An Exploration of Proximity Measures in Information Retrieval

Tao Tao
Microsoft Corporation
Redmond, WA 98052
taotao@microsoft.com

ChengXiang Zhai
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801
czhai@uiuc.edu

ABSTRACT

In most existing retrieval models, documents are scored primarily based on various kinds of term statistics such as within-document frequencies, inverse document frequencies, and document lengths. Intuitively, the proximity of matched query terms in a document can also be exploited to promote scores of documents in which the matched query terms are close to each other. Such a proximity heuristic, however, has been largely under-explored in the literature; it is unclear how we can model proximity and incorporate a proximity measure into an existing retrieval model. In this paper, we systematically explore the query term proximity heuristic. Specifically, we propose and study the effectiveness of five different proximity measures, each modeling proximity from a different perspective. We then design two heuristic constraints and use them to guide us in incorporating the proposed proximity measures into an existing retrieval model. Experiments on five standard TREC test collections show that one of the proposed proximity measures is indeed highly correlated with document relevance, and by incorporating it into the KL-divergence language model and the Okapi BM25 model, we can significantly improve retrieval performance.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval models

General Terms

Algorithms

Keywords

Proximity, retrieval heuristics

1. INTRODUCTION

One of the most fundamental research questions in information retrieval is how to operationally define the notion of relevance so that we can score a document w.r.t. a query appropriately. A different definition generally leads to a different retrieval model. In the past a few decades, many different retrieval models have been proposed and tested, including vector space models [26, 25], classic probabilistic models [23, 29, 9], and statistical language models [21, 12, 16, 31, 18, 30, 32, 17, 6].

In most existing retrieval models, documents are scored primarily based on various kinds of term statistics such as within-document frequencies, inverse document frequencies, and document lengths [7], but the proximity of matched query terms in a document has not been exploited. Intuitively, given two documents that match the same number of query words, we would like to rank the document in which all query terms are close to each other above the one where they are apart from each other. Thus query term proximity is another potentially useful heuristic that can be incorporated into a retrieval model.

For example, consider the query "search engine" and the following two documents, both matching the two query terms once:

Example 1 *Document 1: "... search engine ..."*

Example 2 *Document 2: "... search .... engine ..."*

Intuitively, Document 1 should be ranked higher because its two query terms are adjacent to each other. In contrast, the two query terms in Document 2 are far apart, thus their combination does not necessarily imply the meaning of "search engine".

Interestingly, while intuitively quite appealing, such a proximity heuristic has so far been largely under-explored in the literature. Indeed, although several studies have looked into proximity [14, 15, 1, 10, 5, 22, 3, 2], the results are non-conclusive; it is still unclear how we should model proximity and how we can incorporate a proximity measure into an existing retrieval model. The proximity heuristic has also been *indirectly* captured in some retrieval models through using larger indexing units than words that are derived based on term proximity (e.g., [20]), but these models can only exploit proximity to a limited extent since they do not measure the proximity of terms. (See Section 2 for a detailed review of them.)

In this paper, we systematically study the effectiveness of the query term proximity heuristic through modeling term proximity *directly* and incorporating proximity measures into an existing retrieval model. We first study how to measure query term proximity independently of other relevance factors such as Term Frequency (TF) and Inverse Document Frequency (IDF); this way, we can isolate the proximity factor and see clearly its impact on modeling document relevance. Since it is unclear what is the best way to measure proximity, we systematically explore several different measures. They capture query term proximity from different perspectives. For example, one such measure (called "minimum coverage") is the minimum span of text in a document covering all the query terms at least once. Intuitively, the smaller the minimum coverage of a document is, the more likely it is relevant. Along

SIGIR'07, July 23–27, 2007, Amsterdam, The Netherlands.

similar lines, we propose four other proximity distance measures: span, minimum pairwise distance, average pairwise distance, and maximum pairwise distance.

To assess the potential of these proximity distance measures for improving a retrieval function, we compute their correlations with document relevance. The results show that minimum pairwise distance is more promising than others.

Next, we study how to exploit the proposed proximity distance measures to improve a retrieval model. Since the existing retrieval models have captured other retrieval heuristics very well and have proved to be effective over many different test collections, we study how to add proximity on top of them rather than develop a completely new retrieval with proximity heuristics from scratch. Specifically, we would add proximity to an existing retrieval function as a complementary scoring component to slightly adjust the relevance score of a document. This way, we can focus on evaluating the influence of the proximity feature on retrieval performance.

To incorporate a proximity distance measure into an existing retrieval function, we first define two heuristic constraints, in a similar way as in [7, 8], to capture the desirable properties of the new retrieval function that we would like to develop. These constraints suggest that the contribution of a proximity distance measure should follow a function of a convex shape. Our final function therefore uses a popular logarithm function to convert a proximity distance measure to a proximity feature value, which is then combined with two existing retrieval functions – the KL-divergence language model [16] and the Okapi BM25 model [23].

We evaluate our final models on five representative standard TREC data sets. The results show that the three pairwise distance measures are all effective for improving retrieval accuracy while the other two span-based measures are not effective, likely due to the problem of normalization. In particular, of all the proximity distance measures, we have found that the minimum pairwise distance measure is the best and when added on top of the KL-divergence retrieval model and the Okapi BM25 retrieval model, it can improve retrieval performance significantly.

The rest of the paper is organized as follow: We review the related previous work in Section 2, and report the experiment data and their statistics in Section 3. In Section 4, we propose and examine five proximity distance measures. In Section 5, we study how to incorporate the proximity heuristic into two existing retrieval models. We report experiment results in Section 6, and finally conclude our work in Section 7.

2. RELATED WORK

Keen's studies [14, 15] are among the early efforts to study the effectiveness of proximity in retrieval systems, in which, a "NEAR" operator was used to quantify the proximity of query terms. It has two major deficiencies: First, the experiments were conducted on very small data sets, thus the conclusions may not generalize well. Second, it was developed based on the Boolean retrieval model, which is generally regarded as less effective than the modern ranking-based full text retrieval models. The work[1] is one of the follow-up studies also dealing with Boolean queries. The studies [5] and [10] appear to be the first to evaluate proximity on TREC data sets. Both of them measure proximity by using a so-called "span" measure — the text segment containing all query term instances. The evaluation results are not conclusive. We will also evaluate this span feature in this paper. In addition, we also propose several other measures that are more effective than the span measure.

Some researchers studied proximity particularly based on the BM25 retrieval model [22, 3, 2]. They heuristically added proximity to the BM25 retrieval function, but their experiments are not conclusive and they have not reached a robust effective retrieval function through exploiting proximity. In our paper, we also combine the proximity measures with BM25. Compared with this previous work, our work is more systematic and results in an effective retrieval function in which proximity is effectively combined with other retrieval heuristics.

An indirect way to capture proximity is to use high-order n-grams as units to represent text. For example, in [27], bigram and trigram language models are shown to outperform simple unigram language models. However, query terms are not always adjacent to each other in documents. For example, if "search" and "engine" in the example given in Section 1 are separated by only a single word, a bigram language model would not be able to capture the proximity. We may attempt to capture such proximity by increasing the length of an n-gram. However, this would increase the size of the parameter space significantly, making parameter estimation inaccurate because we often have only an extremely small sample for parameter estimation, i.e., a document. A more general way to indirectly capture proximity through using appropriate "matching units" is Metzler and Croft's work on term dependency [20]. In this work, term structures with different levels of proximity can be defined in a general probabilistic model. Unfortunately, one has to pre-define the levels of proximity. Moreover, parameter estimation would be more difficult as we attempt to distinguish proximity at finer granularity levels. Thus in reality, it is impossible for these indirect methods of incorporating proximity to capture proximity in its full spectrum.

Our work is also related to passage retrieval [24, 4, 13, 19, 28, 11], where documents are often pre-segmented into small passages, which are then taken as units for retrieval. Since matching a passage implies imposing a proximity constraint on the matched query terms, passage retrieval can also capture proximity at a coarse granularity level, though it is clear that proximity can only be captured in a limited way with this approach.

3. EXPERIMENT DATA

We used several representative standard TREC data sets in our study [1]: AP (Associated Press news 1988-90), DOE (Department of Energy abstracts), FR (Federal Register), TREC8 (the ad hoc data used in TREC8), WEB2g (WT2g Web data). They represent different sizes and genre of text collections. Table 1 shows the statistics of these data. Throughout this paper, we will use these five data sets to do data analysis and evaluate proximity models.

	#document	queries	#total qrel
AP	164597	51-100	4805
DOE	226087	DOE queries	2047
FR	45820	51 - 100*	502
TREC8	528155	401-450	4728
WEB2g	247491	401-450	2279

Table 1: Experiment data sets. *We remove the queries without relevant documents in FR collection. Thus, there are only 21 queries left.

4. MEASURING PROXIMITY

Intuitively, we hope to reward a document where the matched query terms are close to each other. However, the issue is com-

[1] http://trec.nist.gov/

plicated because a query may have more than two terms and each term may occur multiple times in a document. In this section, we propose several proximity measures to capture this notion of closeness.

We start with assuming that we can segment a document into some units (e.g., terms or sentences). Based on a given segmentation method, we can then measure the length of any text segment by the number of units in the text segment and measure the distance between two term occurrences based on the number of units in between the two occurrences. In this paper, we assume that the unit for segmentation is a term. However, the proposed measures can be directly applied to other choices of the unit.

When a document matches two query terms each once, it would be natural to measure the proximity by the distance between the two matched query terms. However, in general, a document may match more than two query terms and each query term may occur multiple times in the document. A main challenge is thus to construct an overall proximity distance measure that can account for an arbitrary number of matched query terms.

We propose two kinds of approaches: (1) Span-based approaches: We measure the proximity based on the length of a text segment covering all the query terms. (2) Distance aggregation approaches: We measure the proximity by aggregating pair-wise distances between query terms. Relatively speaking, the first kind is more "global" because it must account for *all* query terms. In contrast, the second kind is more "local" because it may be more sensitive to the distance of an individual pair depending on how aggregation is done. Below we define five specific proximity distance measures in these two categories of approaches. We will use the following short document d as an example to explain our definitions.

$$d = t_1, t_2, t_1, t_3, t_5, t_4, t_2, t_3, t_4$$

4.1 Span-based proximity distance measures

Definition 1 (Span) *Span [10] is defined as the length of the shortest document segment that covers all query term occurrences in a document, including repeated occurrences.*

For example, in the short document d, the Span value is 7 for the query $\{t_1, t_2\}$.

Definition 2 (Min coverage (MinCover)) *MinCover is defined as the length of the shortest document segment that covers each query term at least once in a document.*

In the above example, if the query is $\{t_1, t_2\}$, its MinCover would be 2, but if the query is $\{t_1, t_2, t_4\}$, its MinCover would be 5 (the length of the segment from the second position to the sixth position).

4.2 Distance aggregation measures

Here we first define a pairwise distance between individual term occurrences, and then aggregate the pairwise distances to generate an overall proximity distance value. Specifically, we first pair up all the unique matched query words and measure their *closest* distances in documents. For example, when a query has three different words $\{t_1, t_2, t_3\}$ and a document matches all the three words, we can obtain three different pairs of query term combinations: $\{t_1,t_2\}$, $\{t_1, t_3\}$, and $\{t_2, t_3\}$. In the example document d, the closest distance for all these three pairs is 1 as they have all occurred next to each other somewhere. We use $\mathcal{D}is(t_1, t_2; D)$ to denote the closest distance between the occurrences of term t_1 and term t_2 in document D.

We now consider three different aggregation operators (i.e., Minimum, Average, and Maximum) and define the following three distance measures:

Definition 3 (Minimum pair distance (MinDist)) *The minimum pair distance is defined as the smallest distance value of all pairs of unique matched query terms. Formally,*
$MinDist = \min_{q_1,q_2 \in Q \cap D, q_1 \neq q_2}\{\mathcal{D}is(q_1, q_2; D)\}$.

For example, the MinDist of the example document d for query $Q = \{t_1, t_2, t_3\}$ is 1.

Definition 4 (Average pair distance (AveDist)) *The average pair distance is defined as the average distance value of all pairs of unique matched query terms. Formally,*
$AveDist = \frac{2}{n(n-1)} \sum_{q_1,q_2 \in Q \cap D, q_1 \neq q_2} \mathcal{D}is(q_1, q_2; D)$*, where n is the number of unique matched query terms in D, and in the sum, we count $\mathcal{D}is(q_1, q_2; D)$ and $\mathcal{D}is(q_2, q_1; D)$ only once.*

For example, the AveDist of the example document d for query $Q = \{t_1, t_4, t_5\}$ is $(1 + 2 + 3)/3 = 2$.

Definition 5 (Maximum pair distance (MaxDist)) *The maximum pair distance is defined as the largest distance value of all pairs of unique matched query terms. Formally,*
$MaxDist = \max_{q_1,q_2 \in Q \cap D, q_1 \neq q_2}\{\mathcal{D}is(q_1, q_2; D)\}$.

Note that all aggregation operators are defined over the pairwise distances on the *matched* query terms. The pairwise distance between two query terms is always based on their *closest* positions in a document. In the case when a document matches only one query term, MinDist, AveDist, and MaxDist are all defined as the length of the document.

All five measures can be calculated efficiently. We elaborate the calculation of MinCover briefly because it is not very straightforward. Assume that a document matches K unique query terms, and the total number of occurrences of these K query terms is N. We can record the positions of these N occurrences in order in the inverted index so that we can scan them one by one. While scanning, we maintain a list of length K, in which we store the *last position* of each seen query term. In other words, if a term t occurs twice, we would record the location of the first occurrence when the scanning hits the first one and update it when we hit the second one. In each step, we calculate the span solely based on the information in the list, and finally select the smallest span value we have ever obtained during the scanning process. Since K is often very small, the algorithm is close to linear in terms of N.

4.3 Evaluation of proximity distance measures

The five proximity distance measures defined above all capture proximity of matched query terms intuitively. We now look into the question whether they can potentially be exploited to improve a retrieval model. To answer this question, we examine the correlations between these measures and the relevance status of documents.

We use the KL-divergence retrieval method [16] to retrieve top 1000 documents for each query, calculate different proximity distance measures for each document, and then take the average of these values for relevant and non-relevant documents respectively. Intuitively, we expect the proximity distance measures on relevant documents to have smaller values than those on non-relevant documents since the query terms are expected to be closer to each other in a relevant document than in a non-relevant document.

We first report the Span and MinCover values in Table 2. In this table, we separate non-relevance scores and relevant scores in

	Span		MinCover	
	non-rel.	rel	non-rel.	rel
AP88-89	354.48	453.92	84.65	93.17
FR88-89	1457.01	5672.11	140.98	439.48
TREC8	505.98	796.82	49.96	51.37
WEB2g	1028.76	3370.65	102.46	150.22
DOE	48.52	73.58	15.81	14.43

Table 2: Global measures on different data sets

	Span		MinCover	
	non-rel.	rel	non-rel.	rel
AP88-89	50.78	46.43	30.09	27.63
FR88-89	104.13	150.90	46.38	127.93
TREC8	56.25	57.43	21.92	20.13
WEB2g	108.38	153.48	58.11	56.88
DOE	108.38	153.48	7.288	5.857

Table 3: Normalized global measures on different data sets

two columns. For example, the first number 354.48 means that the average of "Span" values of all non-relevant documents is 354.48.

Disappointingly, all the results are negative: the proximity distance values of relevant documents are all much larger than those of non-relevant documents. This counter-intuition result indicates that we may have missed important factors in relating proximity to document relevance. We notice that not all query terms appear in every document, and also some terms appear more frequently in one document than in another. When a document has more query terms, those terms would tend to span widely. Thus, both global measures (i.e., Span and MinCover) favor documents with fewer query term occurrences. To correct this bias, we therefore introduce a normalization factor. Specifically, we propose to normalize Span by dividing it by the total number of occurrences of query terms in the span segment, and normalize MinCover by dividing it by the number of unique query terms. We report the results from the normalized measures in Table 3.

We can make some interesting observations in Table 3. While Span still shows negative results, MinCover is now indeed slightly smaller on relevant documents than on non-relevant documents in most cases, suggesting the existence of weak signals. Even for Span, we also observe that the normalized version appears to be less negative than the non-normalized version.

The results about the three "local" measures are shown in Table 4.

We can now observe some interesting positive correlations in Table 4: While MaxDist results are still negative, both AveDist and MinDist are indeed positive. In particular, the MinDist measure has consistently smaller values for relevant documents than non-relevant ones.

The observations in this section suggest three things: First, normalization is an important factor for "global" measures. Second, a "local" measure can be expected to perform better than a "global" measures. In particular, MinDist is likely to be the best measure among all the five measures. As will be shown later in Section 6, these predictions are indeed true.

5. PROXIMITY RETRIEVAL MODELS

In this section, we study incorporation of the proposed proximity distance measures into an existing retrieval model. Since the raw values of proximity distances are generally not comparable with the values of a retrieval function, it is non-trivial to find an effective way of combining them. As we will show in Section 6, simply adding a good proximity distance measure to a retrieval function does not necessarily lead to a better retrieval function.

Our idea is to first figure out a way to transform a proximity distance measure appropriately so that it would make "reasonable" contributions to retrieval scores. Specifically, given a proximity distance function $\delta(Q, D)$ defined on document D and query Q, we would like to compute a retrieval score "adjustment factor" (denoted as $\pi(Q, D)$) based on $\delta(Q, D)$, i.e., $\pi(Q, D) = f(\delta(Q, D))$, where f is some transformation function possibly with a parameter. In order to obtain some guidance on designing this transformation function, we follow the axiomatic retrieval framework proposed in [7, 8] and define the following two constraints to help us design the transformation function:

First, we would like $\pi(Q, D)$ to positively contribute to the retrieval score of a document. We thus define the following basic proximity heuristic which simply says that a smaller $\delta(Q, D)$ implies a larger $\pi(Q, D)$.

Constraint (proximity heuristic) *Let Q be a query and D be a document in a text collection. Let D' be a document generated by switching the positions of two terms in D. If $\delta(Q, D) > \delta(Q, D')$, then $\pi(Q, D) < \pi(Q, D')$.*

Second, we would like the contribution from a distance measure to drop quickly when the distance value is small and become nearly constant as the distance becomes larger. The rational of this heuristic is the following: small distances between terms often imply strong semantic associations, thus we should reward cases where terms are really close to each other; however, when distances are large, the terms are presumably only loosely associated, thus the score contribution should not be so sensitive to the difference in distances as when the distances are small. This heuristic is formally defined as follows:

Constraint (Convex curve) *Let Q be a query and D_1, D_2, and D_3 be three documents that only differ in their term orders, but would otherwise be identical. That is, they have the same bag of terms, but the order of terms is different in each document. If $\delta(Q, D_1) = \delta(Q, D_2) - 1$ and $\delta(Q, D_2) = \delta(Q, D_3) - 1$, then $\pi(Q, D_1) - \pi(Q, D_2) > \pi(Q, D_2) - \pi(Q, D_3)$.*

These two constraints together suggest a convex curve for π as shown in Figure 1. Its first derivative should be negative and its second derivative should be positive.

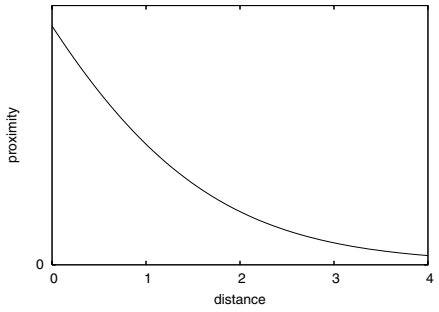


Figure 1: Ideal shape of the proximity transformation function

Such a curve can be obtained using the following function:

$$\pi(Q, D) = \log(\alpha + \exp(-\delta(Q, D))).$$

	MinDist		AveDist		MaxDist	
	non-rel.	rel	non-rel.	rel	non-rel.	rel
AP88-89	30.64	16.18	52.55	43.30	82.78	89.41
FR88-89	39.83	39.35	72.07	148.82	133.62	415.02
TREC8	31.77	19.15	39.33	32.25	48.91	49.92
WEB2g	67.91	61.20	82.73	96.65	100.42	146.44
DOE	11.68	7.66	13.38	10.57	15.31	13.97

Table 4: Local measures on different data sets

In this formula, we use $\exp(-\delta(Q,D))$ to map the distance values to the $[0,1]$ range, and then take a logarithm transformation to force the curve to satisfy the two constraints above. α is a parameter introduced here to allow for certain variations.

To test whether $\pi(Q,D)$ can indeed improve a retrieval function, we combine it with the following two representative state-of-the-art retrieval formulas (i.e., the KL-divergence language model [16] and the Okapi BM25 model [23]):

$$KL(Q,D) = \sum_{w\in q\cap d} c(w,q)\cdot ln(1+\frac{c(w,d)}{\mu\cdot p(w|C)}) + |q|\cdot ln\frac{\mu}{|d|+\mu} \quad (1)$$

$$BM25(Q,D) = \sum_{w\in q,\cap d}\left(ln\frac{N-df(w)+0.5}{df(w)+0.5} \times \frac{(k_1+1)\times c(w,d)}{k_1((1-b)+b\frac{|d|}{avdl})+c(w,d)} \times \frac{(k_3+1)\times c(w,q)}{k_3+c(w,q)}\right) \quad (2)$$

and obtain the following proximity-enhanced new retrieval functions:

$$R_1(Q,D) = KL(Q,D)+\pi(Q,D) \quad (3)$$
$$R_2(Q,D) = BM25(Q,D)+\pi(Q,D) \quad (4)$$

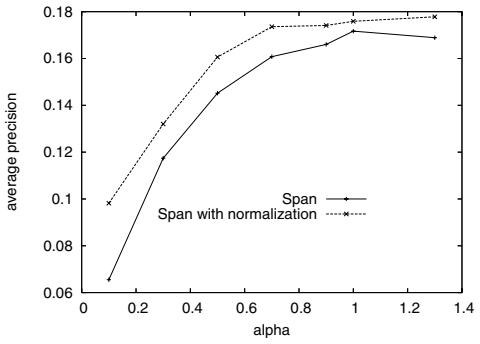
Although our extension of the two formulas is purely heuristic, we believe that exploring these modifications can shed light on how to eventually obtain a unified retrieval model with more principled incorporation of the proximity component. As will be shown in Section 6, both new formulas outperform the corresponding original formulas.

6. EXPERIMENT

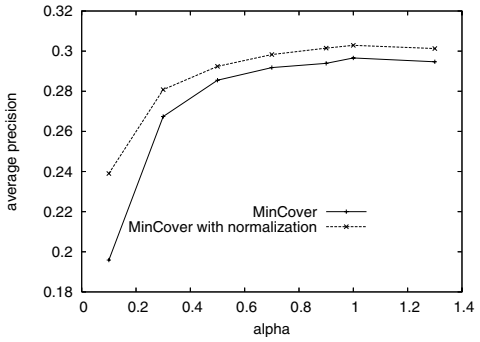
We test the proposed proximity retrieval models on the data sets listed in Section 3. In each experiment, we first use the baseline model (KL-divergence or Okapi BM25) to retrieve 2,000 documents for each query, and then use the proximity retrieval model to re-rank them. The top-ranked 1,000 documents for both the baseline run and the proximity run are compared in terms of their mean average precisions (MAP), which we use as our main evaluation metric.

6.1 Normalization of span-based measures

We first examine the effectiveness of different span-based proximity measures with and without normalization. For both Span and MinCover, we compare the non-normalized version with the normalized version at different α values. We show some representative results (Span on DOE and MinCover on WEB2g) in Figure 2.



(a) Span on DOE



(b) MinCover on WEB2g

Figure 2: Performance with and without normalization

As we expected, normalized Span and MinCover are more stable and more accurate than their corresponding non-normalized versions. This suggests that normalization is important for "global" measures in proximity modeling.

6.2 Best performance

We now turn to the question whether the proximity heuristic can improve retrieval performance. We report the best retrieval performance of all the five proximity measures for both R_1 and R_2 in Table 5. The table has two parts: the upper part is the KL-divergence model and its R_1 variations and the lower part is the Okapi model and its R_2 variations. In each part, the first row shows the retrieval performance of the original model. We use $\mu = 2000$ [31] in the KL-divergence language model. The Okapi BM25 has three main parameters. We set $k_1 = 1.2$ and $k_3 = 1,000$ as suggested in [23] and tune b to be optimal. The rest rows of the table are the best performance of all proximity models achieved by varying α in the range $[0,1]$. For the R_2 variations, we fix b to the optimal value tuned based on the original (baseline) Okapi model so that we only vary one parameter — α.

	method/data	AP	DOE	FR	TREC8	WEB2g
R_1	KL	0.2220	0.1803	0.2442	0.2509	0.3008
	Span	0.2203	0.1717	0.2436	0.2511	0.2992
	MinCover	0.2200	0.1685	0.2659	0.2455	0.2947
	MinDist	**0.2265***	**0.2018***	**0.2718**	**0.2573***	**0.3276***
	AveDist	0.2244	0.1922	0.2683	0.2538	0.3079
	MaxDist	0.2247	0.1913	0.2687	0.2536	0.2966
R_2	BM25	0.2302	0.1840	0.3089	0.2512	0.3094
	Span	0.2292	0.1808	0.3101	0.2468	0.3073
	MinCover	0.2260	0.1815	0.2881	0.2260	0.2966
	MinDist	**0.2368***	**0.2023***	**0.3135**	**0.2585***	**0.3395***
	AveDist	0.2314	0.1960	0.3115	0.2506	0.3148
	MaxDist	0.2323	0.1942	0.3115	0.2492	0.3144

Table 5: The best performance (MAP) of R_1 and R_2. The highlighted numbers are the best one among all values achieved by the comparable methods. Wilcoxon sign tests are done on the row of MinDist over the baseline method. * indicates the improvement is significant at 0.05 level.

method/data	AP	DOE	FR	TREC8	WEB2g
KL	0.368	0.260	0.152	0.452	0.446
R_1 + MinDist	0.374	0.280	0.138	0.460	0.468
BM25	0.376	0.288	0.147	0.446	0.486
R_2 + MinDist	0.418	0.300	0.166	0.456	0.502

Table 6: Pr@10 of the MinDist method over different data sets.

method/data	AP	DOE	FR	WEB2g	TREC8
KL	0.440	0.358	0.326	0.500	0.579
R_1 + MinDist	0.451	0.423	0.365	0.505	0.596
BM25	0.443	0.374	0.351	0.488	0.539
R_2 + MinDist	0.496	0.439	0.449	0.520	0.621

Table 7: Pr@0.1 of the MinDist method over different data sets.

method/data	AP	DOE	FR	WEB2g	TREC8
KL	3162	1018	242	2841	1860
R_1 + MinDist	3162	1018	250	2856	1880
BM25	3181	1044	270	2729	1717
R_2 + MinDist	3204	1043	312	2864	1933

Table 8: Retrieved relevant documents of the MinDist method over different data sets.

The results are consistent with our previous analysis of correlations: the two "global" measures (i.e., Span and MinCover) are not effective in general. Indeed, they hurt the performance in most cases. In contrast, the three "local" measures perform much better. They outperform the baselines in most experiments. We highlight the best values in each column in Table 5. It is very clear that the MinDist distance measure performs the best on every data set.

We do Wilcoxon sign tests on the improvement of MinDist over the baselines. Out of ten tests on five data collections, eight of them pass the test at the significant level of 0.05. In particular, the p-values of the two tests on WEB2g are smaller than 0.0001. We also observe that the improvement on FR88-89 is insignificant for both R_1 and R_2, even though their improvements look substantial. This may be because FR88-89 only has 21 queries (Table 1. When the number of sample points is small, a statistical test tends to support the null hypothesis, since there is insufficient evidence to support the alternative hypothesis.

We also observe from Table 5 that the improvement is not consistent across different data collections. The improvement appears to be most substantial on WEB2g and FR. For example, the MinDist with KL-divergence only improves the MAP value from 0.2220 to 0.2265 on AP, but it can improve MAP from 0.3008 to 0.3276 on WEB2g and from 0.2442 to 0.2718 on FR. We find that both FR88-89 and WEB2g have longer documents compared with the other data sets. Thus our results seem to suggest that proximity is more useful for collections with long documents. Indeed, because the query terms tend to spread in a wider range when documents are long, we may expect proximity measures to be more discriminative, thus more effective for improving retrieval accuracy.

Since it is not easy to interpret MAP values from a user's perspective, we further report the precision at 10 documents for MinDist (the best proximity measure) in Table 6, the precision at 0.1 recall level in Table 7, and the number of retrieved relevant documents at the cutoff 1000 (This score is indeed equivalent to the recall at the cutoff 1000) in Table 8.

We further compare our results with those from using the Markov random field (MRF) model proposed in [20], which can indirectly capture proximity to a certain extent. We set the three parameters (λ_T, λ_O, λ_U) in the MRF model to (0.8, 0.1, 0.1), as suggested in [20]. We compare the MRF results with our R_1 + MinDist results in Table 9. Interestingly, we find both methods perform very similarly on all five data sets, suggesting that a major reason why MRF performs well may be because it can capture proximity. It would be interesting to further analyze the connection between these two different ways of capturing proximity.

6.3 Parameter Sensitivity

The proposed proximity model has one parameter (α). We now look into the sensitivity of performance to this parameter in all the methods, especially the MinDist method. We plot the sensitivity curves of different methods on WEB2g in Figure 3. We see that "global" distances are all less stable and less accurate than "local" distances. This suggests that "local" proximity distances are generally more effective. Moreover, MinDist is clearly the best.

To better understand parameter sensitivity of MinDist, we further plot the performance sensitivity curves with this parameter on all

method/data	AP	DOE	FR	TREC8	WEB2g
R_1 + MinDist	0.2265	0.2018	0.2718	0.2573	0.3276
MRF	0.2270	0.2057	0.2695	0.2583	0.3284

Table 9: MAP Comparison of R_1+MinDist and MRF.

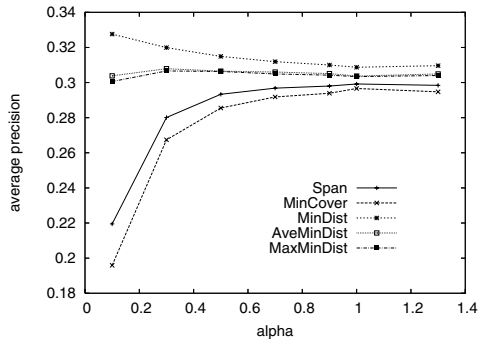


Figure 3: Sensitivity to parameter α of different methods

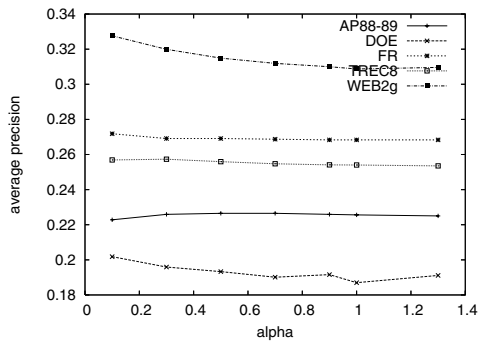


Figure 4: Parameter (α) sensitivity of MinDist over different data sets

five data sets in Figure 4. All curves appear to be stable, and setting $\alpha = 0.3$ appears to work well.

6.4 Effectiveness of the convex decay curve

We defined two constrains and used them to guide our design of proximity model in Section 5. Here, we want to demonstrate that this selection is non-trivial. For the purpose of comparison, we combine the proximity distance measure directly with the KL-divergence model. We name this new model as R_3:

$$R_3 = KL(Q, D) + \beta \times \text{Dir}\,(Q, D)$$

where $\beta \in (-\infty, \infty)$ is also a parameter to adjust the balance between the original KL-divergence score and the proximity score. Again, we use the best proximity distance measure, MinDist, for comparison. We do a *very large* range of exhaustive search for optimal β values.

The results are shown in Table 10, where we see that R_3 can hardly improve over the baseline, indicating the proximity part does not function at all. This shows that even if a proximity value is reasonable, simply adding it to a retrieval formula does not always work. In our study, we first try to develop some constraints, and then use them to guide our design of the function. As is shown in our experiment results, this axiomatic method [7, 8] can help us find an effective retrieval function.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we systematically explored the query term proximity heuristic. We proposed five different proximity distance measures, each modeling proximity from a different perspective. We evaluated their correlations with document relevance and found that the two span-based measures are generally not as well correlated with relevance as the three aggregated measures based on pairwise distances, and normalization is critical for span-based measures. In particular, the MinDist proximity distance measure is found to be highly correlated with document relevance.

We further define two heuristic constraints and use them to guide us to incorporate the proposed proximity distance measures into an existing retrieval model. Experiment results on five representative TREC test collections show that while span-based proximity measures cannot improve over the baseline, the pairwise distance-based measures can improve over the baseline most of the cases. The best performing measure is MinDist, which can be effectively combined with KL-divergence language model and the Okapi BM25 model to improve their retrieval performance significantly.

Our work can be extended in several directions: First, although we have found empirically that MinDist is the best among the five proximity measures proposed, further understanding of why it is the best is needed. This may also help us find even better proximity measures. Second, the transformation function for incorporating proximity into an existing model is only one of many possible choices of functions that can satisfy the two constraints. It is thus very interesting to further explore other possibly more effective transformation functions. Finally, it would be very interesting to develop a unified model to combine proximity heuristic and other retrieval heuristics such as TF-IDF weighting and document length normalization.

8. ACKNOWLEDGMENTS

This material is based in part upon work supported by the National Science Foundation under award number IIS-0347933. We thank Donald Metzler for helping us with testing the Markov random field model. We also thank the anonymous SIGIR 07 reviewers for their useful comments.

9. REFERENCES

[1] M. Beigbeder and A. Mercier. An information retrieval model using the fuzzy proximity degree of term occurences. In *Proceedings of the 2005 ACM Symposium on Applied Computing (SAC 05)*, pages 1018–1022, 2005.

[2] S. Buttcher, C. Clarke, and B. Lushman. Term proximity scoring for ad-hoc retrieval on very large text collections. In *SIGIR '03: Proceedings of the 26nd annual international ACM SIGIR conference on Research and development in information retrieval*, 2006.

	AP	DOE	FR	TREC8	WEB2g
KL	0.2220	0.1803	0.2442	0.2509	0.3008
R_1 + MinDist	0.2265	0.2018	0.2718	0.2573	0.3276
R_3 + MinDist	0.2221	0.1864	0.2440	0.2509	0.3005

Table 10: Comparison between R_1 and R_3 over MinDist

[3] S. Buttcher and C. L. A. Clarke. Efficiency vs. effectiveness in terabyte-scale information retrieval. In *Proceedings of TREC 2005*, 2005.

[4] J. P. Callan. Passage-Level Evidence in Document Retrieval. In W. B. Croft and C. van Rijsbergen, editors, *Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 302 – 310, Dublin, Ireland, July 1994. Spring-Verlag.

[5] C. L. A. Clarke, G. V. Cormack, and F. J. Burkowski. Shortest substring ranking. In *Proceedings of the Fourth Text REtrieval Conference (TREC-4)*, pages 295–304, 1995.

[6] W. B. Croft and J. Lafferty. *Language Modeling for Information Retrieval*. Kluwer, 2003.

[7] H. Fang, T. Tao, and C. Zhai. A formal study of information retrieval heuristics. In *Proceedings of the 27th annual international conference on Research and development in information retrieval*, pages 49–56. ACM Press, 2004.

[8] H. Fang and C. Zhai. An exploration of axiomatic approaches to information retrieval. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 480–487, 2005.

[9] N. Fuhr. Probabilistic models in information retrieval. *The Computer Journal*, 35(3):243–255, 1992.

[10] D. Hawking and P. Thistlewaite. Proximity operators - so near and yet so far. In *Proceedings of the Fourth Text REtrieval Conference (TREC-4)*, pages 131–143, 1995.

[11] M. A. Hearst. Improving full-text precision on short queries using simple constraints. In *Symposium on Document Analysis and Information Retrieval (SDAIR). Nevada, Las Vegas, 1996.*, 1996.

[12] D. Hiemstra and W. Kraaij. Twenty-one at trec-7: Ad-hoc and cross-language track. In *Proc. of Seventh Text REtrieval Conference (TREC-7)*, 1998.

[13] M. Kaszkiel and J. Zobel. Effective ranking with arbitrary passages. *Journal of the American Society of Information Science*, 52(4):344–364, 2001.

[14] E. M. Keen. The use of term position devices in ranked output experiments. *The Journal of Documentation*, 47(1):1–22, 1991.

[15] E. M. Keen. Some aspects of proximity searching in text retrieval systems. *Journal of Information Science*, (18):89–98, 1992.

[16] J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of SIGIR'2001*, pages 111–119, Sept 2001.

[17] J. Lafferty and C. Zhai. Probabilistic relevance models based on document and query generation. In W. B. Croft and J. Lafferty, editors, *Language Modeling and Information Retrieval*. Kluwer Academic Publishers, 2003.

[18] V. Lavrenko and B. Croft. Relevance-based language models. In *Proceedings of SIGIR'2001*, Sept 2001.

[19] X. Liu and W. B. Croft. Passage retrieval based on language models. In *Proceedings of CIKM 2002*, pages 375–382, 2002.

[20] D. Metzler and W. B. Croft. A markov random field model for term dependencies. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 472–479, 2005.

[21] J. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the ACM SIGIR*, pages 275–281, 1998.

[22] Y. Rasolofo and J. Savoy. Term proximity scoring for keyword-based retrieval systems. In *Proceedings of the 25th European Conference on IR Research (ECIR 2003)*, pages 207–218, 2003.

[23] S. E. Robertson, S. Walker, S. Jones, M. M.Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In D. K. Harman, editor, *The Third Text REtrieval Conference (TREC-3)*, 1995.

[24] G. Salton, J. Allan, and C. Buckley. Approaches to Passage Retrieval in Full Text Information Systems. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 49–58, 1993.

[25] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24:513–523, 1988.

[26] G. Salton, C. S. Yang, and C. T. Yu. A theory of term importance in automatic text analysis. *Journal of the American Society for Information Science*, 26(1):33–44, Jan-Feb 1975.

[27] F. Song and B. Croft. A general language model for information retrieval. In *Proceedings of the 1999 ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 279–280, 1999.

[28] S. Tellex, B. Katz, J. Lin, A. Fernandes, and G. Marton. Quantitative evaluation of passage retrieval algorithms for question answering. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 41–47, 2003.

[29] H. Turtle and W. B. Croft. Evaluation of an inference network-based retrieval model. *ACM Transactions on Information Systems*, 9(3):187–222, July 1991.

[30] C. Zhai and J. Lafferty. Model-based feedback in the KL-divergence retrieval model. In *Tenth International Conference on Information and Knowledge Management (CIKM 2001)*, pages 403–410, 2001.

[31] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of SIGIR'2001*, pages 334–342, Sept 2001.

[32] C. Zhai and J. Lafferty. Two-stage language models for information retrieval. In *Proceedings of SIGIR'2002*, pages 49–56, Aug 2002.