

Probabilistic Structured Query Methods

Kareem Darwish

Electrical and Computer Engineering Department
and UMIACS

University of Maryland, College Park, MD 20742
{kareem,oard}@glue.umd.edu

Douglas W. Oard

College of Information Studies and UMIACS
University of Maryland, College Park, MD 20742
{kareem,oard}@glue.umd.edu

ABSTRACT

Structured methods for query term replacement rely on separate estimates of term frequency and document frequency to compute a weight for each query term. This paper reviews prior work on structured query techniques and introduces three new variants that leverage estimates of replacement probabilities. Statistically significant improvements in retrieval effectiveness are demonstrated for cross-language retrieval and for retrieval based on optical character recognition when replacement probabilities are used to estimate both term frequency and document frequency.

Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval – *Query formulation, Retrieval models, and Search process.*

General Terms

Algorithms and Theory.

Keywords

CLIR, OCR, Arabic, Term Replacement, and Structured Queries.

1 INTRODUCTION

There are many situations in which it is desirable to match a query term with different terms in a document. Well known examples include stemming (where any word that shares the same stem should be matched), thesaurus expansion (where terms with similar meanings should be matched), and cross-language retrieval (where terms with similar meanings in different languages should be matched). When the mappings among matching terms are known in advance, the usual approach is to conflate the alternatives during indexing. That is the typical way in which stemming is implemented, for example. Query-time implementations are necessary when appropriate matching decisions depend on the nature of the query, as might be the case with systems that provide the searcher with interactive control over thesaurus expansion. In this paper, presently known techniques for query-time

replacement are reviewed, new techniques that leverage estimates of replacement probabilities are introduced, and experiment results that demonstrate improved retrieval effectiveness in two applications (Cross-Language Information Retrieval (CLIR) and retrieval of scanned documents based on Optical Character Recognition (OCR)) are presented.

CLIR has received more attention than any other query-time replacement problem in recent years, and several effective techniques are now known. Query translation research has developed along two broad directions, typically referred to as “dictionary-based” and “corpus-based” techniques. Broadly speaking, corpus-based techniques seek to optimize retrieval effectiveness through reliance on observed translation probabilities in aligned corpora, while dictionary-based techniques are optimized for the case where reliable estimates of translation probability are not available.

A key idea in the so-called vector-space approach to information retrieval is reliance on two statistics: (1) term frequency (TF), the number of occurrences of a term in a document, and (2) document frequency (DF), the number of documents in which a term appears. TF is a measure of aboutness, which has beneficial effects on both precision and recall. DF is a measure of specificity, and its principal effect is on precision. In general, high TF and low DF are preferred, with the optimal combination of those factors typically being determined through experimentation (c.f., [15]).

Pirkola appears to have been the first to try separately estimating TF and DF for query terms in a CLIR application [13], using the InQuery synonym operator to implement what he called “structured queries.” InQuery’s synonym operator was originally designed to support monolingual thesaurus expansion, so it estimates TF and DF as follows [11]:

$$TF_j(Q_i) = \sum_{\{k|D_k \in T(Q_i)\}} TF_j(D_k) \quad (1)$$

$$DF(Q_i) = |\bigcup_{\{k|D_k \in T(Q_i)\}} \{d | D_k \in d\}| \quad (2)$$

where Q_i is a query term, D_k is a document term, $TF_j(Q_i)$ is the term frequency of Q_i in document j , $DF(Q_i)$ is the number of documents that contain Q_i , d is a document, and $T_f(Q_i)$ is the set of known replacements (in this case, translations) for the term D_k . Essentially, these equations treat any occurrence of a replacement as an occurrence of the query term. This represents a very cautious strategy in which a high DF for any replacement will result in a high “joint DF ” (and thus a low weight) for that query term. Retrieval results are then dominated by query terms that have no “unsafe” (very common) replacements. For example, the Arabic query term “علي” can either mean “on” or the proper name “Ali.” If “Ali”

appears in few documents but “on” appears in many, equation (2) will treat “علي” as if it were at least as common as “on.” When there is not a large disparity in DF , equation (1) has a kind of query expansion effect. For example, the Arabic word “خبز” can be translated as “bread” or “bake,” and equation (1) would (with proper stemming) reward an occurrence of “baking bread.”

Corpus-based approaches to CLIR have generally developed within a framework based on language modeling rather than vector space models, at least in part because modern statistical translation frameworks offer a natural way of integrating translation and language models [19]. In general, language modeling approaches to retrieval rely on collection frequency (CF) in place of DF :¹

$$CF(Q_i) = \sum_{k \in C} TF_k(Q_i) \quad (3)$$

where C represents the collection, and the other terms are as defined above. Whether DF is better than CF depends on how we model the searcher’s task—when the goal is to find entire documents, DF models the concept of “selectivity” with higher fidelity.

The next section introduces a set of replacement strategies that leverage observed replacement probabilities (from corpora) while retaining the vector space model’s concept of DF . The effectiveness and efficiency of this strategy relative to comparable baselines is then shown in subsequent sections for two applications: CLIR, and retrieval of scanned documents using OCR. The paper then concludes with some notes on limitations of the new techniques and opportunities for future work on this problem.

2 BEYOND PIRKOLA’S METHOD

Kwok was the first to introduce a variant to Pirkola’s method, aiming to reduce implementation complexity by replacing the union operator with a sum [8]:

$$DF(Q_i) = \sum_{\{k|D_k \in T(Q_i)\}} DF(D_k) \quad (4)$$

Another alternative, not previously explored, would be to use the maximum document frequency of any replacement (MDF):

$$DF(Q_i) = \text{MAX}_{\{k|D_k \in T(Q_i)\}} [DF(D_k)] \quad (5)$$

All three variants (Pirkola, Kwok, and MDF) lower bound the DF for a query term by the DF of its most common replacement, and the experiments reported in Sections 3 and 4 below show no statistically significant differences in retrieval effectiveness.

All three techniques treat every known replacement as equally likely. This risks a somewhat counterintuitive result: introduction of a translation dictionary with improved coverage of rare translations could actually harm retrieval effectiveness. To see this problem, consider the case of a query term in which 99.9% of its instances should be translated as some rare term (e.g., “superfluous”), but in 0.1% of the cases a translation that happens to be a common term (e.g., “the”) would be correct. In such cases, the common term leads to a high joint DF , effectively diminishing the degree to which

that query term contributes to the way in which documents are ranked. This exact situation actually arises often with dictionaries built from aligned corpora using statistical methods, since there is always some chance that any term might be observed to be used as a replacement for any other term. One way to address this problem would be to use a weighted variant of Kwok’s method:

$$DF(Q_i) = \sum_{\{k|D_k \in T(Q_i)\}} [DF_j(D_k) \times wt(D_k)] \quad (6)$$

In general, any monotone function of the replacement probability could be used for $wt(D_k)$. For the experiments reported below, the weight is simply set to the best available estimate of the replacement probability.

Improbable translations that are common terms can also cause problems with the TF computation in equation (1), since common terms are likely to have higher TF ’s as well. One way to limit this effect is to use a weighted sum in the TF computation:

$$TF_j(Q_i) = \sum_{\{k|D_k \in T(Q_i)\}} [TF_j(D_k) \times wt(D_k)] \quad (7)$$

Again, for the experiments reported below, the replacement probability estimate is used as the weight. Finally, either TF formula could be combined with any way of computing DF . In the experiments reported below, the following combinations were tried:

Method	TF Formula	DF Formula
Pirkola	(1)	(2)
Kwok	(1)	(4)
MDF	(1)	(5)
WDF	(1)	(6)
WTF	(7)	(4)
WTF/DF	(7)	(6)

Tables 1 and 2 use arbitrary figures to illustrate how the joint TF and DF would be calculated for each method.

Another way of leveraging information about replacement probabilities would be to simply ignore the least likely replacements. Such an approach potentially offers two potential benefits. First, it reveals the extent of the adverse effect of low-probability replacements on each technique. Second, it offers a principled way of tuning the degree of dictionary coverage to optimize the retrieval effectiveness. Two teams (from the University of Massachusetts [9] and the University of Maryland [2]) tried variants of this approach for Text Retrieval Conference’s CLIR track in 2002. For the experiments reported below, a greedy method was used, with replacements retained in order of decreasing probability until a preset threshold on the cumulative probability was first exceeded. That approach guarantees that at least one replacement will be retained.

¹ Hiemstra’s work is a notable exception [6].

Table 1: An example in which a 100 document collection is searched using the replacements for the Arabic word كتب.

Replacement for كتب (ktb)	Book	Wrote	Office	Report	Staff
Replacement Probability	0.48	0.40	0.06	0.04	0.02
DF	12	20	8	20	70
Weighted DF	5.76	8.00	0.48	0.80	0.14
TF – Doc A	0	0	1	5	2
Weighted TF – Doc A	0	0	0.06	0.20	0.04
TF – Doc B	2	2	0	0	0
Weighted TF – Doc B	0.96	0.80	0	0	0

Table 2: The joint TF and DF of different structured query methods for the example in Table 1.

Method	Joint DF	Joint TF	
		Doc A	Doc B
Pirkola	> 70	8	4
Kwok	100	8	4
MDF	70	8	4
WTF	> 70	0.3	1.76
WDF	15.2	8	4
WTF/DF	15.2	0.3	1.76

In the figures shown below, mean uninterpolated average precision is reported for every threshold value between 0.1 and 1.0, in increments of 0.1. The experiments were run using a modified version of PSE, a locally developed (and freely available) vector space retrieval system that uses Okapi BM25 weights. There are two parameters, K_1 and b , that tweak the effect of term frequency and document length respectively in the Okapi formula. The values of K_1 and b were set to 2 and 0.75 respectively as recommended by Robertson and Sparck Jones [14]. Differences in the means were tested for statistical significance using a Wilcoxon signed-rank test and reported as significant for values of $p < 0.05$.

3 CLIR

The CLIR experiments reported in this section were performed using the TREC 2002 CLIR track collection, which contains 383,872 articles from the Agence France Press (AFP) Arabic newswire, 50 topic descriptions written in English, and associated relevance judgments [12]. Queries were formed automatically using all the words in the title field of the topic description, which was designed to be representative of the style of queries typically issued in Web search applications. The documents were stemmed using Al-Stem (a freely available standard resource from the TREC CLIR track), diacritics were removed, and normalization was performed to convert the letters ya (ي) and alef maqsoura (أ) to ya (ي) and all the variants of alef (أ) and hamza (ء), namely alef (أ), alef hamza (أ), alef maad (أ), hamza (ء), waw hamza (ؤ), and ya

hamza (أ), to alef (أ). The English queries were stemmed before translation using the Porter stemmer for compatibility with the translation resources described below.

3.1 Estimating Replacement Probabilities

Five translation resources of three types were combined for this application. Combining resources is useful, because (a) the coverage of the combined resources is typically better than any of the individual resources, and (b) combining resources can serve to reinforce good translations. The resources were:

1. Two bilingual term lists that were constructed using two Web-based machine translation systems (Tarjim and Al-Misbar [17][18]). In each case, sets of isolated unique English words found in a 200 MB collection of Los Angeles Times news stories [10] were submitted for translation from English into Arabic. Each system returned at most one translation for each submitted word. Together, the two term lists covered about 15% of the unique Arabic stems in the TREC collection (measured by using Al-Stem on both the term list and the collection).
2. The Salmone Arabic-to-English dictionary (from Tufts University), from which we extracted only the translations. No translation preference information is indicated in this dictionary. The coverage of the resulting term list, measured in the same way, was about 7% of the unique Arabic stems in the TREC collection.
3. Two translation probability tables, one for English-to-Arabic and one for Arabic-to-English. These tables were constructed from tables provided by BBN, which were in turn constructed from a large collection of aligned English and Arabic United Nations documents using the Giza++ implementation of IBM’s model 1 statistical machine translation design. The coverage of the Arabic-to-English table, measured in the same way, was 29% of the unique Arabic stems in the TREC collection.

These translation resources were combined in the following manner:

1. All resources that were originally provided as Arabic-to-English were inverted. For the translation probability table, the probabilities for each translation pair were retained and then the inverted tables were renormalized so that the values of the “probabilities” for each source-language term summed to one. This process likely introduced some error, since probabilities for rare events may not have been accurately estimated.
2. A uniform distribution was used to assign probabilities to the translations obtained from machine translation systems and the Salmone dictionary. Tarjim and Al-Misbar each returned at most one translation for an English word, although two English words might share a common translation. When n alternatives were known from a single source, each was assigned a probability of $1/n$.
3. A uniform distribution was then assumed over the translation resources containing each English term. The translation probabilities resulting from step 2 were therefore combined by summing the probabilities for a given Arabic translation across the sources in which it appeared and then dividing by the number of sources in which the English term had appeared. For example, if Tarjim, Al-Misbar and Salmone contained the English

term, with Tarjim containing some specific translation with probability 1.0, Al-Misbar lacking that translation (i.e., assigning it a probability of 0.0), and Salmone assigning it a probability of 0.5 (because two translations were known), then the final estimate of the translation probability would be $1/3 + 0/3 + 0.5/3 = 0.5$.

The resulting translation resource contained what appeared to be reasonable estimates of translation probabilities, and covered 36% of the unique Arabic stems in the TREC collection.

3.2 Results

Figure 1 shows the mean uninterpolated average precision for each of the six structured query methods across a range of threshold values; Table 3 shows the same results in tabular form. As a baseline, one-best query translation (using only the most likely translation) was also run. This widely reported baseline seems appropriate in this case because any cumulative probability threshold will result in use of at least the most probable translation for each query term. Kwok’s and Pirkola’s methods turned out to be essentially indistinguishable, with the MDF method performing nearly as well (statistically significantly worse only at threshold values of 0.2 and 0.3).

Table 3: CLIR: Mean average precision, title queries. Black (gray) cells represent statistically better (worse) results, compared to the one-best translation baseline.

	Cumulative Probability Threshold – CLIR									
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Baseline	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16
Pirkola	0.16	0.16	0.17	0.19	0.17	0.14	0.09	0.08	0.07	0.01
Kwok	0.16	0.16	0.17	0.18	0.17	0.14	0.10	0.08	0.07	0.00
MDF	0.16	0.16	0.17	0.18	0.16	0.13	0.08	0.08	0.07	0.02
WTF	0.16	0.17	0.18	0.19	0.18	0.16	0.12	0.11	0.11	0.03
WDF	0.16	0.16	0.17	0.17	0.16	0.13	0.09	0.11	0.10	0.07
WTF/DF	0.16	0.17	0.18	0.19	0.20	0.20	0.19	0.19	0.19	0.19

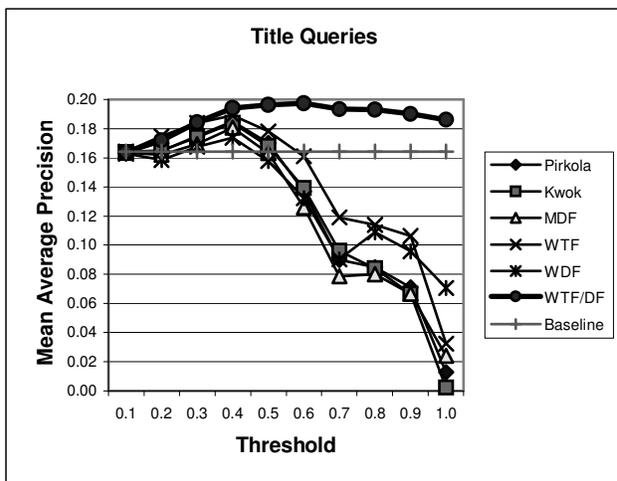


Figure 1: CLIR: Dependence of retrieval effectiveness on cumulative probability threshold, title queries.

The WTF/DF method produced results that were statistically significantly better than the one-best baseline for every

threshold value except 0.1, 0.2, and 1.0. Moreover, WTF/DF was the only weighted technique that did not exhibit a dramatic decrease in effectiveness as the threshold increased. The best WTF/DF result (at a threshold of 0.6) is statistically indistinguishable from the best results with Pirkola, Kwok, or WTF (in each case, at a threshold of 0.4), but the reduced dependence on accurate tuning of the threshold makes WTF/DF clearly the preferred method.

4 OCR-BASED RETRIEVAL

As for OCR-based retrieval, previous approaches to the problem have focused primarily on correcting OCR errors [7][16] or on fuzzy matching techniques that are less sensitive to OCR errors than exact string matching [1][5]. This section demonstrates the generality of the query-time replacement techniques introduced above, combining *TF* and *DF* evidence for a novel technique in which query terms are replaced with document terms that are likely to have resulted from OCR errors.

The experiments were conducted with the *Zad* collection, which was developed at the University of Maryland [3]. The collection consists of 2,730 documents extracted from *Zad Al-Me'ad*, a printed book for which an accurately character coded electronic version (the “clean text”) is also available. Three sets of OCR results for those documents are also available: print resolution (300x300 dots per inch (dpi), as originally scanned), and down-sampled versions at resolutions corresponding to the fine fax resolution (200x200 dpi) and the standard fax resolution (200x100 dpi). Note that down-sampling models a process different from the faxing process. The *Zad* collection includes 25 written topic descriptions and relevance judgments for every topic-document pair. Character normalization was performed as described above, and character 3-grams (3g) or character 4-grams (4g) were indexed. Darwish and Oard have found those to be among the most effective term types for OCR-based retrieval of Arabic [3].

4.1 Estimating Replacement Probabilities

Term replacement probabilities were estimated using a position-sensitive unigram character distortion model trained on 5,000 words of automatically aligned clean and OCR-degraded texts from the *Zad* collections. This alignment process was used to simulate manual error correction for a small sample of the collection being searched.² Since the appearance of Arabic characters varies by position, the standard four character positions (beginning, middle, end, isolated) were modeled.

Formally, given a clean word with characters $C_1..C_i..C_n$ and the resulting word after OCR degradation $D_1..D_j..D_m$, where D_j resulted from C_i , ϵ is the null character, L is the position of the letter in the word (beginning, middle, end, or isolated), and $\#$ is the word boundary, the three edit operations for the models would be:

$$P_{\text{substitution}}(C_i \rightarrow D_j | L) = \frac{\text{count}(C_i \rightarrow D_j | L)}{\text{count}(C_i | L)}$$

² Smaller and larger training sets were tried, but no improvement resulted from using more than 5,000 words.

$$P_{\text{deletion}}(C_i \rightarrow \epsilon | L) = \frac{\text{count}(C_i \rightarrow \epsilon | L)}{\text{count}(C_i | L)}$$

$$P_{\text{insertion}}(\epsilon \rightarrow D_j | L) = \frac{\text{count}(\epsilon \rightarrow D_j | L)}{\text{count}(C_i | L)}$$

If the count in the numerator was zero, the computation would be repeated without conditioning on position. If the count remained zero, a value of zero was recorded.

A separate model was trained for each resolution. Two factors made automatic alignment of the OCR output to the clean text challenging. First, the printed and clean text versions in the Zad collection were obtained from different sources that exhibited minor differences (mostly substitution or deletion of particles such as *in*, *from*, *or*, and *then*). Second, some areas in the scanned images of the printed page exhibited image distortions that resulted in relatively long runs of OCR errors. The alignment was performed using SCLITE from the National Institute of Standards and Technology (NIST). SCLITE employs a dynamic programming string alignment algorithm, which attempts to minimize the Levenshtein edit distance between two strings. Conceptually, the algorithm uses identical matches to anchor alignment, and then uses word position with respect to those anchors to estimate an optimal alignment on the remainder of the words.

SCLITE was originally developed for speech recognition applications, but in OCR applications additional character-level evidence is available. SCLITE alignments were therefore accepted only if the number of character edit operations were less than or equal to 50% of the length of the shorter of the two matched words. To align the words that were not aligned by SCLITE the following algorithm was used:

- Using the existing alignments as anchors, given an unaligned word at offset l from the preceding anchor in a clean document, sequentially compare it to the words, in the corresponding degraded document between the corresponding pair of anchors with offset l' from the preceding anchor where $||l' - l| \leq 5$.
- When comparing two words, if the difference between their respective word lengths was less than or equal to 2 characters and the number of edit operations between the two words (using Levenshtein edit distance) was less than a certain percentage q of the word length of the shorter one, then the newly aligned words were used as anchors. Initially, q was set to 60%.
- Steps 1 and 2 were iterated two more times using the new anchors with q equal to 40% and 20% in an attempt to find more alignments.

This alignment technique works well for print resolution, but it is a significant source of errors for highly degraded cases (e.g., standard fax resolution).

For each aligned word pair, corresponding characters were then aligned back tracing the Levenshtein edit distance algorithm to identify insertions, deletions, and substitutions.

The resulting model was then used to assign a probability to possible distortions of each query term as follows:

- For each character in a clean query term, generate all substitutions or deletions that have non-zero probability (i.e., were observed at least once in the training data). The unchanged character was also generated at this step as a “substitution.”
- For each possible insertion point, generate all possible single insertions. Possible insertion points are before the first character, between any pair of characters, and after the last character. A null insertion was generated at each point to cover the remainder of the probability mass.
- For each string that could result from the power set of all possible substitutions or deletions and all possible insertions, compute the probability of generating that string as the product of the associated insertion, substitution, and deletion probabilities.

A more efficient implementation would be desirable in an operational setting, but this approach sufficed for the experiments reported below.

4.2 Results

Figure 2 shows the mean uninterpolated average precision at print resolution for each structured query method across threshold values, and Table 4 shows the same data in tabular form. Figure 3 and Table 5 present the corresponding results for fine fax resolution. As a baseline, the same index terms (3g or 4g) were run with the clean (undistorted) queries, since any threshold on the cumulative probability results in a superset of that baseline case.

Table 4: Print: Mean average precision, title queries. Black (gray) cells represent statistically better (worse) results, compared to the clean query baseline.

		Cumulative Probability Threshold – Print									
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
3g	Baseline	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43
	Pirkola	0.44	0.44	0.44	0.45	0.44	0.45	0.47	0.42	0.35	0.09
	Kwok	0.44	0.44	0.44	0.45	0.44	0.44	0.47	0.42	0.29	0.08
	MDF	0.44	0.44	0.44	0.45	0.45	0.46	0.47	0.42	0.35	0.12
	WTF	0.44	0.44	0.44	0.44	0.44	0.44	0.47	0.43	0.39	0.18
	WDF	0.44	0.44	0.44	0.45	0.45	0.46	0.46	0.41	0.32	0.12
	WTF/DF	0.44	0.44	0.44	0.44	0.44	0.45	0.46	0.46	0.47	0.47
4g	Baseline	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
	Pirkola	0.46	0.46	0.46	0.46	0.48	0.48	0.49	0.47	0.41	0.20
	Kwok	0.46	0.46	0.46	0.46	0.48	0.48	0.49	0.47	0.40	0.21
	MDF	0.46	0.46	0.46	0.46	0.48	0.48	0.48	0.48	0.42	0.20
	WTF	0.46	0.46	0.46	0.46	0.47	0.47	0.49	0.48	0.42	0.26
	WDF	0.46	0.46	0.46	0.46	0.48	0.48	0.47	0.47	0.41	0.23
	WTF/DF	0.46	0.46	0.46	0.46	0.47	0.47	0.48	0.50	0.49	0.49

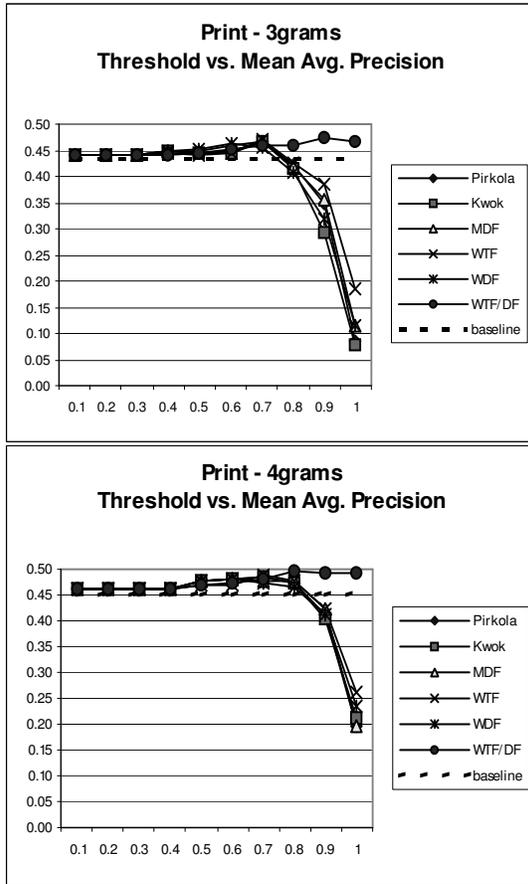


Figure 2: Print: Dependence of retrieval effectiveness on cumulative probability threshold, title queries.

No statistically significant differences were observed for any condition between the Pirkola, Kwok and MDF methods, which tends to confirm the observation made above that the Kwok's simpler implementation results is a suitable base case for the weighted extension to *DF*. For print resolution, every structured query technique achieved a statistically significant improvement over the baseline when used with the better of the two indexing terms (4g). Among these, WTF/DF achieved the greatest improvement (9.7% relative), and exhibited the greatest range of threshold values over which the improvement was statistically significant (0.6 to 1.0). Therefore, as with CLIR, WTF/DF is clearly the preferred technique in this application.

No statistically significant improvements over the baseline were observed for the fine fax resolution or the standard fax resolution (not shown). This may, however, reflect limitations in the automated alignment of the training data from which the replacement probabilities were used rather than limitations in the replacement techniques that were tried. The same general trends are observable in Figure 3 as in Figure 2, so the use of WTF/DF is certainly not counterindicated for the fine fax condition.

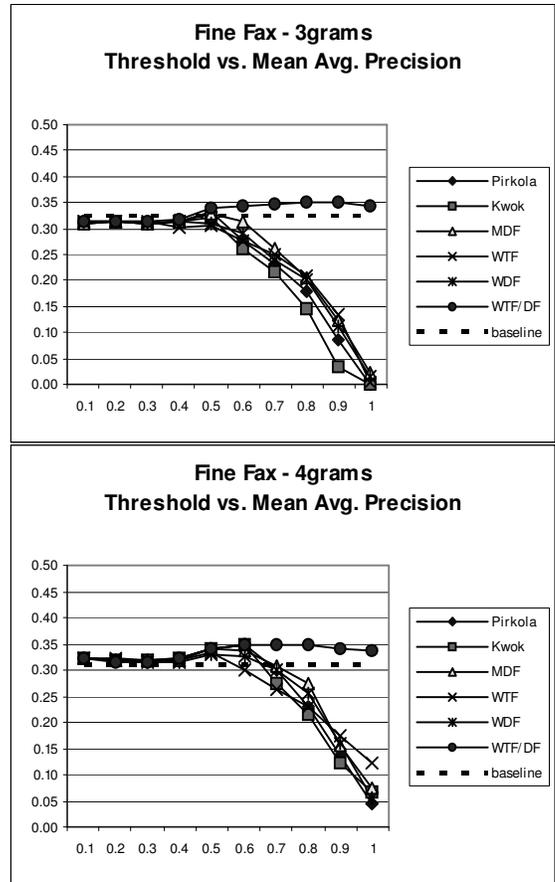


Figure 3: Fine fax: Dependence of retrieval effectiveness on cumulative probability threshold, title queries.

Table 5: Fine fax: Mean average precision, title queries. Black (gray) cells represent statistically better (worse) results, compared to the clean query baseline.

		Cumulative Probability Threshold - Fine Fax									
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
3g	Baseline	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32
	Pirkola	0.31	0.31	0.31	0.31	0.33	0.27	0.23	0.18	0.09	0.00
	Kwok	0.31	0.31	0.31	0.31	0.32	0.26	0.22	0.14	0.04	0.00
	MDF	0.31	0.31	0.31	0.31	0.33	0.31	0.26	0.20	0.12	0.02
	WTF	0.31	0.31	0.31	0.30	0.31	0.28	0.25	0.21	0.13	0.00
	WDF	0.31	0.31	0.31	0.31	0.31	0.29	0.24	0.20	0.11	0.02
	WTF/DF	0.31	0.31	0.31	0.32	0.34	0.34	0.35	0.35	0.35	0.34
4g	Baseline	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31
	Pirkola	0.32	0.32	0.32	0.32	0.34	0.35	0.30	0.23	0.14	0.04
	Kwok	0.32	0.32	0.32	0.32	0.34	0.35	0.28	0.21	0.12	0.07
	MDF	0.32	0.32	0.32	0.32	0.34	0.34	0.31	0.27	0.16	0.07
	WTF	0.32	0.32	0.32	0.32	0.33	0.30	0.26	0.23	0.17	0.12
	WDF	0.32	0.32	0.32	0.31	0.33	0.33	0.30	0.26	0.16	0.06
	WTF/DF	0.32	0.31	0.32	0.32	0.34	0.35	0.35	0.35	0.34	0.34

5 CONCLUSION AND FUTURE WORK

This paper introduced a family of methods for query term replacement that exploit both estimates of replacement probabilities and the vector space model's concept of "document frequency." Both Kwok's method and MDF were

found to achieve retrieval effectiveness values similar to that obtained with Pirkola's structured query method, so Kwok's method seems to be a good basis from which to build probabilistic structured query methods. Inclusion of rare translations in a CLIR application was shown to be problematic for all three methods, however. Use of only the most likely of those translations turned out to be an effective expedient, but only when an appropriate threshold on cumulative probability was selected. Of the three probabilistic structured query methods introduced in this paper, WTF/DF was the clear winner, yielding both the greatest retrieval effectiveness and the least sensitivity to the threshold tuning. Similar results were obtained for retrieval of OCR-degraded text based on generation of the most likely degraded variants of each query term.

A number of interesting directions for future work are suggested by these results:

1. Term weight tuning. The Okapi BM25 parameters that were used have proven to be effective in direct matching applications, but structured query methods have the effect of making each occurrence of a term somewhat less informative than it would be in direct match applications. In focusing on WTF/DF, the next logical step will be to explore the Okapi parameter space when using that method.
2. Other applications. The WTF/DF method can be used in any application where replacement probabilities can be reliably estimated. Examples of potential applications include thesaurus expansion, speech retrieval, corpus-based stemming, and perhaps gene sequence matching.
3. Structured document indexing. Query processing and document processing exhibit a strong duality, so it may be possible to leverage some of the techniques developed here at indexing time rather than query time for applications in which the replacement probabilities are fixed (e.g., translation based indexing [11]).

Access to reliable estimates of replacement probabilities from corpus statistics is becoming increasingly common, so the techniques described in this paper offer immediate utility. Perhaps even more importantly, the ideas presented here point the way towards promising future work on this important problem.

ACKNOWLEDGMENTS

The authors are grateful to David Doermann for his advice on OCR-based retrieval.

REFERENCES

- [1] Baeza-Yates, R. and G. Navarro, "A Faster Algorithm for Approximate String Matching." Proceedings of Combinatorial Pattern Matching (CPM'96), Springer-Verlag LNCS, v. 1075, pages 1-13, 1996.
- [2] Darwish, K. and D. Oard, "CLIR Experiments at Maryland for TREC 2002: Evidence Combination for Arabic-English Retrieval," TREC 2002.
- [3] Darwish, K. and D. Oard, "Term Selection for Searching Printed Arabic," SIGIR 2002, 261-268, 2002.
- [4] Gey, F. and D. Oard, "The TREC-2001 Cross-Language Information Retrieval Track: Searching Arabic Using English, French or Arabic Queries," TREC 2001, 16-25.
- [5] Harding, S., W. Croft, and C. Weir, "Probabilistic Retrieval of OCR Degraded Text Using N-Grams." European Conference on Digital Libraries, 1997
- [6] Hiemstra, D. Using language models for information retrieval Ph.D. Thesis University of Twente, Enschede, 2001.
- [7] Hong, T., "Degraded Text Recognition Using Visual and Linguistic Context." Ph.D. thesis, Computer Science Department, SUNY Buffalo, 1995.
- [8] Kwok, K. L., Personal communication.
- [9] Larkey, L., J. Allen, M. E. Connell, A. Bolivar, and C. Wade, "UMass at TREC 2002: Cross Language and Novelty Tracks," TREC 2002.
- [10] NIST, Text Research Collection Volume 5, April 1997.
- [11] Oard, D. W. and F. Ertunc "Translation-Based Indexing for Cross-Language Retrieval," ECIR 2002: 324-333, 2002.
- [12] Oard, D. W. and F. Gey, "The TREC-2002 Arabic/English CLIR Track," TREC 2002.
- [13] Pirkola, A. "The Effects of Query Structure and Dictionary setups in DictionaryBased Cross-language Information Retrieval," Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 55-63, 1998.
- [14] Robertson, S. E. and K. S. Jones. "Simple proven approaches to text retrieval." 1997, Cambridge University Computer Laboratory.
- [15] Robertson, S. E., S. Walker, M. Hancock-Beaulieu, A. Gull, and M. Lau, "Okapi at TREC-3," In the Fourth Text REtrieval Conference (TREC-3), 73--96, 1996.
- [16] Taghva, K., J. Borsack, and A. Condit, "An Expert System for Automatically Correcting OCR Output." Proceedings of the SPIE - Document Recognition, pages 270--278, 1994.
- [17] tarjim.ajeab.com, Sakhr Technologies, Cairo, Egypt www.sakhr.com
- [18] www.almisbar.com, ATA Software Technology Limited, North Brentford Middlesex, UK.
- [19] Xu, J., Weischedel, R., and Nguyen, C. Evaluating a Probabilistic Model for Cross-lingual Information Retrieval. In Proceedings of SIGIR, 2001, pages 105-110, 2001.