How Effective are Proximity Scores in Term Dependency Models?

Xiaolu Lu
RMIT University
Melbourne, Australia
xiaolu.lu@rmit.edu.au

Alistair Moffat
The University of Melbourne
Melbourne, Australia
ammoffat@unimelb.edu.au

J. Shane Culpepper
RMIT University
Melbourne, Australia
shane.culpepper@rmit.edu.au

ABSTRACT

The dominant retrieval models in information retrieval systems today are variants of TF×IDF, and typically use *bag-of-words* processing in order to balance recall and precision. However, the size of collections continues to increase, and the number of results produced by these models exceeds the number of documents that can be reasonably assessed. To address this need, researchers and commercial providers are now looking at more expensive computational models to improve the quality of the results returned. One such method is to incorporate term proximity into the ranking model. We explore the effectiveness gains achievable when term proximity is a factor used in ranking algorithms, and explore the relative effectiveness of several variants of the term dependency model. Our goal is to understand how these proximity-based models improve effectiveness.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Retrieval models, search process*; H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Performance evaluation*

Keywords

Experimentation, Measurement, Dependency Ranking Models

1. INTRODUCTION

Users of search services care about both result quality and retrieval time. But as collection size grows, balancing efficiency and effectiveness becomes increasingly difficult. Although the top ranked results returned by existing information retrieval models can satisfy a user's basic requirements, many weakly relevant or non-relevant documents are also returned. This occurs because many relevance models only consider a bag-of-words representation of documents [14], without taking into account the locations within the document at which those matching words occur. That is, bag-of-word retrieval models do not allow for the intuition that if the query terms occur near each other in a document, it may indicate that the document is more relevant. For example, consider the query "`scalable vector graphics`" (TREC Ad-hoc topic 849), and three different example documents:

1. . . . *word word word word word word word* scalable vector graphics *word word word word word word word word word*. . .
2. . . . *word word word word* scalable *word* vector *word word* graphics *word word word word word word word word word*. . .
3. . . . scalable *word word word word word* vector *word word word word word word word word word* graphics *word*. . .

Of the three, document 1 should arguably have the highest ranking score, since query terms appear together, and are more likely to represent the concept being sought. But document 2 also has merit as a potential match, since even though they are not adjacent as a phrase, the query terms are only separated by a few words, and can be regarded as reinforcing each other. In document 3, the terms are more distantly connected and it is the least likely to be a good match for the query. Taking adjacency and proximity into consideration when ranking has the potential to provide higher quality answers.

Increased effectiveness has a cost in terms of either index space, or query time, or a combination of both. Existing approaches to improving efficiency make use of term pair co-occurrence indexes, and employ early termination in order to balance space costs and query time [3, 19]. But unless the index can be very large, use of proximity-based metrics beyond co-occurrence usually requires on-the-fly computation of proximity scores, and possibly high retrieval times in even moderately sized collections. The best balance between efficiency and effectiveness for proximity-based models remains largely unexplored.

In this paper, we investigate the effectiveness gains achievable when proximity factors are included in ranking algorithms. We focus mainly on the *term dependency models* described by Metzler and Croft [12], as several recent studies have shown these approaches can provide high levels of effectiveness in a variety of settings. The effectiveness of the models was evaluated using the TREC8, GOV2 and ClueWeb09A datasets. Experimental results corroborate the hypothesis that proximity does improve result quality, across all three of these data collections.

The rest of the paper is organized as follows: Section 2 introduces related work on proximity ranking models and term dependency models used in this paper. Section 3 then describes the experiments and compares the models; finally, Section 4 summarizes our results and outlines future work.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
ADCS'14 , November 27–28 2014, Melbourne, Victoria, Australia
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3000-8/14/11 ...$15.00.
http://dx.doi.org/10.1145/2682862.2682876.

2. BACKGROUND

Related Work Proximity based ranking models have been widely studied in two ways: first, as variants of classic retrieval models such as Okapi BM25 [4, 8, 10, 14, 15, 19], and the KL-divergence (KLD) language model [9, 17, 18]; and second, as an inherent feature such as is the case with term dependency models [12].

Büttcher et al. [4] augment Okapi BM25 to calculate the distance between query terms and the adjacent terms. However, calculating proximity for all terms is computationally expensive. To address that issue, several recent studies have looked at the trade-offs possible through term pair co-occurrence indexing, or other similar means [3, 5, 7, 14]. In contrast to viewing query terms separately, Song et al. [15] group query terms into non-overlapped phrases referred to as a *span*. This provides context when ranking term occurrences. Terms in the same span are assigned the same contribution, and this contribution score replaces TF in Okapi BM25 for measuring term dependency. Further work by Svore et al. [16] finds the most important characteristics of a span that improve effectiveness. Instead of grouping terms into a span, He et al. [8] segment documents using a fixed size sliding window and then count the number of n-grams in the window. Since the counting method cannot indicate distance between terms in the same window, He et al. use *survival analysis* to differentiate whether n-grams appear loosely or tightly within a window. Their survival analysis-based experiments show that a counting window is sufficient for ad-hoc retrieval. However, determining a proper window size is nontrivial.

As a variant of the KLD model, the Positional Language Model (PLM) [9] assumes that a word can appear multiple times in the same document. A PLM is defined for each position in a document and used to predict whether a term occurred at position i. Different non-increasing distribution functions are used to generate *virtual documents*, and KLD is applied for scoring them, with position information translated in to frequency information. However, choosing a distribution function and assigning tunable parameters in the model were not explored. Recent work by Vuurens and de Vries [18] proposed a non-parametric KLD based model called Cumulative Proximity Expansion (CPE). Based on heuristic observations, Vuurens and de Vries assign different scores according to the distance between query terms.

Metzler and Croft [12] propose three variants of term dependency models – full independence (FI), sequential dependence (SD) and full dependence (FD). Both FD and SD consider ordered query phrases. Moreover, the SD model (SDM) incorporates unordered co-occurrences of query terms within a fixed distance, whereas the FD model (FDM) takes all query terms into consideration. Bendersky and Croft [1] have extended the previous work to *concept dependency models*, which do not treat query terms independently.

Retrieval Models We focus here on proximity features in the term dependency models proposed by Metzler and Croft [12]. Metzler and Croft [12] suggest that the SD model (SDM) is more suitable on small and homogeneous collections with longer queries, whereas the FD model (FDM) is better for larger and less homogeneous collections with shorter queries.

In the experiments described in the next section, the standard parameter weightings for these components are used, as originally described by Metzler [11]. Table 1 shows all of the models and feature weights explored in this work. There are three features used in FD and SD models: those based on terms (λ_t); those based on ordered phrases (λ_{op}); and those based on unordered windows (λ_{uw}).

Model	Combination of features
BOW	$1.00 \cdot \lambda_t$
BOW+OP	$0.85 \cdot \lambda_t + 0.15 \cdot \lambda_{op}$
BOW+UW	$0.85 \cdot \lambda_t + 0.15 \cdot \lambda_{uw}$
FDM	$0.80 \cdot \lambda_t + 0.10 \cdot \lambda_{op} + 0.10 \cdot \lambda_{uw}$
SDM	$0.85 \cdot \lambda_t + 0.10 \cdot \lambda_{op} + 0.05 \cdot \lambda_{uw}$

Table 1: Retrieval models and features used. The SDM model used an unordered window of fixed size 8, and was bigrams-only for ordered and unordered windows. The window size of the FDM model is $4 \cdot t$ where t is the number of terms being considered. That is, two-term combinations use an unordered window of size 4; three-term combinations an unordered window of size 8; and so on. Note also that in the SDM approach, λ_{op} and λ_{uw} apply to bigram and term-pair occurrences respectively.

Dataset	Documents	Topics
TREC 8	556,077	2004 Robust Track Topics 301-700
GOV2	25,205,179	2004-2006 Ad-hoc Track Topics 701-850
ClueWeb09A	150,955,773	2009, 2011, 2012 Ad-hoc Track Topics 1-50, 100-200

Table 2: Document collections and topics used in the evaluation.

3. EXPERIMENTS

Experimental Setup All experiments were performed using Indri[1], Krovetz stemming, and dependency models generated using Metzler's Indri configuration[2]. Three different TREC test collections and topic sets used are summarized in Table 2. Note that we use a pruned ClueWeb collection in which only documents with a spam score greater than 70 are included. This greatly improves the quality of results for all models tested. See Cormack et al. [6] for further information. For each query, the experiments return the top 1000 documents. Results are evaluated using Normalized Discounted Cumulative Gain (NDCG) evaluated to depth ten; Rank-Biased Precision (RBP) using $p = 0.95$ and evaluated to depth 1000 [13]; and Average Precision (AP), also evaluated to depth 1000. Results are then averaged over query sets, and query by query scores used for statistical testing.

Experimental Results The left half of Figure 1 shows the performance, relative to the BOW baseline, of BOW+OP, BOW+UW, FDM, and SDM. The results for the Robust track were similar, and are not shown. As expected, all of the enhanced BOW models improve the results for many queries, but also reduce the effectiveness of some queries. Perhaps the most noticeable outlier is the ordered phrase component (λ_{ox}). In both collections it causes a 25+% degradation for several queries. Since the SDM and FDM methods also incorporate a λ_{ox} component, albeit with a reduced weighting, they also experience a degradation in performance. In contrast, the proximity-alone scheme reduces performance in only a few queries, suggesting that it is more robust.

The breakdown by query length shows a similar trend. The BOW+UW variant has a few negative outliers but in general, de-

[1] `http://www.lemurproject.org/indri.php`
[2] `http://ciir.cs.umass.edu/~metzler/dm.pl`

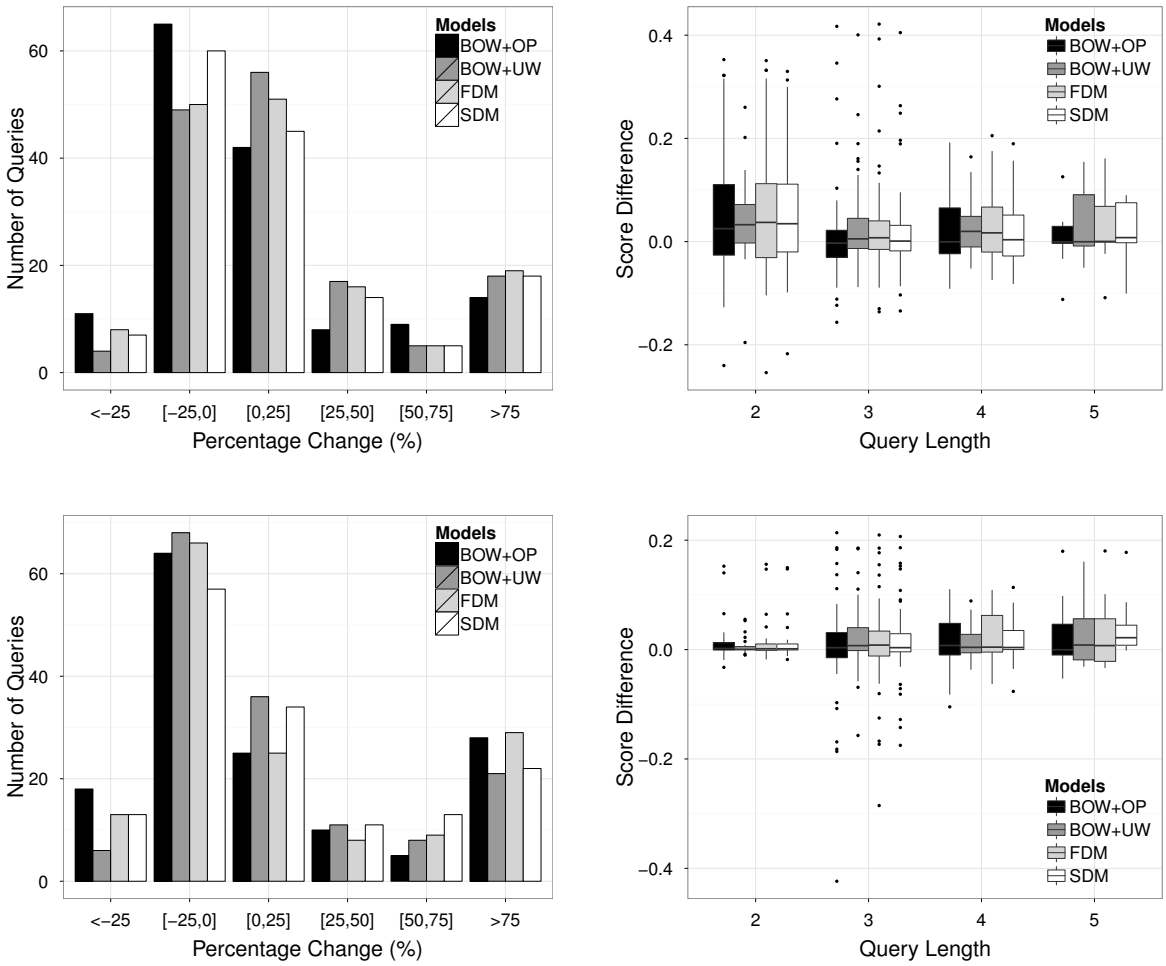

Figure 1: Comparing the effectiveness of retrieval models, using GOV2 (top pair of graphs) and ClueWeb09 (bottom pair of graphs). The left-hand column shows the distribution of query performance deltas, relative to the BOW baseline, categorized by delta size; the right-hand column shows score differential relative to BOW, categorized by query length.

pendence models using OP components are more likely to incur a noticeable degradation for certain queries.

Table 3 shows the three queries which are hurt most by the λ_{ox} component. These queries exemplify the pitfalls of presuming that all contiguous subsequences of the query are "good" phrases. Brute force partitioning of "`source of the nile`" results in fragments such as "`source of`" and "`source of the`", which are likely to have an unexpectedly high impact, while not being relevant to the query. On the other hand, it is a little surprising that treating "`iceland government`" as a phrase component hurts retrieval effectiveness. Query partitioning is a difficult problem even when done manually; and these results show that we still have a lot to learn about how queries should be handled automatically.

Table 4 lists measured effectiveness scores for all five retrieval models and three test collections. All of the dependency models increase overall effectiveness in general. The interesting trend is that BOW+UW is always better than BOW+OP, and has very similar performance to SDM and FDM. In fact, BOW+UW is marginally better than SDM for all metrics on the GOV2 collection. Average precision and RBP show that BOW+UW can achieve better effectiveness than BOW+OP on ClueWeb09. However, we also notice that the abnormal results on ClueWeb09 collection compared to the other datasets. This may be caused by using the default configuration of weighting parameters which were tuned by Metzler on much smaller datasets. A similar observation was also made by Vuurens and de Vries [18]. As suggested by Metzler and Croft [12], using an unlimited window size with SDM or FDM performs better on large collections, which could also be the reason. We will look more closely at tuning parameter sensitivity on collections in future work. Despite using an out-of-the-box configuration, most of the results are statistically significant with the exception of BOW+OP which failed to show a significant improvement over BOW runs for larger data collections.

Also worth noting in connection with Table 4 are the high RBP residuals for the ClueWeb experiments, evidence that large numbers of unjudged documents are being retrieved in the runs, and suggesting that the measured scores are not precise. With a residual that is of similar magnitude to the typical scores, the ClueWeb results need to be treated with caution. The Average Precision scores are also likely affected by this issue, but the extent of it cannot be quantified. Because of the shallow evaluation depth used, the NDCG@10 scores are not affected by unjudged documents.

Model	TREC8			GOV2			ClueWeb09A		
	NDCG	RBP	AP	NDCG	RBP	AP	NDCG	RBP	AP
BOW	0.4366	0.3080	0.2477	0.4396	0.4775	0.2920	0.2110	0.2553	0.1168
BOW+OP	$0.4512^{\ddagger}$	$0.3192^{\dagger}$	$0.2599^{\dagger}$	$0.4603^{\dagger}$	0.4940	0.3121	0.2266	0.2708	0.1248
BOW+UW	$0.4522^{\ddagger}$	$0.3201^{\ddagger}$	$0.2606^{\ddagger}$	$0.4892^{\ddagger}$	$0.5126^{\ddagger}$	$0.3221^{\ddagger}$	0.2224	$0.2729^{\ddagger}$	$0.1278^{\ddagger}$
FDM	$0.4512^{\ddagger}$	$0.3223^{\ddagger}$	$0.2635^{\ddagger}$	$0.4870^{\ddagger}$	$0.5119^{\ddagger}$	$0.3265^{\ddagger}$	$0.2308^{\dagger}$	$0.2778^{\ddagger}$	$0.1292^{\dagger}$
SDM	$0.4557^{\ddagger}$	$0.3218^{\ddagger}$	$0.2618^{\ddagger}$	$0.4781^{\ddagger}$	$0.5062^{\ddagger}$	$0.3193^{\ddagger}$	$0.2284^{\dagger}$	$0.2777^{\ddagger}$	$0.1299^{\ddagger}$

Table 4: Effectiveness for ranking models for TREC8, GOV2, and ClueWeb09A. The weighting parameter $\lambda_t = 0.85$ was used in BOW+OP and BOW+UW. The effectiveness metrics used were NDCG evaluated to depth 10 (column NDCG); RBP using $p = 0.95$, evaluated to the full run depth of 1,000 documents (column RBP); and Average Precision, also evaluated to full depth (column AP). For RBP the residual uncertainties (corresponding to unjudged documents appearing in the rankings) were all less than 0.04 for TREC8 and GOV2; but between 0.34 and 0.35 for ClueWeb09A. A superscript † represents $p < 0.05$, and a superscript ‡ represents $p < 0.01$, in both cases using a paired t-test compared to BOW as a baseline run.

Query	Dataset	Model	ΔAP
"source of the nile"	ClueWeb09A	BOW+OP	–0.4236
		FDM	–0.2852
		SDM	–0.0816
"iceland government"	GOV2	BOW+OP	–0.2402
		FDM	–0.2539
		SDM	–0.2173
"animals in alzheimer s research"	GOV2	BOW+OP	–0.1121
		FDM	–0.1087
		SDM	–0.1008

Table 3: Outlier queries identified in Figure 1 for the ClueWeb09A and GOV2 collections. These queries suffer from degraded effectiveness in models containing a λ_{ox} component.

4. CONCLUSION

We have investigated the relative effectiveness of proximity in dependency-based ranking models. Our preliminary study suggests that proximity is a more reliable component in dependency models than ordered phrase components. Care must be taken when incorporating ordered phrase components, as the results are noticeably degraded in some queries for reasons that are hard to anticipate.

Recent work on dependency models has focused on selective weighting of concepts in queries [2] or using local, global, and approximate statistics to improve effectiveness [10]. We do not investigate these enhancements in this work, but it is clear that selective weighting based on the query terms have the potential to improve effectiveness, across all aspects of dependency model processing.

Acknowledgment This work was supported by the Australian Research Council's *Discovery Projects* Scheme (DP140101587 and DP140103256). Shane Culpepper is the recipient of an Australian Research Council DECRA Research Fellowship (DE140100275).

References

[1] M. Bendersky and W. B. Croft. Modeling higher-order term dependencies in information retrieval using query hypergraphs. In *Proc. SIGIR*, pages 941–950, 2012.

[2] M. Bendersky, D. Metzler, and W. B. Croft. Parameterized concept weighting in verbose queries. In *Proc. SIGIR*, pages 605–614, 2011.

[3] A. Broschart and R. Schenkel. High-performance processing of text queries with tunable pruned term and term pair indexes. *ACM Trans. Information Systems*, 30(1):1–32, 2012.

[4] S. Büttcher, C. L. A. Clarke, and B. Lushman. Term proximity scoring for ad-hoc retrieval on very large text collections. In *Proc. SIGIR*, pages 621–622, 2006.

[5] I. Cetindil, J. Esmaelnezhad, T. Kim, and C. Li. Efficient instant-fuzzy search with proximity ranking. In *Proc. ICDE*, pages 328–339, 2014.

[6] G. V. Cormack, M. D. Smucker, and C. L. A. Clarke. Efficient and effective spam filtering and re-ranking for large web datasets. *Information Retrieval*, 14(5):441–465, 2011.

[7] A. Feuer, S. Savev, and J. A. Aslam. Implementing and evaluating phrasal query suggestions for proximity search. *Information Systems*, 34(8):711–723, 2009.

[8] B. He, J. X. Huang, and X. Zhou. Modeling term proximity for probabilistic information retrieval models. *J. of Information Sciences*, 181 (14):3017–3031, 2011.

[9] Y. Lv and C. Zhai. Positional language models for information retrieval. In *Proc. SIGIR*, pages 299–306, 2009.

[10] C. Macdonald and I. Ounis. Global statistics in proximity weighting models. In *Proc. SIGIR Web N-gram Wrkshp.*, pages 30–37, 2010.

[11] D. Metzler. *Beyond bags of words: Effectively modeling dependence and features in information retrieval.* PhD thesis, University of Massachusetts Amherst, 2007.

[12] D. Metzler and W. B. Croft. A Markov random field model for term dependencies. In *Proc. SIGIR*, pages 472–479. ACM, 2005.

[13] A. Moffat and J. Zobel. Rank-biased precision for measurement of retrieval effectiveness. *ACM Trans. Information Systems*, 27(1):2, 2008.

[14] R. Schenkel, A. Broschart, S. Hwang, M. Theobald, and G. Weikum. Efficient text proximity search. In *Proc. SPIRE*, pages 287–299, 2007.

[15] R. Song, M. J. Taylor, J.-R. Wen, H.-W. Hon, and Y. Yu. Viewing term proximity from a different perspective. In *Advances in Information Retrieval*, pages 346–357. Springer, 2008.

[16] K. M. Svore, P. H. Kanani, and N. Khan. How good is a span of terms? Exploiting proximity to improve web retrieval. In *Proc. SIGIR*, pages 154–161, 2010.

[17] T. Tao and C. Zhai. An exploration of proximity measures in information retrieval. In *Proc. SIGIR*, pages 295–302, 2007.

[18] J. B. P. Vuurens and A. P. de Vries. Distance matters! Cumulative proximity expansions for ranking documents. *Information Retrieval*, 17(4):380–406, 2014.

[19] H. Yan, S. Shi, F. Zhang, T. Suel, and J.-R. Wen. Efficient term proximity search with term-pair indexes. In *Proc. CIKM*, pages 1229–1238, 2010.