IIT at TREC-2002

Linear Combinations Based on Document Structure and Varied Stemming for Arabic Retrieval

Information Retrieval Laboratory
Department of Computer Science
Illinois Institute of Technology
Chicago, IL 60616
Abdur Chowdhury, Mohammed Aljlayl, Eric Jensen, Steve Beitzel,
David Grossman, Ophir Frieder
{abdur, aljlayl, ej, steve, grossman, ophir}@ir.iit.edu

Abstract
For TREC 10 we participated in the Named Page Finding Task and the Cross-Lingual Task. In the web track, we explored the use of linear combinations of term collections based on document structure. Our goal was to examine the effects of different term collection statistics based on document structure in respect to known item retrieval. We parsed documents into structural components and built specific term indexes based on that document structure. Each of those indices have their own collection statistics for term weighting based on the type of language used for that structure in the collection. For producing a single ranked list, we examined a weighted linear combination approach to merging results. Our approach to known item retrieval was equal or above the median 58% of the time and 71% above the mean score of submitted runs. In the Arabic track we participated in Arabic Cross-language Information Retrieval (CLIR) and in Arabic monolingual information retrieval. For the monolingual retrieval, we examined the use of two stemming algorithms. The first is a deeper approach, and the second is a pattern-based approach. For the Arabic CLIR, we explored the retrieval effectiveness by using a machine translation (MT) system and translation probabilities obtained from parallel documents collection provided by the United Nations (UN).

Keywords: Known-item search, document structure retrieval, linear combination of retrieval strategies, cross-lingual Arabic retrieval, light-stemming, pattern-based stemming

Named Page Finding Task

Many years of research have been devoted to examining the question of what are the best retrieval strategies for retrieving information, this year we explore a variation on the task where a specific or known-item is sought after given a query or topic. Our research this year specifically explores three basic questions about this task:

How do document structure approaches compare to traditional ranking strategies given that the task and evaluation metrics have changed?

1. What type of document structure can be exploited to improve the effectiveness of this task, in comparison to traditional approaches?

2. How effective are weighted linear combination approaches to combining evidence from document structure retrieval approaches.

Many ranking strategies have been examined in the past. Three of the most studied algorithms are PDLN (Pivoted Document Length Normalization) [1], Okapi BM25 [2], Self-Relevance [3] due to their effectiveness in prior TREC evaluations. In our calibrations, we have found BM25 to perform well so we use it as a baseline.

Some work has already been done on the extraction and storage of HTML term information [4]. Additionally, much has been done with the use of link information to identify hubs and authorities [5]. Since many content developers use HTML elements/tags to improve the readability of their documents, we hypothesize that simply using these tags may improve effectives. There are many different tags that could be used, (e.g.; title, section headers, anchor text, bold, underlines, comments, etc.), but we initially focus on only three types: title, anchor text and text.

Finally, we examine the fusion of different document structure indexes to produce a single ranked list for the know-item task. Those different document representations can be merged with linear combinations maximizing mutual evidence. When combining evidence we extend prior research of weighted linear combination approaches.

In recent years, the category of work known as data fusion or multiple-evidence described a range of techniques in information retrieval whereby multiple pieces of information are combined to achieve improvements in retrieval effectiveness. These pieces of information can take many forms including different query representations, different document representations, and different retrieval strategies used to obtain a measure of relationship between a query and a document. Several researchers have used combinations of different retrieval strategies to varying degrees of success in their systems [6, 7]. Belkin, et al. examined the effects of combining several different query representations to achieve improvements in effectiveness [8, 9]. Lee examined the effect of using different weighting schemes to retrieve different sets of documents using a single query and document representation, and a single retrieval strategy [10].

Fox and Shaw examined combination algorithms that increase the score of a document based on repeated evidence of its relevance, as done in [6]. One of the algorithms designed by Fox and Shaw, CombMNZ, has proven to be a simple, effective method for combining result sets. It was used by Lee in his fusion experiments, and has become the standard by which newly developed result combination algorithms are judged. More recent research in the area of meta-search engines has led to the proposal of several new result combination algorithms of even greater complexity, making use of training data and techniques such as voting algorithms and Bayesian inference [11, 12, 13]. Although these algorithms were shown to behave comparably and occasionally superior to CombMNZ, for our research we use Fox's CombMNZ algorithm, leaving other linear combination approaches as a topic of further research.

In the next section we describe our experimental approach to examine the above questions. In the results section we present our results from this year's experiments. Lastly, we conclude and present future possible research directions

Methodology

To conduct our research we use the IIT retrieval system AIRE [14]. Our system builds a traditional inverted index based on a given document structure(s). Additionally, our system uses conflation classes [15] instead of a more commonly used stemmer such as Porter [16]. Those classes have been modified over the years as problem term variants have been encountered. Additionally, AIRE uses a generated statistical phrase list, where the statistical phrases were generated with a news collection and IDF filtering to reduce the final phrase list size. Phrases are generated from phrases via a bi-gram sliding window algorithm and weighted with 25% importance in relation to keyword weighting for retrieval. Basic term weighting uses the Okapi BM25, Equation 1.

$$\sum \log\left(\frac{(N-n)+.5}{(n+.5)}\right) * \left(\frac{(k1+1)*tf}{(K+tf)} * \frac{(k3+1)*qtf}{(k3+qtf)}\right)$$

$$K = k1*((1-b)+b*dl/avdl)$$

Equation 1: Okapi BM25

Where:
- *tf = frequency of occurrences of the term in the document*
- *qtf = frequency of occurrences of the term in the query*
- *dl = document length*
- *avdl = average document length*
- *N = is the number of documents in the collection*
- *n = is the number of documents containing the word*
- *k1 = 1.2*
- *b = 0.75 or 0.25 (we use .25)*
- *k3 = 7,* set to 7 or 1000, controls the effect of the query term frequency on the weight -- smaller is less.

We indexed the 18GB government collection producing a full-text index, HTML title term index, and an anchor text index. The anchor text index differed from the other indexes, in that an additional mapping stage was required so referencing anchor text data can be linked to the referenced TREC document name. For our experimental layout we first produce a baseline run based on BM25, conflation classes, phrases, full-text index, referred to as the (base) run with the results summarized in Table 4.

Two additional result sets were created; the first one was produced using only the title index and the second produced from only using the anchor text index. With those three indexes and result sets our original three questions can be examined. With a baseline result set, additional document structure techniques can be compared in relation to each other (our first question). Our second question we briefly explore by examining anchor text and title text in relation to full text retrieval. In the next section we present results examining the effectiveness of the various structures and their combinations (our third question) with respect to baseline ad-hoc retrieval strategies.

Our linear combination is a three-step process. First our scores are normalized from each document representation retrieved set using min-max normalization, Equation 3. The advantage of this method is that it preserves all relationships of the data values exactly. It does not introduce any potential bias into the data. Secondly, the final scores are calculated using CombMNZ, Equation 2. Where each individual score is biased via alpha and beta weights assigned to the document structure.

CombMNZ = SUM(Individual Similarities) * Number of Nonzero Similarities

Equation 2: CombMNZ

$$V' = (V - min) * (new\_max - new\_min) / (max - min) + new\_min$$

Equation 3: Min-Max Normalization

For our linear combination experiments we did not have relevance judgments, thus for our submitted runs we submitted runs based on guesses for the best weighting of linear combinations. Additionally, we limited the combinations of results and weighting to the experiment show in Figure 1.

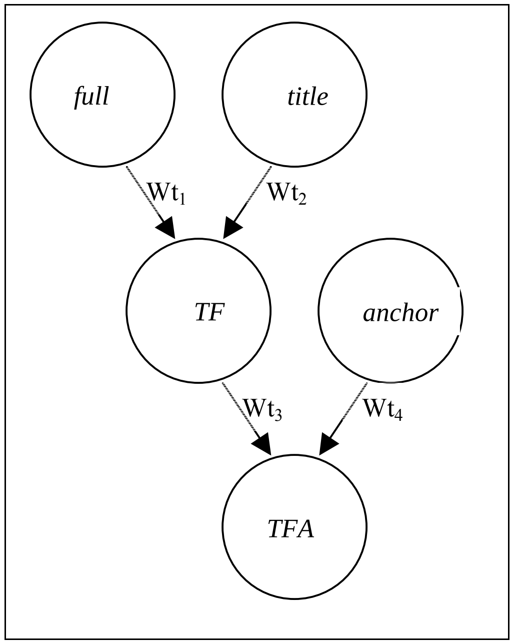

Figure 1: Linear Combination Hierarchy

Results

Our first sets of results examine the retrieval effectiveness of the various document structure elements. In Table 1 we see that the full text index significantly outperforms anchor and title indexes. This is not all that surprising given that anchor retrieval depends on the vocabulary of the referring text, thus not all documents have suitable referring text for the given query set. Similarly, title only retrieval is dependant on the author's vocabulary meeting the query language for this query set and is a subset of the vocabulary used for full text thus performing less than full text is not unexpected.

While the vocabulary for title and anchor text alone may not be as effective as full text for this task/query set, we further examine if combinations of the structures can improve effectiveness. In the second set of experiments we fused the title and full text indices with the CombMNZ algorithm with various alpha, beta weights for each document representation index. Table 2 displays the results of those experiments, while we did not have the final qrels, the run we chose ended up being the best combination of the two retrievals, (alpha=.2 and beta=.8) as highlighted in the table. Effectiveness improvements with any combination of full text and title text retrieval are not found. While a slight improvement is found in the top 10, this does not seem to be a significant improvement of any type.

	Full Text	Anchor	Title
MRR	.587	.156	.323
In Top 10	111	31	67
Found	128	40	82

Table 1: Document Structure Index Runs

MRR	T10	Found	α	β
.402	82	134	.9	.1
.421	90	134	.8	.2
.446	102	134	.7	.3
.468	108	134	.6	.4
.502	109	134	.5	.5
.545	110	134	.4	.6

.559	110	134	.3	.7
.572	***112***	***134***	***.2***	***.8***
.578	111	133	.1	.9

Table 2: Title & Full Fusion, Title = α, Full= β

MRR	T10	Found	α	β
.576	**114**	**134**	**.9**	**.1**
.565	119	134	.8	.2
.539	117	134	.7	.3
.441	115	134	.6	.4
.391	108	134	.5	.5
.297	86	133	.4	.6
.268	68	133	.3	.7
.246	50	133	.2	.8
.218	40	133	.1	.9

Table 3: TF & Anchor Fusion, FT = α, anchor= β

We then fused the combined evidence from (full text and title text) with anchor information and explored various weighting variables. Our submitted run, was the most effective in terms of MRR, but did not yield the greatest number in the top 10. Although the MRR is slightly worse than our full text approach, the number of correct results in the top 10 and "found" increased slightly. After receiving the relevance judgments from NIST we explored other various combination orderings, but found no improvements or negative effects for various orderings of fused results in retrieval effectiveness for all combinations.

These results are surprising given that most popular search engines use document structure for improving the effectiveness of their services. While their improvements may come for other aspects of their approach, using the information as we did showed no significant advantage.

	Base	**TF**	**TFA**
MRR	.587	.576	.58
In Top 10	111/74%	114/76%	117/78%
Not Found	22/14.6%	20/13.3%	19/12.6%
>= Mode	82/54.6%	80/53.3%	75/50%
=>Median	88/58.6%	92/61.3%	87/58%
>= Mean	107/71.3%	104/69.3%	108/72%

Table 4: Submitted Result Summary

Our baseline full text retrieval approach for the known-item task was 58% of the time equal or above the median and 71% above the mean score of submitted runs. Additionally, our approach produced the item in the top 10 results 74% of the time and only missed the know-item 14% of the time with 150 queries. Our results using document structure marginally improved top 10 and found statistics, but did not improve MRR. These results are rather surprising in that the BM25 approach had been designed, tuned and tested for a different task and metric. While its success validates the robustness of the algorithm, more research needs to be conducted using document structure to determine how that information should be incorporated into the ranking strategy or that it does not benefit know-item retrieval.

Named-Item Summary

For TREC 10 we explored the use of linear combinations of term collections based on document structure features. Our goal was to examine the effects of different term collection statistics based on document structure in respect to known item retrieval. Our approach is to dissect a document into structural parts and build specific term indexes based on that document structure. Each of those indices would have their own collection statistics for term weighting based on the type of language used for that structure in the collection. For producing a single ranked list, we examined a weighted linear combination approach to merging results.

While our document structure linear combination experiments did not yield any promising results, our approach to known item retrieval was 58% of the time equal or above the median and 71% above the mean score of submitted runs. Additionally, our approach produced the item in the top 10 results 74% of the time and only missed 14% of the known-items out of 150 topics.

Cross-lingual Track

In the Arabic track, we participated in Arabic Cross-language Information Retrieval (CLIR) and in Arabic monolingual information retrieval. We dedicated our effort to improve the retrieval effectiveness of Arabic monolingual retrieval, as we believe it is essential for any Arabic IR or CLIR systems. For the monolingual retrieval, we used two stemming algorithms. The first is a deeper light–based approach, and second is pattern-based approach. For the Arabic CLIR, we explored the retrieval effectiveness by using two recommended standard resources. The resources are a machine translation (MT) system and translation probabilities obtained from parallel documents collection provided by the United Nations (UN). The Arabic AIRE retrieval system is used for experimentation. We used the IIT similarity function and Rocchio relevance feedback.

Background

Unlike alphabets based on the Roman script, the orientation of writing in Arabic is from right-to-left. The shape of most of the characters depends on their position within a word and the character adjacent to them. Most Arabic words are morphologically derived from a list of roots. The root is the bare verb form; it can be triliteral, quadriliteral, or pentaliteral. Most of these roots are made up of three consonants. The Arabic language uses a root-and-pattern morphotactics; patterns can be thought of as templates adhering to well-known rules. These patterns generate nouns and verbs. Roots are interdigitated with the patterns to form Arabic surface forms.

Arabic words are classified into three main parts of speech, nouns (including adjectives and adverbs), verbs, and particles. All verbs and some nouns are derived from a root. Arabic sentences are either verbal or nominal. Verbal sentences contain a verb before the subject, and may contain complements. Nominal sentences begin with a subject followed by a noun, an adjective, a prepositional phrase, or an adverb. In formal writing, Arabic sentences are delimited by commas and periods as in English.

Arabic Monolingual Retrieval

Unlike Indo-European languages such as English, the Arabic language is a highly inflected language. From an Arabic root, many surface forms can be derived. The surface forms of a word have a great impact on a language like Arabic with a strong morphology since surface forms comprise at least two morphemes: a three consonantal root conveying semantic meaning and a word pattern carrying syntactic information. Moreover, most connectors, conjunctions, prepositions, pronouns, and possession forms are attached to the Arabic surface form. Retrieving based on surface form results in low retrieval effectiveness as concluded in [17,18].

Another strategy is to retrieve based on the root of the Arabic word. The goal of the root-based stemmer is to detect and extract the root of an Arabic surface word and it requires very deep syntactic analysis. Al-Shalabi [19] developed a system that detects the root and the pattern of Arabic words with verbal roots. Khoja [20] designed and experimented a novel algorithm for root detection. The retrieval based on the roots improves the retrieval effectiveness as compared to the surface form of the Arabic words. As in our earlier efforts [17,18], light stemming outperforms the root-based stemming. Therefore, light stemming approaches have potential promise [17]

A deeper light stemming approach

The aim of this algorithm is to conflate more related terms in a conflation class than the classes produced in [18]. To achieve this goal, we used a training corpus to identify the frequent suffixes and prefixes. The corpus was obtained from two Saudi Arabian newspapers, namely, Alriyadh and Aljazirah from the year 1999 to 2001. This corpus consists of more than one million words that cover a variety of subjects. The maximum length of the prefixes and suffixes is four letters and the minimum is two letters. Considering more than four letters as prefix of suffix results in ambiguous term after stripping them out. Also, one letter is not enough to form a valid suffix or prefix.

This *automatic* algorithm adheres to the following steps:

1- Check whether the given Arabic word is Arabicized,
2- Remove any diacritics in the given Arabic terms,
3- Start an aggressive normalization,
4- Check for the prefix Waw,
5- Check for duplicate prefixes,
6- Detect definite articles,
7- Check for suffixes,
8- Check for prepositions that attached to the given Arabic stem,
9- Check for prefixes,
10- Normalize the Alf-Maksorah and the Alf.

Throughout the above steps, each step is associated with an event, when the event occurs, an action will be taken. The algorithm checks the length of stem to decide whether to fire the associated action. The minimum length of the stem is three letters. Choosing three letters as minimum maintains the semantic of the Arabic word since most Arabic words are built up from three consonants. In Table 5 we describe some candidate suffixes that considered for removal that we obtained from corpus statistics.

Suffix	Example	Meaning
اتهن	معلماتهن	Their teachers (plural feminine)
اتها	معلماتها	Her teachers (singular feminine)
هما	كتابهما	Their book (dual masculine)
يين	مدنيين	Civilians
تين	تفاحتين	Two apples (dual feminine)

Table 5: Some suffixes derived from the corpus

A pattern-based stemming approach

This approach uses patterns to detect the affixes of the given Arabic word. The algorithm starts first to match a pattern on the given Arabic word. For the case of liberal matching mode, if the matched letters are greater than one, then the algorithm considers that pattern as valid then prefixes and the suffixes will be removed. A more restrictive mode can be applied, i.e., increasing the number of matching between the given Arabic terms and the patterns to consider the current pattern for candidacy. The pattern-based algorithm adheres to the following steps:

1. Remove any diacritics in the given Arabic terms.
2. Normalization such as Alf, and Ya-Maksorah.
3. Check for the prefix Waw
4. Check for duplicate prefixes
5. Detect definite articles
6. Match the given Arabic term on a list of patterns. If there is at least one letter match in the given Arabic term, then the algorithm strips out the suffixes and prefixes of that term based on the matched pattern. If the algorithm fails to extract and remove the suffixes or the prefixes from the given Arabic terms, then the algorithm proceeds executing from step 7 to the end.

To clarify the roles of patterns in Arabic morphology, consider the root (كتب). This root is transliterated as "*ktb*", which is measured with pattern (فعل). The pattern (فعل) is transliterated as "*fàl*". "*f*' corresponds to the first letter (ف), *"à"* corresponds to middle letter (ع), and "*l*" corresponds to last letter (ل). The pattern preserves *f*, *à*, and *l* in the same order, whereas vowels and other letters can be added to form a pattern. As shown in Table 6, many patterns are derived from the base pattern "*f à l*" of the root "*ktb*". As shown, the pattern "*f à alh*" form the word (كتابه) by attaching the vowel (ا) and letter (ه) to the root "*ktb*". Locating the original letters of the given Arabic word in the pattern is essential step to remove the prefixes and suffixes.

Arabic word	Pattern	Meaning
كاتب	faàl	writer
كتابه	fàalh	writing
الكاتبان	fàl	the two writer
الكاتبين	faàl	The two writers (dual masculine in accusative form)

Table 6: patterns and their surface forms

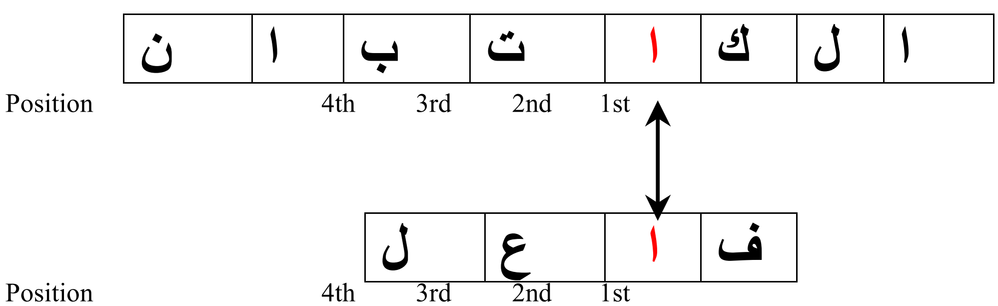

Figure 2. Matching the word "الكاتبان" and the pattern "فاعل"

Figure 2 illustrates the process of matching and stripping out the prefixes and suffixes. Before considering a suffix or prefix for removal, a matching process between the pattern and the given Arabic term is performed. For liberal matching, at least one letter from the pattern should match the Arabic term in same position.

Arabic Cross-language Information Retrieval (CLIR)

In the cross-lingual track, we experimented using the recommended standard resources that are provided by TREC for query translation. We used two means of query translations, machine translation system (MT) and the translation probability that are derived from the UN corpus via BBN [22].

Ajeeb MT system

Machine Translation systems can be defined as any computer-based system that seek automatically to transform a target text from one language into another language by using context information. One of the approaches being used for CLIR is using the existing machine translation system which usually involves automatic translation of the queries, from one language to another. We used ajeeb MT system (www.ajeeb.com) for translating the provided 50 queries (titles and descriptions) from English to Arabic.

Translation probability

Translation probability means that if a term in the source language has several translations in the target language, each term in the target language gets probability. BBN construct translation probabilities that are derived from parallel corpus. The parallel corpus was obtained from the United Nations (UN). The statistical machine translation GIZA++ was used to provide the transaction probabilities. The probability p(a|e) has several terms as candidate for translation, we selected the highest probability for each entry.

Results Analysis

In Arabic monolingual retrieval, our results demonstrate the usefulness of using stemming in improving the retrieval precision. As shown in Table 7, the initial investigation of the pattern-based algorithm, which is in the liberal mode, achieved an improvement over the deeper light stemming algorithm.

	Deeper light stemming	Pattern-based stemming
Average Precision	0.3419	0.3473

Table 7: Average precisions of Deeper light and Pattern-based approaches

In both stemming algorithm, some queries got as close as 0% measured in average precision. The reason behind this drop of retrieval effectiveness is that these queries have fatal error in spelling as well as some has ambiguous term. For example query number 32 got 0% measured in average precision.

صيانة بلوغا في بحرقزوين

The term "بحرقزوين" appears as one single term. As well as, the term "بلوغا" is an ambiguous term. These reasons make our algorithm performed poorly in this query.

Figure 3 demonstrates the average precision at several levels of recalls (0-1) of the pattern-based and deeper light algorithms.

In cross-lingual retrieval, the results of using the translation probabilities performed poorer than machine translation approach as shown in Table 8. The reason behind this drop of retrieval effectiveness is that the construction of the translation probabilities is based on an aggressive stemmer [21] which increases the chance of the co-occurrence of two different terms.

	Machine translation	Statistical translation
Average Precision	0.2453	0.2285

Table 8. Average precisions of Deeper light and Pattern-based approaches

Figure 4 demonstrate the average precision at several levels of recall (0-1), as shown the machine translation system is more effective than the statistical translation.

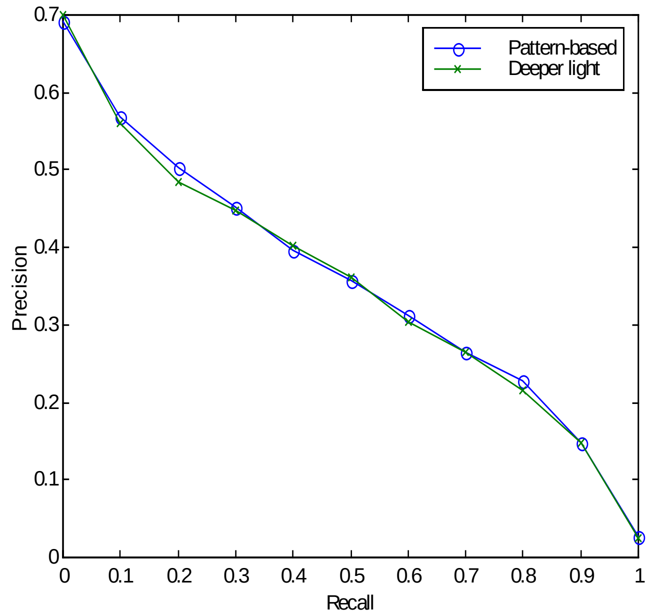

Figure 3. Average precision of pattern-based and deeper light algorithms.

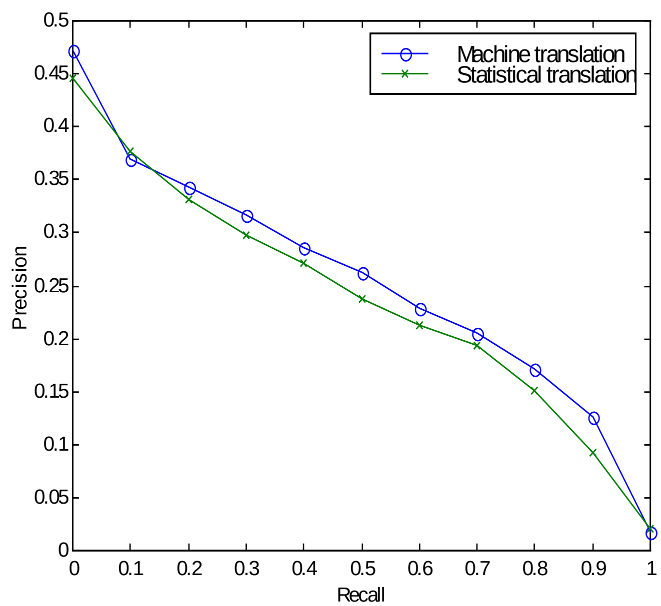

Figure 4. Average precision of machine translation and translations probabilities

CLIR Summary

We showed that stemming is an important approach to improve the retrieval effectiveness of any Arabic information retrieval systems. In TREC-2002, we participated in both monolingual and cross-lingual retrieval. Our focus in this year is on the improvement of Arabic monolingual information retrieval systems. We presented a new automatic algorithm for stemming, namely, the pattern-based. We experimented with this algorithm by using the liberal mode. In future work, we plan to make it more restricted for matching the given Arabic word and the pattern as well as to increase the number of patterns to enhance the rule-based operations of the algorithm.

In cross-lingual retrieval, we experimented with the standard resources that are provided via TREC11. We found that the machine translation system achieved superior performance compared to translation probabilities. One reason for this is that the construction of the translation probabilities derived from the UN parallel corpus is based on an aggressive stemmer. In addition, some terms are not covered for translation. We plan to enhance the quality of the extracted parallel terms and to add more terms for wider coverage.

References

[1] A. Singhal, et al., "Pivoted document length normalization", ACM-SIGIR, 1996.

[2] S. Robertson, et al., "Okapi at TREC-4", NIST TREC-4, November 1995.

[3] K. Kwok, et al., "TREC-7 Ad-Hoc, High precision and filtering experiments using PIRCS", NIST TREC-7, November 1998.

[4] S. Brin, L. Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine", WWW7 / Computer Networks 30(1-7): 107-117 (1998).

[5] J. Kleinberg, "Authoritative sources in a hyperlinked environment", Journal of the ACM, vol 46, #5, pp 604--632, 1999.

[6] E.A. Fox and J.A. Shaw, "Combination of Multiple Searches," Proceedings of the 2nd Text Retrieval Conference (TREC-2), NIST Special Publication 500-215, pp. 243-252, 1994.

[7] B.T. Bartell, G.W. Cottrell, and R.K. Belew, "Automatic Combination of multiple ranked retrieval systems," Proceedings of the 17th Annual ACM-SIGIR, pp. 173-181, 1994.

[8] N.J. Belkin, C. Cool, W.B. Croft and J.P. Callan, "The effect of multiple query representations on information retrieval performance," Proceedings of the 16th Annual ACM-SIGIR, pp. 339-346, 1993.

[9] N.J. Belkin, P. Kantor, E.A. Fox, and J.A. Shaw, "Combining evidence of multiple query representation for information retrieval," Information Processing & Management, Vol. 31, No. 3, pp. 431-448, 1995.

[10] J.H. Lee, "Combining Multiple Evidence from Different Properties of Weighting Schemes," Proceedings of the 18th Annual ACM-SIGIR, pp. 180-188, 1995.

[11] J. Aslam and M. Montague, et al., "Models for Metasearch", Proceedings of the 24th Annual ACM-SIGIR, 2001.

[12] M. Montague, et al., "Relevance Score Normalization for Metasearch", Proceedings of ACM-CIKM, 2001.

[13] M. Montague, et al., "Condorcet Fusion for Improved Retrieval", Proceedings of ACM-CIKM, November 2002.

[14] A. Chowdhury, et al., "Improved query precision using a unified fusion model", Proceedings of the 9th Text Retrieval Conference (TREC-9), 2000.

[15] J. Xu, B. Croft, "Corpus-based stemming using co-occurrence of word variants". ACM TOIS, January, 1998.

[16] Porter, "An algorithm for suffix stripping". Program, 14(3):130—137, 1980.

[17] Aljlayl, M. and Frieder, O. "On Arabic Search: Improving the Retrieval Effectiveness via a Light Stemming Approach", ACM Eleventh Conference on Information and Knowledge Management, Mclean, VA, November, 2002.

[18] Aljlayl, M., Beitzel, S., Jensen E., Chowdhury, A., Holmes, D., Lee, M., Grossman, D., and Frieder, O."IIT at TREC-10", Proceedings of the Tenth Annual Text Retrieval Conference (TREC-10), NIST, November, 2001.

[19] Al-Shalabi, R, and Evens, M. 1998. A Computational Morphology System for Arabic. Workshop on Computational Approaches to Semitic Languages, COLING -ACL.

[20] Khoja, S. 1999. "Stemming Arabic Text". Lancaster, U.K., Computing Department, Lancaster University.

[21] TREC 2002 Arabic/English Cross-Language IR Track Resources.

[22] Xu, J., Fraser, A., Weischedel, R. "TREC 2001 Cross-lingual Retrieval at BBN". Proceedings of the Tenth Annual Text Retrieval Conference (TREC-10), NIST, November, 2001.