describes the concepts "sushi" and "risotto":

> Sushi and risotto are both rice-based dishes that one might eat for dinner in a restaurant. Sushi is a Japanese food that might be served with tempura, seaweed, miso soup, and tea.

Table 1. Top 20 affines for four inputs derived from a collection of late 1980s newspaper articles.

"Cryptography"	"Shakespeare"	"Antarctica"	"Space shuttle"
cryptography	shakespeare	antarctica	shuttle
cryptographic	theatre	antarctic	space
encryption	plays	expedition	nasa
computer	play	polar	orbit
firmware	festival	ice	astronauts
hardware	actors	whales	aeronautics
modules	theater	earth	launch
transmitted	stratford	whale	flight
codes	actor	sea	earth
devices	stage	whaling	rocket
implementations	juliet	ozone	mission
authentication	comedy	chile	satellite
counterintelligence	characters	ocean	challenger
machines	drama	scientific	telescope
processing	productions	environmental	spacecraft
decipher	hamlet	layer	astronaut
intelligence	rsc	hole	atlantis
digital	royal	scientists	payload
algorithms	repertory	ultraviolet	manned
publications	production	continent	crew

> Risotto is Italian in origin, and may contain ricotta or parmesan cheeses, herbs, spinach, or capers. Establishments that serve risotto also tend to serve pasta and ravioli.

Although we cannot claim to generate the above paragraph automatically, we can produce many of its component facts. The five columns of Table 2 are computed based on the affines for "sushi" and "risotto." The left- and right-most columns list the most closely related terms for sushi and risotto, respectively; the center column indicates terms that tend to occur when both sushi and risotto are present; the second column contains terms that occur with sushi but not with risotto; and the fourth column contains words that occur with risotto but not with sushi. Since the techniques used to identify related terms and concepts are statistical, they can be applied to any language.

Table 2. Discovering the relationship between "sushi" and "risotto."

Only sushi	Sushi NOT risotto	Sushi AND risotto	Risotto NOT sushi	Only risotto
sushi	japanese	menu	paella	risotto
restaurant	tempura	restaurant	balsamic	sauce
fish	seaweed	sauce	ricotta	dishes
japanese	miso	chicken	ravioli	pasta
restaurants	her	chef	spinach	menu
menu	san	dishes	pasta	restaurant
sauce	bars	dinner	capers	spinach
chicken	art	grilled	polenta	wine
fried	sashimi	restaurants	parmesan	dish
bar	fish	rice	mascarpone	chicken
rice	soy	vegetables	pesto	chef
food	traditional	cooked	herbs	lunch
chef	york	salad	breakfast	dinner
dishes	she	food	cured	cheese
dinner	tea	wine	trattoria	cooking
shrimp	pieces	lunch	garlicky	grilled
cooked	avocado	cream	carpaccio	soup
beef	golden	dish	porcini	salad
grilled	space	cheese	basil	italian
vegetables	music	fried	shrimps	restaurants

Language Model Similarity Metric

HAIRCUT uses a language model to estimate the probability that a document is relevant to a user's query.[26–28] A language model is a mechanism that generates strings of a language. For example, a program that repeatedly generates the string "All work and no play makes Jack a dull boy" is a language model, albeit an impoverished one. More interesting language models will generate different strings with different probabilities. For example, we might hope that a language model designed to capture English would generate "I am not a crook" with a higher probability than "This is historic times," which would in turn have a higher probability than "Ich bin ein Berliner."

A language model can be derived from a text by simply treating each term's relative frequency in the text (i.e., the number of times the term appears in the text divided by the total number of terms in the text) as the probability that the language model will generate the term

$$P(t\,|\,D)=\frac{f(t,D)}{|D|}\,,$$

where t is a term, D is a document, and $f(t, D)$ is the number of occurrences of t in D. Such a probability estimate is called a *maximum likelihood estimate*. A model built in this way is called a *unigram model* because words are generated independently of one another, with no thought for what words have come before.

To use a language model as a similarity metric, we ask, "What is the probability that a language model derived from a given document would generate the user's query?" We can ask this question about each document in the collection and rank documents according to the resulting probabilities. Using Bayes' law, and assuming that no document or query is *a priori* more likely than any other, we get

$$P(D\,|\,Q)=\frac{P(Q\,|\,D)P(D)}{P(Q)}\approx P(Q\,|\,D)\,,$$

where D is a document and Q is a query. In its simplest form, we can estimate this as

$$P(D\,|\,Q)=\prod_{q\in Q}P(q\,|\,D)\;.$$

Unfortunately, this formula leads to a probability of zero for any document that does not contain all of the query terms. Because this is undesirable (would we really want an article about the "Hubble telescope" to be deemed unrelated to the query "Hubble space telescope" if it didn't happen to use the word *space*?), we would like to give a non-zero probability to all terms, even those that don't appear in the document. A simple way to do this is to apply *Jelinek-Mercer smoothing*,[29] which uses linear interpolation between the language model for the document and the language model for the collection as a whole:

$$P(D \mid Q) = \prod_{q \in Q} [\alpha P(q \mid D) + (1 - \alpha) P(q \mid C)] ,$$

where Q is a query, D is a document, C is the collection as a whole, and α is a smoothing parameter. The probabilities on the right side of the equation are replaced by their maximum likelihood estimates when scoring a document. The language model has the advantage that term weights are mediated by the corpus. In addition to relieving the developer of the burden of identifying a term weighting scheme, this feature admits the potential for improved performance with a larger corpus. That is, the more text you have to train your model, the better that model approximates natural language.

This language model assumes that all query terms are independent. This is untrue for words, but wildly untrue for *n*-grams (after all, adjacent *n*-grams share all but one letter). Nonetheless, the metric does not appear to suffer for its unrealistic assumption, even when applied to *n*-grams. The one effect that this increased level of dependence appears to have is to decrease the optimal value of the smoothing parameter α.

Our early tests on the language model had it consistently outperforming two well-known and high-performing retrieval models: Okapi BM25 (Ref. 30) and weighted cosine[31] (although the differences may not have been statistically significant). The other methods seem to work reasonably well with *n*-grams too (e.g., Ref. 32).

Pre-translation Query Expansion

In monolingual retrieval, query expansion and blind relevance feedback have been shown to be remarkably effective, especially when an initial query formulation lacks terms present in many relevant documents. This might occur when a query is very short or when specific domain terminology (e.g., medicine or engineering) is used.

In a multilingual setting it seems plausible that query expansion prior to translation, or *pre-translation expansion*, would indeed be helpful.[33] If a translation resource contains only a small number of translations of search terms, then the degradation arising from the translation process will cause many important query words to be unavailable for document ranking. However, if many additional words related to the query are translated, then the number of translations available for searching the target language is increased. This method presumes that the set of translated terms still represents the query semantics (i.e., the user's information request is not significantly altered by expansion and translation).

There have been many positive reports regarding the benefits of query expansion for CLIR, but negative reports have been made frequently as well. We believe that differences in test collections, retrieval systems, language pairs, and translation resources obfuscate the conclusions of prior studies. Gey and Chen[34] wrote an overview of the TREC-9 CLIR track that focused on using English queries to search a Chinese news collection. Their summaries of work by several top-scoring track participants reveal a disconcerting lack of consistency as to the merits of query expansion methods:

- There was a 10% improvement in average precision with either pre- or post-translation expansion, but only short queries benefited from the use of both.
- Pre-translation query expansion did not help.
- The best cross-language run did not use post-translation expansion.
- Pre-translation expansion yielded an improvement of 42% over an unexpanded base run.
- The best run used both pre- and post-translation expansion.
- Post-translation query expansion yielded little improvement.

With inconsistent results like these, it is impossible to ascertain which techniques work. Each of the systems referred to above used different translation resources, and we believe this lies at the heart of the confusion. When translation resources are good, little is gained by expanding a query because it is already being translated adequately. On the other hand, when translation resources are poor, large gains in accuracy may be obtained through expansion. Experiments presented later in this article clearly demonstrate this relationship between translation resource quality and the efficacy of pre-translation expansion.

Parallel Collections

To access foreign language content, one must attempt the translation of query words into another language. Oard and Diekema[35] describe four distinct approaches to translation, which may be used individually or in combination: machine translation (MT) systems, bilingual dictionaries, parallel corpora, and reliance on the similarity between closely related languages (Fig. 1). Each method has distinct advantages and disadvantages.

MT systems are easy to use; one merely enters a source language passage as input, and the software produces a translation in the desired target language. A major disadvantage is that MT systems produce only one translation, neglecting to provide information about translation alternatives. MT systems are also black boxes whose internal mechanics are generally unknown; they may have sophisticated or simplistic modules for handling syntax and morphology. The quality of commercial MT systems is considered low, although what appears to be a poor translation to a human user might be perfectly acceptable to an information retrieval system. The biggest problem with using MT systems for CLIR, however, is availability; MT is available for only a few common

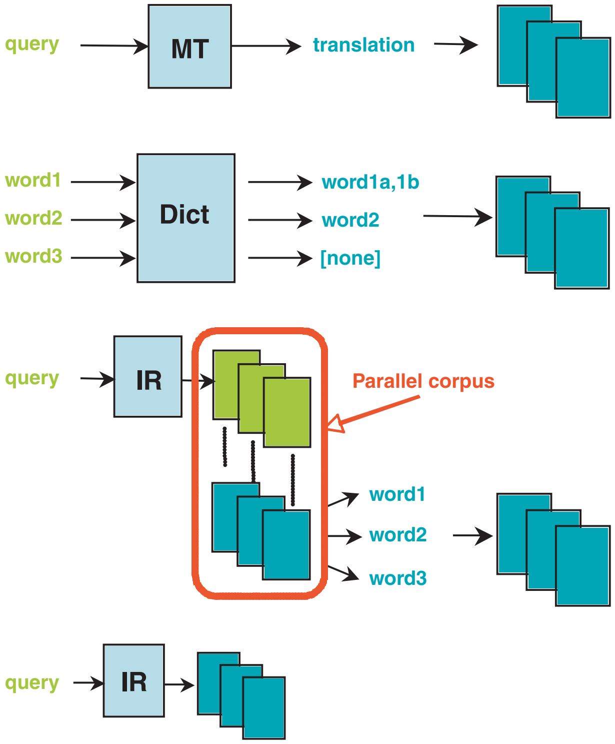


Figure 1. Four approaches to query translation. The first method uses machine translation (MT); the second uses a translation dictionary (Dict); the third uses information retrieval (IR) to retrieve documents that have translations in the target language, then extracts important words from those translations; and the fourth uses the query directly, without translation.

language pairs, and developing an MT system for a new pair of languages is expensive and time-consuming.

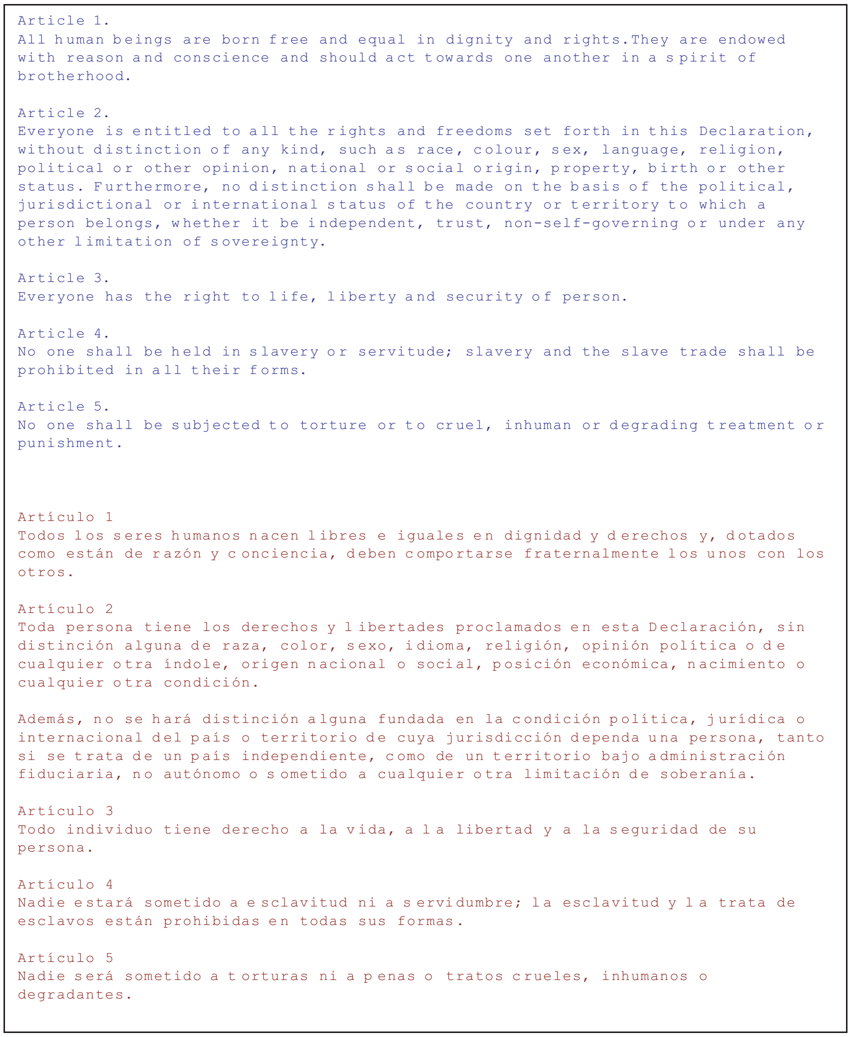
Bilingual dictionaries map words or phrases in one language to translations of those words or phrases in another language. A *wordlist* contains just the mappings; a dictionary also contains extraneous information about words or phrases such as part of speech, etymology, and definitions that are not required to translate a term. One problem with bilingual wordlists is handling multiword phrases because it is difficult for systems to determine what constitutes a compound phrase (e.g., “white box” is not, but “White House” and “black box” are). Other problems include dealing with inflectional forms (e.g., if the wordlist contains an entry for “read” but not for “reading,” how should one translate the latter?), and proper names, which are seldom found in dictionaries.

A bilingual *parallel corpus* is a collection of documents in which each document has an available translation in the other language. Parallel corpora are somewhat rare and expensive to produce; they can be obtained from large multinational organizations (e.g., the UN, NATO, the WHO), international newspapers with a multilingual audience, and works that are translated into many languages such as religious texts. In 1990, Landauer and Littman[36] applied latent semantic indexing (LSI) to the problem of identifying translation candidates for words using the Canadian Hansard Corpus, which consists of transcripts of Canadian parliamentary proceedings in both English and French. Subsequently, there has been great interest in exploiting parallel texts for linguistic applications in general and for translation in particular. Two disadvantages of parallel corpora for translation are their relative rarity, especially for less commonly spoken languages, and the fact that they are often about content in a particular domain, potentially restricting their use for high-quality translation in other domains. A small sample from a parallel text is shown in Fig. 2.

The final approach, *reliance on language similarity*, is a desperate attempt to operate across languages when translation resources are unavailable. Many languages are related to others and may share a large number of words in common because of a common history (e.g., Swedish rule over Norway in the 19th century or language advancement as a result of colonial expansion). This has led to the observation that some rudimentary success in multilingual access may be obtained without translation at all, relying on the fact that some words are lexicographically similar across languages. For example, the word “automobile” has the same meaning in English and French; such words are called cognates. This method tends to perform poorly compared to the alternatives presented here and is only practicable between related languages. (In a later section we describe how we have gotten surprising mileage out of this technique.)

The question remains: given the choice of the above resources, which method or combination of methods is preferable for CLIR? In a series of experiments involving European languages, we determined that lexical coverage is key to accurate CLIR performance;[37] that is, the more comprehensive the translation resource, the better it will do for query translation. Furthermore, we determined that the relationship between size and accuracy is approximately linear, and that automated methods for improving performance when impoverished resources are used can be remarkably effective. We relied on pre-translation query expansion described in the previous section. Pre-translation expansion increases the number of terms in the query. This redundancy improves the robustness of retrieval results when there are gaps in the translation resource. We found that translation using parallel corpora yields excellent results. The remainder of this section describes in more detail how parallel texts can be used for translation.

The first requirement is to obtain reasonably large texts. For a series of tests using Western European languages, we mined parallel texts from the Web, targeting the *Official Journal of the EU* (http://europa.eu.int/). Between December 2003 and April 2004, 80 GB of parallel documents were downloaded. The *Official Journal* documents are mainly legislative texts pertaining to typical governmental functions (e.g., agriculture and foreign trade). Each document is manually translated



Article 1.
All human beings are born free and equal in dignity and rights.They are endowed with reason and conscience and should act towards one another in a spirit of brotherhood.

Article 2.
Everyone is entitled to all the rights and freedoms set forth in this Declaration, without distinction of any kind, such as race, colour, sex, language, religion, political or other opinion, national or social origin, property, birth or other status. Furthermore, no distinction shall be made on the basis of the political, jurisdictional or international status of the country or territory to which a person belongs, whether it be independent, trust, non-self-governing or under any other limitation of sovereignty.

Article 3.
Everyone has the right to life, liberty and security of person.

Article 4.
No one shall be held in slavery or servitude; slavery and the slave trade shall be prohibited in all their forms.

Article 5.
No one shall be subjected to torture or to cruel, inhuman or degrading treatment or punishment.

Artículo 1
Todos los seres humanos nacen libres e iguales en dignidad y derechos y, dotados como están de razón y conciencia, deben comportarse fraternalmente los unos con los otros.

Artículo 2
Toda persona tiene los derechos y libertades proclamados en esta Declaración, sin distinción alguna de raza, color, sexo, idioma, religión, opinión política o de cualquier otra índole, origen nacional o social, posición económica, nacimiento o cualquier otra condición.

Además, no se hará distinción alguna fundada en la condición política, jurídica o internacional del país o territorio de cuya jurisdicción dependa una persona, tanto si se trata de un país independiente, como de un territorio bajo administración fiduciaria, no autónomo o sometido a cualquier otra limitación de soberanía.

Artículo 3
Todo individuo tiene derecho a la vida, a la libertad y a la seguridad de su persona.

Artículo 4
Nadie estará sometido a esclavitud ni a servidumbre; la esclavitud y la trata de esclavos están prohibidas en todas sus formas.

Artículo 5
Nadie será sometido a torturas ni a penas o tratos crueles, inhumanos o degradantes.

Figure 2. Sample from the Universal Declaration of Human Rights in English and Spanish[37] (source: United Nations, http://www.unhchr.ch/udhr/index.htm). Such *parallel texts* allow words in one language to be translated into another without an explicit translation dictionary.

and produced in the official EU languages (11 languages prior to May 2004; 20 following EU enlargement, including diverse languages such as Maltese, Polish, and Turkish). These documents were converted to plain text using publicly available software (pdftotext). In this fashion we obtained approximately 500 MB of aligned text per language—roughly 85 million words in each language. At this writing, this is one of the largest parallel collections ever produced for linguistics research; the next largest we are aware of was produced from oral parliamentary debate and consists of about 160 MB per language.[38]

Once text files have been obtained in this fashion, it remains to align document subsections, index the respective collections, and induce candidate translations. Documents are typically segmented into paragraphs or sentences: sentence-splitting approaches can achieve 99% accuracy despite the myriad uses of the period in English. Alignment is the process of identifying for a paragraph or sentence the corresponding passage in the other language. Algorithms have been proposed that make use of the fact that short passages (measured in characters or words) typically translate into short passages, whereas long passages are translated into longer ones. We used the *char_align* software developed by Church[39] to identify correspondences, then ordered the aligned document fragments and gave each a unique identifier. These documents were indexed using the HAIRCUT retrieval system.

Most parallel collections are bilingual; however, working with 10 EU languages, we could have potentially aligned all possible pairs (45 cases). If English is assumed to be one of the languages of interest, alignments can be obtained for English and each of the other nine languages. Using HAIRCUT we can quickly identify a document translation (or mate) and begin the process of calculating translations.

Our algorithm for extracting translation pairs from parallel collections such as the Universal Declaration of Human Rights is as follows:

For each word in the source language collection, *SCol*,

1. Identify the set *S* of documents containing that word in *SCol*
2. Identify the set *T* of corresponding documents in *TCol*
3. Determine the terms occurring in *T* with the strongest statistical association
4. Output the *k* highest scoring terms as translations

This algorithm is general and may be customized in Step 3 to use a variety of statistical associations. We use our affinity statistic for this purpose. Other viable measures include mutual information and chi squared, each of which compares a joint probability (the likelihood of being found in a document in *T*) to the overall frequency in a language (the likelihood of being found in a document in *SCol* or *TCol*). Although mappings between words are typically extracted, there is nothing to preclude the derivation of other types of mappings when alternative indexing methods are used. For example, if *n*-grams are used, statistical relationships between *n*-grams in one language and those in a different language can be identified, allowing an *n*-gram query in one language to be "translated" directly into an *n*-gram query in another language.

THE HAIRCUT SYSTEM

HAIRCUT (The Hopkins Automated Information Retriever for Combing Unstructured Text) is a Java-based text retrieval engine developed at APL. We are particularly interested in language-neutral techniques for HAIRCUT because we lack the resources to do significant language-specific work.

HAIRCUT has a flexible tokenizer that supports multiple term types such as words, word stems, and character *n*-grams. All text is read as Unicode using Java's built-in Unicode facilities. For alphabetic languages, the tokenizer is typically configured to break words at spaces, downcase them, and remove diacritics. Punctuation is used to identify sentence boundaries and then removed. Stop structure (the noncontent-bearing part of a user's query such as "find documents that" or "I'm interested in learning about") is then optionally removed. We manually developed a list of 459 English stop phrases to be removed from queries. Each phrase was then translated into the other supported languages using various commercial MT systems. We do not have the means to verify the quality of such non-English stop structure, but its removal from queries seems to improve accuracy.

The resulting words, called *raw words*, are used as the main point of comparison with *n*-grams. They also form the basis for the construction of *n*-grams. A space is placed at the beginning and end of each sentence and between each pair of words. Each subsequence of length *n* is then generated as an *n*-gram. A text with fewer than $n - 2$ characters generates no *n*-grams in this approach. This is not problematic for 4-grams, but 6-grams are unable to respond, for example, to the query "IBM." A solution is to generate an additional indexing term for each word of length less than $n - 2$; however, this is not part of our ordinary processing.

Besides the character-level processing required by the tokenizer, and the removal of our guesses at stop structure, HAIRCUT has no language-specific code. We have occasionally run experiments using one of the Snowball stemmers,[40] which attempt to conflate related words with a common root using language-specific rules, but this is not a regular part of our processing. Nor do we do any decompounding, lemmatization, part-of-speech tagging, chunking, parsing, or other linguistically motivated techniques.

The HAIRCUT index is a typical inverted index; each indexing term is associated with a *postings list* of all documents that contain that term. The dictionary is stored in a compressed B-tree, which is paged to disk as necessary. Postings are stored on disk using gamma compression[41] to reduce disk use. Both document identifiers and term frequencies are compressed. Only term counts are kept in our postings lists; we do not keep term position information. We also store a bag-of-words representation of each document on disk to facilitate blind relevance feedback and term relationship discovery.

Blind relevance feedback for monolingual retrieval, and pre- and post-translation expansion for bilingual retrieval, are accomplished in the same way. Retrieval is performed on the initial query, and the top retrieved documents (typically 20) are selected. The terms in those documents are weighted according to our affinity statistic. The highest-weighted terms (typically 50) are then selected as feedback terms.

EXPERIMENTS

Experiments that investigate the retrieval performance of a system or algorithm require a test set containing a set of information needs (i.e., user queries), a fixed collection of documents, and judgments establishing which documents are relevant to each query. In the past, small test collections containing only a few thousand documents were used for information retrieval experimentation. The advantage of these small collections was that each document could be examined manually to determine whether it was responsive to a given query.

In 1991, the first large-scale evaluation was undertaken in the United States, using a collection of over a half-million newspaper articles. Instead of exhaustively analyzing each document for relevance to each query, top-ranked documents from competing retrieval systems were pooled, and only this small set of documents was examined for relevance. Such pooling permits evaluation over large collections by making the assumption that unseen documents are not relevant. Although the inexhaustiveness of the judgments is a natural concern, repeated statistical analyses have shown that the resulting test set judgments can be reliably used to compare systems.

There are currently three international evaluations that promote information retrieval research. TREC, which is in its 14th year, operates in the United States under the auspices of the National Institute for Standards and Technology. It has pioneered the investigation of many aspects of document retrieval, including retrieval against speech archives, retrieval of Web documents, retrieval of foreign language documents, and open-domain question answering. The second evaluation is CLEF, which is organized in and funded by the European Union. It naturally concentrates on research involving European languages. The third evaluation is based in Japan and is called NTCIR (for NII-NACSIS Test Collection for Information Retrieval); it is in its fifth cycle and is run by the Japanese National Institute of Informatics. APL has participated in numerous TREC, CLEF, and NTCIR workshops.

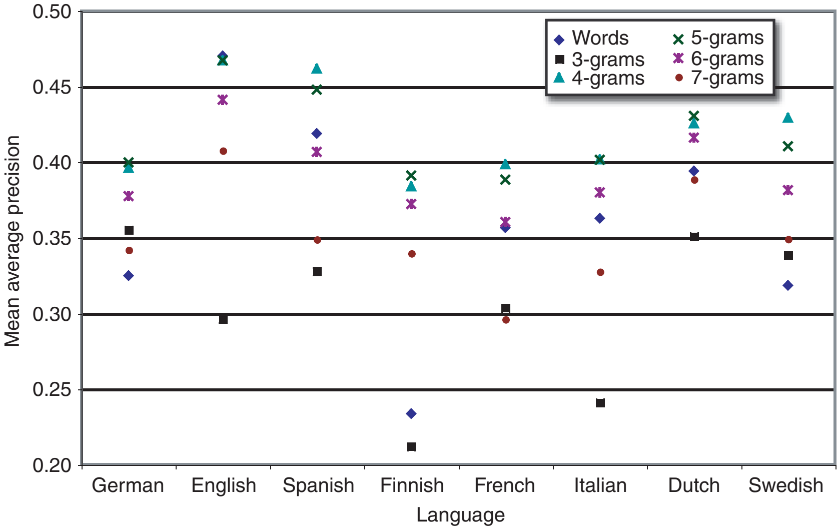


Figure 3. Comparison among various *n*-gram lengths and the use of ordinary words as indexing terms across eight languages. The 4-grams and 5-grams routinely result in the highest performance.

To evaluate retrieval system performance, the measures of precision, i.e., the percentage of retrieved documents that are relevant, and *recall*, i.e., the percentage of the relevant documents that are retrieved, are typically combined into a single metric called *average precision*. Average precision can be thought of as the area under the curve when precision is plotted against recall. By evaluating performance on a number of topics and averaging performance across them, we obtain *mean average precision*, the most widely used and analyzed retrieval performance measure. Mean average precision is correlated to other more intuitive measures, such as *number of relevant documents in the top 10*; however, it is more sensitive than most other such measures. In the experiments that follow we report mean average precision, which varies between zero (abysmal) and one (exceptional).

Monolingual Use of N-grams

We first investigate the relative performance of character *n*-grams and words for monolingual retrieval. Using the CLEF-2002 document set, which covers eight languages, each collection was indexed using six different representations: character *n*-grams of lengths 3 through 7, and words. Subsequently the set of queries for each respective language was run against the appropriate collection. These results, presented in Fig. 3, showed that character *n*-grams significantly outperformed words as indexing terms, particularly in the more linguistically complex languages. While 3-grams and 7-grams did not typically fare well, 4-grams and 5-grams both performed admirably in every language (see Ref. 42 for additional detail).

It is not only in European languages that *n*-grams have an advantage over words. For the TREC-11 evaluation we investigated monolingual and bilingual retrieval in Arabic. Character *n*-grams showed a greater than 40% advantage over words in monolingual retrieval, which we attribute to Arabic's difficult morphology.

Comparing Translation Resources

When translation is combined with retrieval, there are more factors to account for. Although we show that *n*-grams are still a recommended choice, one must also consider which type of translation resource to use.

In this evaluation, the four translation methods described in Fig. 1 were compared against a monolingual baseline as follows. Taking the CLEF-2001 English document collection, topic statements were translated in Dutch, French, German, Italian, and Spanish. Words were used to index the documents. Three different lengths of topics were also applied: keywords (short), sentence (medium), and paragraph (long). Finally, pre-translation expansion, which is applicable for wordlists and parallel corpora, was considered. Results using the German topics are shown in Fig. 4.

Several conclusion can be drawn from Fig. 4. First, two intuitive facts are shown: retrieval using the English queries is better than the translated German ones, and longer queries are more accurate than shorter ones. Regarding translation resources, the use of commercial MT software was found to result in higher performance than dictionaries or parallel corpora, and each was superior to reliance on language similarity. However, when pre-translation expansion was applied, performance using dictionaries and parallel corpora outperformed MT. For common languages, relative bilingual performance 90% as good as monolingual performance was feasible.

Pre-Translation Expansion

As mentioned previously, the use of pre-translation expansion had created some confusion in the information retrieval community. While some practitioners found it beneficial and advocated its use, the results were not uniform. To clarify this situation we investigated the use of pre-translation expansion, post-translation expansion (pseudo-relevance feedback), a combination of both types of expansion, and no expansion. In particular, we measured how retrieval performance depended on the caliber of translation resource used. This was accomplished by synthetically degrading two types of resources (bilingual wordlists and parallel corpora) by randomly failing to translate some words. The two translation resources were degraded in 10% increments up to 100% degradation, which corresponds to no translation at all. The use of such a resource depends entirely on language similarity.

This method was applied to five language pairs using the CLEF-2001 test set;[37] Fig. 5 plots the results obtained when Dutch queries were translated to search English documents using a parallel corpus. Post-translation expansion alone was not found to be very helpful, but pre-translation expansion conveyed a large benefit in the majority of cases. The use of both pre-translation and post-translation expansion could yield a marginal improvement. The drop in performance due to weaker resources was approximately a linear function of translation resource quality, suggesting that pre-translation expansion mitigated losses caused by translation inaccuracies. In additon, pre-translation expansion was noticeably effective when resource quality was poor, suggesting that the technique could be particularly useful in situations where a retrieval capability is needed quickly for a rare language (e.g., in a humanitarian crisis in a Third World country).

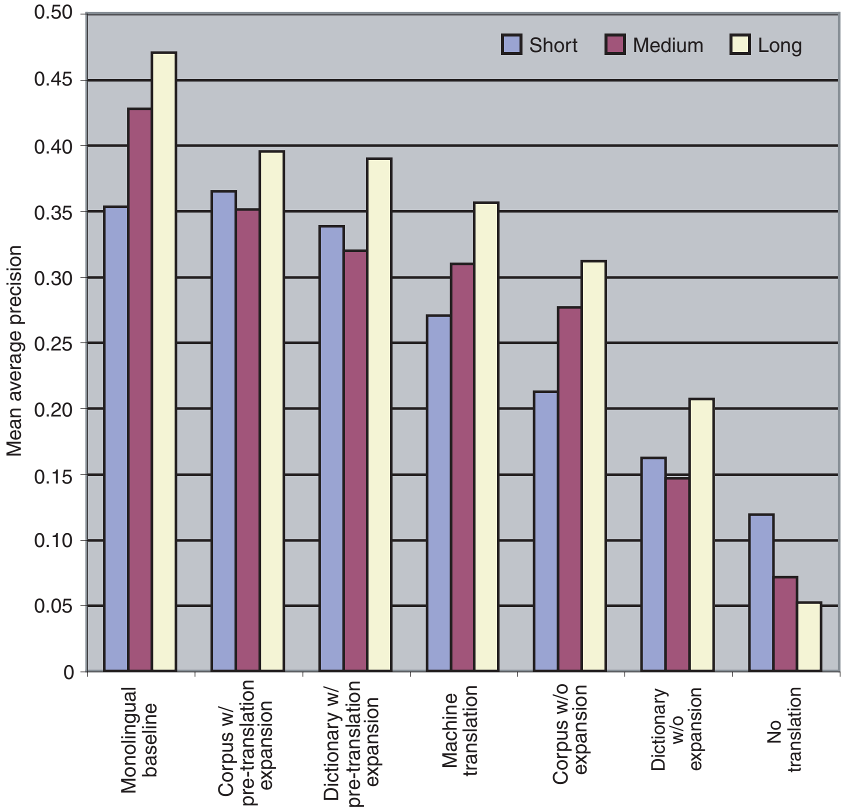


Figure 4. Comparison of six translation methods and an English monolingual baseline across three query lengths. The bilingual experiments used German queries to search English documents from the CLEF-2001 test set.

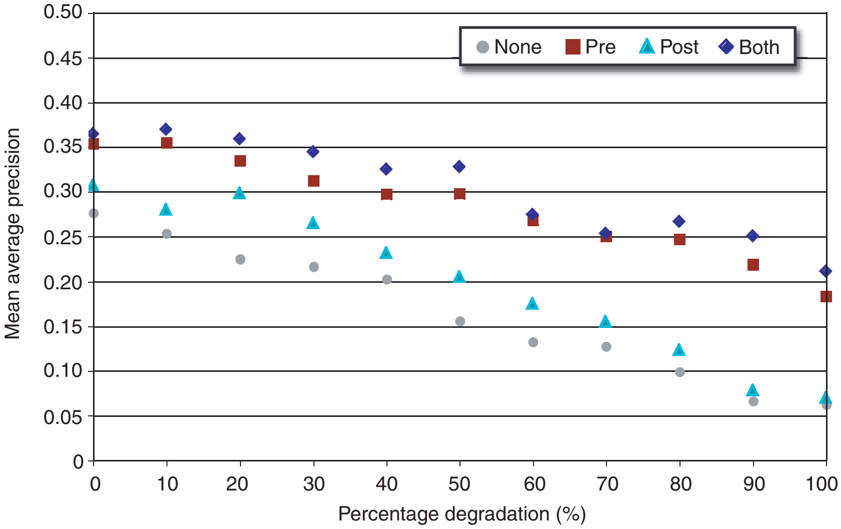


Figure 5. Retrieval performance for various combinations of pre-translation and post-translation expansion when translation resources are degraded. Pre-translation expansion is effective in the majority of cases; however, the relative gain depends on the quality of the available translation resources.

Cross-Language Retrieval Without Translation

Finally, to continue this theme of performing retrieval when resources are very poor, we considered CLIR without translation. The method shows most promise between closely related languages and could be used for transitive translation, where a query is translated into one or more intermediate languages before being translated into the language of the target document collection. We investigated this approach using English and Spanish subcollections from the CLEF 2002 test set. Topics in nine languages were used, so there was a single monolingual run and eight bilingual runs. The effectiveness of three tokenization methods was compared: 4-grams, 6-grams, and words.

Mean average precision is reported in Fig. 6. Performance using the untranslated queries varied by language, as expected. For example, the Portuguese and Italian queries did exceedingly well on Spanish documents; this makes sense as all are Romance languages.

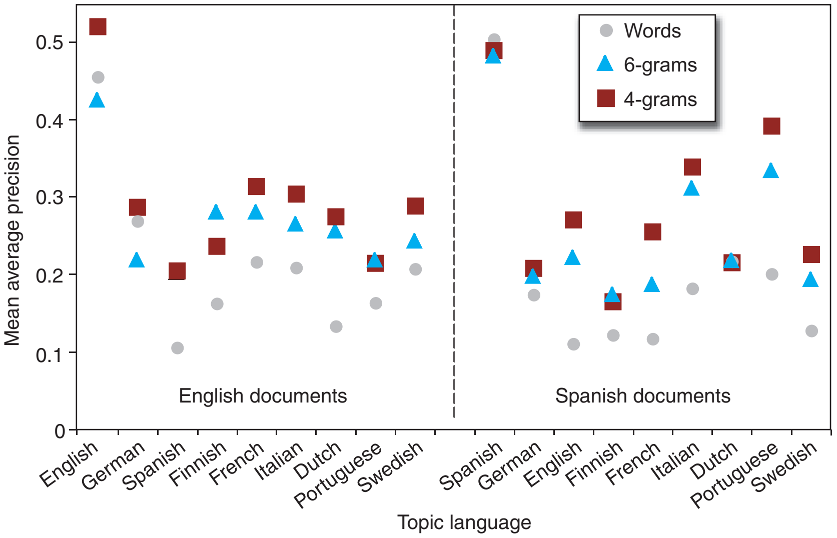


Figure 6. Viability of cross-language retrieval with no translation resources. Performance of no translation retrieval from queries in various European languages to English or Spanish documents is shown. The use of 4-grams outperforms the use of 6-grams or words, and obtains bilingual performance of 50 to 60% of a monolingual baseline. (From Ref. 42, Fig. 2; reprinted with permission.)

In general, performance approached 50 to 60% of that of a monolingual baseline when 4-grams were used (90% relative performance is considered good when translation is applied). The 4-grams were more effective than the longer 6-grams, which is natural, since longer *n*-gram sequences will have fewer matches in morphology in both related and unrelated languages. The 4-grams also exhibited performance 50% or more above that of words. In more recent work, we have examined the benefits of using multiple intermediate languages to improve on this approach.[43]

CLIR without translation is not a preferred scenario; however, it could be the only option when no translation resources are available. For example, translating English to Galician (a minority language spoken by 3 million inhabitants in northern Spain) may not be feasible, but our results suggest that translation of English to Spanish or Portuguese, followed by untranslated retrieval on Galician documents, might work quite well.

CONCLUSION

Through participation in the TREC, CLEF, and NTCIR evaluations, retrieval performance was investigated using document collections in Arabic, Chinese, Dutch, English, Finnish, French, German, Italian, Japanese, Korean, Portuguese, Russian, Spanish, and Swedish. We have found overwhelming support for the contention that high performance is possible without dependence on language-specific approaches. The HAIRCUT system has been developed using five language-neutral techniques: *n*-gram tokenization, affinity sets, a language model similarity metric, pre-translation query expansion, and exploitation of parallel collections. These techniques are effective across a wide range of languages as evidenced by HAIRCUT's consistently high performance in international evaluations. We believe that the techniques described here can help intelligence analysts handle future crises—whatever the language requirements.

REFERENCES

[1] Cross-Language Evaluation Forum (CLEF) Web site, http://www.clef-campaign.org/.

[2] Text REtrieval Conf. (TREC) Web site, http://trec.nist.gov/.

[3] Shannon, C., "Scientific Aspects of Juggling," in *Claude Elwood Shannon: Collected Papers*, N. J. A. Sloane and A. D. Wyner (eds.), IEEE Press (1993).

[4] Shannon, C., "A Mathematical Theory of Communication," *Bell Syst. Tech. J.* **27**, 379–423 and 623–656 (1948).

[5] De Heer, T., "Experiments with Syntactic Traces in Information Retrieval," *Inf. Storage Retriev.* **10**, 133–144 (1974).

[6] Willett, P., "Document Retrieval Experiments Using Indexing Vocabularies of Varying Size. II. Hashing, Truncation, Digram and Trigram Encoding of Index Terms," *J. Doc.* **35**, 296–305 (1979).

[7] De Heer, T., "The Application of the Concept of Homeosemy to Natural Language Information Retrieval," *Inf. Process. Mgmt.* **18**, 229–236 (1982).

[8] Mah, C. P., and D'Amore, R. J., "Complete Statistical Indexing of Text by Overlapping Word Fragments," *ACM SIGIR Forum* **17**(3), 6–16 (1983).

[9] D'Amore, R. J., and Mah, C. P., "One-time Complete Indexing of Text: Theory and Practice," in *Proc. 8th Annu. Int. ACM SIGIR Conf. on R&D in Information Retrieval*, Grenoble, France, pp. 155–164 (1985).

[10] Teufel, B., "Natural Language Documents: Indexing and Retrieval in an Information System," in *Proc. 9th Int. Conf. on Information Systems*, Minneapolis, MN, pp. 193–201 (1988).

[11] Comlekoglu, F. M., *Optimizing a Text Retrieval System Utilizing N-gram Indexing*, Ph.D. Thesis, George Washington University (1990).

[12] Cavnar, W. B., "Using an N-gram-based Document Representation with a Vector Processing Retrieval Model," in *Proc. Third Text REtrieval Conf. (TREC-3)*, NIST Special Publication 500-226, D. K. Harman (ed.), pp. 269–278 (1994).

[13] Damashek, M., "Gauging Similarity with N-grams: Language-Independent Categorization of Text," *Science* **267**, 843–848 (1995).

[14] Harman, D., Buckley, C., Callan, J., Dumais, S., Lewis, D., et al., "Performance of Text Retrieval Systems," *Science* **268**, 1417–1418 (1995).

[15] Cavnar, W. B., and Trenkle, J. M., "N-Gram Based Text Categorization," in *Proc. Third Sym. on Document Analysis and Information Retrieval*, Univ. of Nevada, Las Vegas, pp. 161–169 (1994).

[16] Zamora, E. M., Pollock, J. J., and Zamora, A., "The Use of Trigram Analysis for Spelling Error Detection," *Inf. Process. Mgmt.* **17**, 305–316 (1981).

[17] Cohen, J. D., "Highlights: Language- and Domain-Independent Automatic Indexing Terms for Abstracting," *J. Am. Soc. Inf. Sci.* **46**, 162–174 (1995).

[18] Mihalcea, R., and Nastase, V., "Letter Level Learning for Language Independent Diacritics Restoration," in *Proc. 6th Conf. on Natural Language Learning (CoNLL-2002)*, pp. 105–111 (2002).

[19] Pearce, C., and Nicholas, C., "TELLTALE: Experiments in a Dynamic Hypertext Environment for Degraded and Multilingual Data," *J. Am. Soc. Inf. Sci.* **47**, 236–275 (1996).

[20] Chen, A., He, J., Xu, L., Gey, F., and Meggs, J., "Chinese Text Retrieval Without Using a Dictionary," in *Proc. 20th Annu. Int. ACM SIGIR Conf. on R&D in Information Retrieval*, Philadelphia, PA, pp. 42–49 (1997).

[21] Lee, J. H., and Ahn, J. S., “Using N-grams for Korean Text Retrieval,” in *Proc. 19th Annu. Int. ACM SIGIR Conf. on R&D in Information Retrieval*, Zurich, Switzerland, pp. 216–224 (1996).

[22] Ogawa, Y., and Matsuda, T., “Overlapping Statistical Word Indexing: A New Indexing Method for Japanese Text,” in *Proc. 20th Annu. Int. ACM SIGIR Conf. on R&D in Information Retrieval*, Philadelphia, PA, pp. 226–234 (1997).

[23] Salton, G., *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*, Addison Wesley (1989).

[24] Qiu, Y., and Frei, H. P., “Concept Based Query Expansion,” in *Proc. 16th Annu. Int. ACM SIGIR Conf. on R&D in Information Retrieval*, Pittsburgh, PA, pp. 160–169 (1993).

[25] Church, K. W., and Hanks, P., “Word Association Norms, Mutual Information, and Lexicography,” *Comput. Linguist.* **16**(1), 22–29 (1990).

[26] Ponte, J. M., and Croft, W. B., “A Language Modeling Approach to Information Retrieval,” in *Proc. 21st Annu. Int. ACM SIGIR Conf. on R&D in Information Retrieval*, Melbourne, Australia, pp. 275–281 (1998).

[27] Miller, D., Leek, T., and Schwartz, R., “A Hidden Markov Model Information Retrieval System,” in *Proc. 22nd Annu. Int. ACM SIGIR Conf. on R&D in Information Retrieval*, Berkeley, CA, pp. 214–221 (1999).

[28] Hiemstra, D., *Using Language Models for Information Retrieval*, Ph.D. Thesis, Center for Telematics and Information Technology, The Netherlands (2000).

[29] Jelinek, F., and Mercer, R., “Interpolated Estimation of Markov Source Parameters from Sparse Data,” in *Pattern Recognition in Practice*, E. S. Gelsema and L. N. Kanal (eds.), North Holland, pp. 381–402 (1980).

[30] Robertson, S. E., Walker, S., and Beaulieu, M., “Okapi and TREC-7: Automatic ad hoc, Filtering, vlc, and Interactive,” in *Proc. 7th Text REtrieval Conf. (TREC-7)*, NIST Special Publication 500-242, E. M. Voorhees and D. K. Harman (eds.), pp. 253–264 (Aug 1999).

[31] Salton, G., and Buckley, C., “Term-Weighting Approaches in Automatic Text Retrieval,” *Inf. Process. Mgmt.* **24**(5), 513–523 (1988).

[32] Savoy, J., “Cross-Language Information Retrieval: Experiments Based on CLEF 2000 Corpora,” *Inf. Process. Mgmt.* **39**(1), 75–115 (2003).

[33] Ballasteros, L., and Croft, W. B., “Phrasal Translation and Query Expansion Techniques for Cross-Language Information Retrieval,” in *Proc. 20th Annu. Int. ACM SIGIR Conf. on R&D in Information Retrieval*, Philadelphia, PA, pp. 84–91 (1997).

[34] Gey, F., and Chen, A., “TREC-9 Cross-Language Information Retrieval (English-Chinese) Overview,” in *Proc. Ninth Text REtrieval Conf. (TREC-9)*, E. M. Voorhees and D. K. Harman (eds.), pp. 15–23 (2001).

[35] Oard, A., and Diekema, A., “Cross-Language Information Retrieval,” *Annu. Rev. Inf. Sci. Technol.* **3**, 223–256 (1998).

[36] Landauer, T. K., and Littman, M. L., “Fully Automated Cross-Language Document Retrieval Using Latent Semantic Indexing,” in *Proc. 6th Ann. Conf. of the UW Centre for the New Oxford English Dictionary and Text Research*, pp. 31–38 (1990).

[37] McNamee, P., and Mayfield, J., “Comparing Cross-Language Query Expansion Techniques by Degrading Translation Resources,” in *Proc. 25th Annu. Int. Conf. on R&D in Information Retrieval*, Tampere, Finland, pp. 159–166 (2002).

[38] Koehn, P., “Europarl: A Multilingual Corpus for Evaluation of Machine Translation,” http://people.csail.mit.edu/people/koehn/publications/europarl/.

[39] Church, K. W., “Char_align: A Program for Aligning Parallel Texts at the Character Level,” in *Proc. 31st Annu. Mtg. of the Assoc. for Computational Linguistics*, Columbus, OH, pp. 1–8 (1993).

[40] Snowball Stemmer Web site, http://snowball.tartarus.org/ (Sep 2002).

[41] Witten, I. H., Moffat, A., and Bell, T. C., *Managing Gigabytes: Compressing and Indexing Documents and Images*, 2nd Ed., Morgan Kaufmann Publishers (1999).

[42] McNamee, P., and Mayfield, J., “Character N-gram Tokenization for European Text Retrieval,” *Inf. Retriev.* **7**(1-2), 73–97 (2004).

[43] Mayfield, J., and McNamee, P., “Triangulation Without Translation,” in *Proc. 27th Annu. ACM SIGIR Conf. on R&D in Information Retrieval*, Sheffield, UK, pp. 490–491 (2004).

ACKNOWLEDGMENT: Parts of this article were taken from Ref. 42, specifically, all of Section 2 (pp. 74, 75), half of Section 3 (pp. 75, 76), and Fig. 2 (p. 89). This material was reproduced with the kind permission of Springer Science and Business Media, © 2004 Kluwar Academic Publishers.

THE AUTHORS

The HAIRCUT Information Retrieval Project, which routinely places among the top systems in the world in international evaluations of cross-language retrieval, is led by **James C. Mayfield**. Prior to joining APL in 1996, Dr. Mayfield was Associate Professor of Computer Science at the University of Maryland, Baltimore County. He is now a member of APL’s Principal Professional Staff, serves as the Supervisor of the Distributed Information Systems Section in the Research and Technology Development Center’s RSI Group, and is an Associate Research Professor in the Whiting School of Engineering. **Paul McNamee** is a Senior Computer Scientist in the Research and Technology Development Center and has worked at APL since 1991. He is currently a Ph.D. student at the University of Maryland, Baltimore County, where he conducts research in information retrieval and multilingual information access. He serves as an Adjunct Faculty member in The Johns Hopkins University’s Part-Time Program in Computer Science, where he has taught courses in text retrieval, artificial intelligence, and Web-based development. The HAIRCUT team has produced more than 40 articles, papers, and patents in the area of human language technologies. Further information on human language technologies research at APL can be obtained from Dr. Mayfield. His e-mail address is james.mayfield@jhuapl.edu.

James C. Mayfield

Paul McNamee