A Comparison of Retrieval Models using Term Dependencies

Samuel Huston and W. Bruce Croft
Center for Intelligent Information Retrieval
University of Massachusetts Amherst
140 Governors Dr, Amherst, Massachusetts, 01003
{sjh,croft}@cs.umass.edu

ABSTRACT

A number of retrieval models incorporating term dependencies have recently been introduced. Most of these modify existing "bag-of-words" retrieval models by including features based on the proximity of pairs of terms (or bi-terms). Although these term dependency models have been shown to be significantly more effective than the bag-of-words models, there have been no previous systematic comparisons between the different approaches that have been proposed. In this paper, we compare the effectiveness of recent bi-term dependency models over a range of TREC collections, for both short (title) and long (description) queries. To ensure the reproducibility of our study, all experiments are performed on widely available TREC collections, and all tuned retrieval model parameters are made public. These comparisons show that the weighted sequential dependence model is at least as effective as, and often significantly better than, any other model across this range of collections and queries. We observe that dependency features are much more valuable in improving the performance of longer queries than for shorter queries. We then examine the effectiveness of dependence models that incorporate proximity features involving more than two terms. The results show that these features can improve effectiveness, but not consistently, over the available data sets.

Categories and Subject Descriptors

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

General Terms

Experimentation, Performance

Keywords

Retrieval models, proximity, evaluation

1. INTRODUCTION

Recent research has demonstrated that retrieval models incorporating term dependencies (dependency models) [6, 16, 19, 21, 30] can consistently outperform benchmark "bag-of-words" models [1, 22, 26] over a variety of collections. We define a dependency model here as any model that exploits potential relationships between two or more words to improve a document ranking. Using a dependency model requires a query processing component and a scoring component. In query processing, groups of words that are potentially related are selected. This can be done in a variety of ways, but the most common is to select words that satisfy some proximity relationship, such as being next to each other in the query. An alternative would be to use linguistic analysis to identify words that have specific syntactic relationships. The scoring component of a dependency model modifies the scores of documents to take into account the presence of the selected query words in a specified relationship, such as satisfying a proximity or linguistic constraint. Most of the best-performing dependency models are based on proximity (or position-based) features and we focus on these models in this paper.

Although there have been a number of comparisons of dependency models to bag-of-words baselines, there has been surprisingly little comparison between these models. Given the importance of dependency models, it is critical to provide comparisons and baselines that can be used to establish the effectiveness of new models, instead of showing an improvement compared to relatively weak bag-of-words baselines. In this study, we compare the effectiveness of recent retrieval models that use term dependencies and, in addition, study the impact of different proximity features.

In the first part of the paper, we describe a systematic comparison of state-of-the-art dependency models that use features based on the proximity of pairs of terms (bi-terms). For this comparison, we use a range of TREC collections, including both short (title) and long (description) queries. By using these collections, query sets, and open source software, our results can be easily reproduced and used as baselines or benchmarks in future studies. The parameters for each model are extensively tuned to maximize performance, and 5-fold cross validation is used to avoid overfitting.[1]

Some dependency models have been proposed recently that use proximity features involving more than two terms [5, 30, 32, 33]. We define a many-term dependency as a set of three of more terms that are assumed to be dependent. The studies comparing many-term dependency features to bi-term features have been inconclusive, and providing more comprehensive evidence of the relative effectiveness of these proximity features is another goal of this paper. To do this, we compare the best bi-term dependency models to both existing many-term dependency models and models created by adding many-term features to the best bi-term dependency mod-

CIKM'14, November 3–7, 2014, Shanghai, China.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2598-1/14/11 ...$15.00.
http://dx.doi.org/10.1145/2661829.2661894.

[1]For space reasons, details of parameter settings and query folds are provided in a companion technical report.

els. Similar to the first part of the paper, the parameters for each model are extensively tuned and 5-fold cross validation is used.

To ensure fair comparisons between the models, we restrict the process of selection or generation of term dependencies from the input query so that it does not rely on external information, or a pseudo-relevance feedback algorithm [12]. These restrictions ensure that each tested model has access to identical information, and computing resources, thus allowing the direct attribution of retrieval effectiveness improvements or degradations to differences in model formulation and the features used. Further, these restrictions make these models widely applicable in many different information retrieval problems.

Results from our experiments show that the performance of all dependency models can be improved significantly through appropriate parameter tuning. This may not be a new or surprising conclusion, but the extent of the improvements possible is quite noticeable, and there are many published results where this tuning does not appear to have been done [2]. We also confirm the previous results showing that dependency models using bi-terms consistently improve effectiveness compared to bag-of-words models. The comparison between the bi-term dependency models shows that the variant of the weighted sequential dependence model [6] tested in this study exhibits consistently strong performance across all collections and query types. In regards to the comparison between short and long queries, we observe that dependency features have more potential to improve longer queries than shorter queries. We also show that many-term proximity features have the potential to improve retrieval effectiveness over the strongest bi-term dependency models. However, more research, and probably more training data, is required to fully exploit these features.

The major contributions of this study are:

- the first systematic comparison of dependency models incorporating bi-terms,
- a comparison of the effectiveness of dependency models across short and long queries
- a systematic comparison of many-term dependency models, including new variations of bi-term models
- tuned parameter settings for each tested retrieval model, for three standard information retrieval collections and query sets.

This paper is structured as follows. Section 2 discusses background and related work. Each of the tested dependency models are discussed in Section 3. The experimental setup is described in Section 4. Section 5 presents a detailed comparison of the retrieval models, and results and analysis from the investigation of many-term dependency features.

2. BACKGROUND AND RELATED WORK

2.1 Identification of dependent term groups

The simplest method of identifying groups of dependent terms in a query is to assume that all query terms depend on all other query terms. Several recently proposed dependency models make this assumption. BM25-TP [24] extracts all pairs of terms from the query. This results in a polynomial relationship between the number of query terms and the number of extracted dependencies. Similarly, the full dependence model (FDM) [19] uses each group in the the power set of query terms. This model produces an exponential number of query features, relative to the number of query terms. The large number of query features makes longer queries impractical for both these models. For this reason, these models are not included in this study.

Two retrieval models, BM25-Span [30] and the positional language model (PLM) also make the assumption that all query terms depend upon all other query terms. However, both models produce just a single group of dependent terms, the group of all query terms. As we will discuss in Section 3, we include both BM25-Span and PLM in our comparison of many-term dependency models.

To improve efficiency, a common alternative is to assume that only adjacent pairs of query terms are dependent. The n-gram language models approach presented by Song and Croft [29] is an early example of this method. Indeed, this approach has been used effectively by many recent dependency models [4, 6, 19, 21, 32, 33]. A key advantage of this dependency assumption is that even long queries remain computationally feasible after the inclusion of all dependency features.

While, in general, pairs of adjacent terms capture useful information in the query, the assumption of positional dependence also has the potential to introduce misleading pairs of terms. For example, consider extracting all sequential term pairs from the query: *desert fox news stories*. While the query intent may have been to find contemporary articles about the WWII field marshal, Erwin Rommel, the assumption of sequential dependence leads to the extraction of *"fox news"*. This pair of terms has the potential to mislead a retrieval model to focus on the news channel, rather than the desired information. Even so, retrieval models that use the sequential dependence assumption have been shown to improve average retrieval performance compared to bag-of-words retrieval models.

Several different linguistic techniques have been used in an attempt to improve dependency extraction. Srikanth and Srihari [31] use syntactic parsing to identify concepts in queries. Gao et al. [13] propose a method of extracting dependencies using a linkage grammar. Park et al. [20] use quasi-synchronous parsing to generate syntactic parse trees, from which dependencies are extracted. Maxwell and Croft [17] have recently shown retrieval performance can be improved through the use of dependency parsing techniques in the extraction of non-adjacent subsets of dependent terms. A common requirement of these methods is a natural language query.

Query segmentation has also been proposed as a method of extracting only the most informative dependencies from an input query [5, 7, 15, 25]. Typically, these methods use a set of features to determine where to segment (or separate) the query. Detected segments are then considered term dependencies. Bendersky and Croft [3] extend this work to classify a subset of detected query segments as key concepts.

None of these techniques have been shown to be consistently better than the simpler adjacent pairs method for identifying good candidate term dependencies. In addition, they commonly rely on external data sources such as linguistic models, Wikipedia articles, and query logs. For this reason, as stated in the introduction, we decided to simplify the comparison presented here by leaving an investigation of the non proximity-based methods to future work. This restriction eliminates the possibility that an observed performance difference can be attributed to unique, external data sources. Further, the investigations presented in this paper can be used to determine an appropriate benchmark for a future study investigating the utility of external data sources in dependency retrieval models.

2.2 Dependency models

Given that a group of terms has been identified as dependent, a dependency model must also specify how to match and evaluate each group of dependent terms for each scored document. Matching methods commonly focus on measuring how often the dependent group of terms occurs in the document, subject to a proximity constraint.

One of the simplest approaches is to match dependencies as n-grams or phrases. This method reports a match when all terms in a dependent group occur sequentially in a document. This type of feature has been used in several dependency models [6, 19, 29, 31]. From a scoring and weighting perspective, an occurrence of a dependent group is generally treated similarly to the occurrence of a single term.

Another common method of matching a dependent group to instances in a document is to count text windows, or constrained regions of the document, that contain all the dependent terms. Window-based features have been used in a number of retrieval models [19, 21, 24, 33]. Similar to the phrase-type features, window-based occurrences may be evaluated as terms [6, 19, 21]. Alternative approaches have also been proposed. For example, Rasolofo and Savoy [24] propose a method that scores windows of term pairs proportional to the inverse, squared width of the matched terms.

Tao and Zhai [33] propose several different aggregate distance functions over the set of all matched query terms in the document. In this study, documents are scored using KL divergence or BM25 combined with a retrieval score "adjustment factor". The adjustment factor transforms an aggregate measure of distance, over all matched term instances in the document into a "reasonable" score contribution. The aggregate functions tested include the minimum, maximum, average, span and min-cover distances. They observe that the minimum distance function in this model optimizes performance. However, the authors also observe that this best performing model is not statistically different from the sequential dependence model. For this reason, we omit the evaluation of these models from this work.

Another approach is to cluster all dependent term occurrences into text spans [30, 32]. Each span is evaluated according to the number of matching terms in the span and size of the span, where spans of a single term use the maximum span size. For these models, the definition of a span is implied by the specific algorithm proposed to cluster term occurrences into spans.

Positional language models [16] propose a method of modeling dependencies between all query terms. Dependencies are implicitly modeled through the evaluation of a specific document position. Each matching query term instance in the document propagates a partial count to the position to be evaluated, through a kernel function. If several queried terms occur near this position, then the probability of relevance for the position increases.

Some learning-to-rank retrieval models also use dependency features. For example, Svore et al. [32] investigate the effectiveness of several BM25-based dependency models, that are decomposed into features in a learning-to-rank retrieval framework. Their study also introduces new features that are detected in matched spans. Using a large set of training data, ($27,959$ queries), they are able to show large retrieval effectiveness improvements over the original formulations. A combination of a lack of training data, and the large number of parameters in these proposed learning-to-rank retrieval models, make evaluation of these models infeasible for this study.

3. RETRIEVAL MODELS

In this paper, we compare the relative effectiveness of bi-term dependency models, and the effectiveness of bi-term models compared to many-term models. We start by performing a systematic comparison of a variety of bi-term dependency models. This comparison allows us to determine the most effective benchmark bi-term dependency model. We then compare this benchmark model against existing many-term dependency models and variations of the benchmark model that include many-term dependencies. In this section, we present and discuss each of the dependency models that will be compared empirically in Section 5. Note that we do not claim to evaluate all existing dependency models, instead, we limited our evaluation to more "popular" models that have been used in multiple studies. We also, as previously noted, do not make use of external data sources or pseudo-relevance feedback.

Table 1: Feature functions used by the WSDM-Internal retrieval model.

Feature	Description
c^1	Constant value for terms (1.0)
$cf(t)$	Collection frequency of term t
$df(t)$	Document frequency of term t
c^2	Constant value for bi-terms (1.0)
$cf(\#1(t_1, t_2))$	Collection frequency of bi-term; t_1, t_2
$df(\#1(t_1, t_2))$	Document frequency of bi-term; t_1, t_2

3.1 Bi-term dependency models

Language Modeling: The language modeling framework for information retrieval [22] has been shown to be an effective bag-of-words retrieval model. This model allows for various assumptions in the estimation of the probability of relevance. Query likelihood (QL) [22] is commonly used as a strong bag-of-words baseline retrieval model. Several different methods have been proposed to model dependencies between terms in this framework.

Metzler and Croft [19] propose the Markov random field retrieval model for term dependencies. This framework explicitly models the relationships between query terms. The sequential dependence model (SDM) assumes that all pairs of sequential terms extracted from the query are dependent. It models each bi- term dependency using two types of proximity features; an ordered and an unordered window. The ordered windows matches n-grams or phrases in each evaluated document, the unordered window matches each pair of terms that occur in a window of 8 terms or less. SDM scores each term, ordered window and unordered window feature using a smoothed language modeling estimate. A weighted linear combination is then used to produce a final estimate of the probability of relevance for a document, given the input query.

Bendersky et al. [6] extend SDM to incorporate term and window specific weights (WSDM). Each term and window is assigned a weight using a linear combination of features, extracted for each term and window. The parameters for this model control the contribution of each feature to the weight of a specific term or window. In their initial study, a total of 18 features are used to estimate the optimal weight for each term and window. In later studies, the feature set is reduced to 13 features without exhibiting diminished performance [4]. Several of these features are computed over external data sources, including a query log, Wikipedia and the Google n-grams collection. In accordance with our restrictions, we limit this model to the set of features that are computed over the target collection only. We label this model variant WSDM-Internal (WSDM-Int). The subset of features used to estimate the weight of each term and bi-term feature are listed in Table 1.

Divergence from Randomness: Retrieval models based on the divergence from randomness (DFR) framework [1] have been used in a number of studies. Term dependencies have recently been introduced to this framework [21]. Similar to SDM, the proximity divergence from randomness model (pDFR) assumes that all adjacent pairs of terms are dependent. In their formulation, terms are scored using the PL2 scoring model, and bi-terms are scored using the BiL2 scoring model. The score for each document is the weighted sum of the term and bi-term components. The authors state that other DFR models can be used to score each component, which we intend to investigate in future work.

In this study, we test two pDFR models. First, we investigate the model proposed by Peng et al. [21] that uses PL2 to score unigrams, and BiL2 to score bigrams (pDFR-Bil2). We also investigate a variation that uses PL2 to score both unigrams and bigrams (pDFR-PL2).

BM25: BM25 [26] is an effective extension of the Binary Independence Model [27] (BIM). It is important to note that BIM makes a strong assumption of term independence [10]. Specifically, the theoretical underpinnings of BIM, and therefore BM25, preclude the inclusion of term dependency information into the estimation of the probability of relevance. Even so, several heuristic retrieval models that combine BM25 components with term proximity-based features have been proposed.

Rasolofo and Savoy [24] present an version of the BM25 model that includes term dependency features. Their model (BM25-TP) extends the BM25 model with a feature that is proportional to the inverse square of the distance between each pair of queried terms, up to a maximum distance of 5. Büttcher et al. [8] investigate the optimization of this function for efficient retrieval. Svore et al. [32] further investigates this model in comparison to other BM25-based dependency models. We note that, as proposed, BM25-TP uses all possible term pairs extracted from the query. The polynomial growth in the number of query features, as the size of the query grows, makes the model infeasible for longer queries. In this study, we use the variation of BM25-TP proposed by Svore et al. [32] that only assumes that all sequential pairs of query terms are dependent.

We also include a variant of this retrieval model (BM25-TP2), also proposed by Svore et al. [32]. This model is similar to BM25-TP, except that the score for each bi-term feature is no longer proportional to the inverse squared distance between each matched term pair. Instead, the feature is scored according to the number of matching instances using a BM25-like function.

3.2 Many-term dependency models

Only a small number of many-term dependency models have been proposed. This, in itself, is considered evidence that many-term dependencies may be less effective than bi-term dependencies. In this study, we investigate two recently proposed models; BM25-Span, and the positional language model (PLM). We also study variations of the SDM and WSDM-Int models that include many-term dependency features.

Song et al. [30] propose the BM25-Span model. This model assumes that all queried terms are dependent. They model this single set of dependent terms by grouping term instances in a given document into *spans*. The BM25-Span model scores the detected spans using a modified BM25 scoring function. The BM25-Span scoring function evaluates each span by interpolating between the span width and number of query terms in each span.

Further extensions to this model have also been proposed. For example, Svore et al. [32] extend this model with 14 new features extracted from each span. They observe significant improvements over the original BM25-Span model. However, many of these features require external data sources, such as training data for phrase, sentence detection algorithms, and various word lists. For this reason, we leave the investigation of this extended span- based retrieval model to future work.

The positional language model (PLM) was proposed by Lv and Zhai [16]. Similar to BM25-Span, this model assumes that all queried terms are dependent on each other. PLM operates by propagating each occurrence of each query term in the document to neighboring locations. The document is then scored at each location, or term position, in the document. Kernel functions are used to determine the exact propagated frequency of each term in the document to the specific position to be scored. Several methods are used for producing an aggregate score from the individual position scores.

Table 2: Descriptions of the potential functions used in various extensions to SDM. Unordered window widths are held constant at 4 times term count [19].

Feature	Description
Uni	Unigram feature
O2 / U2	Ordered / Unordered window of pairs of terms
O3 / U3	Ordered / Unordered window of sets of three terms
O4 / U4	Ordered / Unordered window of sets of four terms

Table 3: Feature functions used by the WSDM-internal-3 retrieval model, in addition to the parameters shown in Table 1.

Feature	Description
c^3	Constant value for 3 term dependencies (1.0)
$cf(\#1(t_1, t_2, t_3))$	Collection frequency of trigram; t_1, t_2, t_3
$df(\#1(t_1, t_2, t_3))$	Document frequency of trigram; t_1, t_2, t_3

In this study, we consider the two best performing variants proposed in the original study; the best-position strategy, using the gaussian kernel (PLM), and the multi-σ strategy, using two gaussian kernels (PLM-2). The best- position strategy scores each document using the score from the maximum scoring position in the document. The multi-σ interpolates between two best-position strategies, with different kernel parameters. As suggested by Lv and Zhai [16], the second kernel in our PLM-2 implementation is a whole document language model, $\sigma_2 = \infty$. We note that this particular kernel is equivalent to the query likelihood model (QL). Both models are implemented using Dirichlet smoothing.

We now describe many-term dependency models that are variants of the SDM and WSDM models. We start by extending SDM [19] to include many-term dependencies. As mentioned previously, SDM is a variant of the Markov random field model that uses three types of potential functions; terms, ordered windows and unordered windows.

We extend this model by adding new potential functions to the model that evaluate larger sets of dependent terms. Table 2 lists each of the potential functions in the extended model. Each function is computed in a similar manner to the functions in the original SDM. Note that the width of each unordered window feature is set at four times the number of terms, as in the original study [19]. Various many-term dependency models are constructed by selecting different subsets of features. For example, the `Uni+O23+U23` model includes the unigram function; two, bi-term functions (O2, and U2); and two, 3 term functions (O3, and U3). In these models `O` indicates an ordered window is used, and `U` indicates an unordered window is used. In each model, each function is associated with one weight parameter. All features are computed as smoothed language model estimates, using Dirichlet smoothing. Similar to SDM, for an input query, each potential function is populated with all sequential sets of terms extracted from the query.

We note that each of these features is also present in the full dependence model (FDM) [19]. However, in FDM all ordered windows are weighted with the parameter λ_O, and similarly, all unordered windows are weighted the parameter λ_U.

We also construct a variant of the WSDM-Int model that incorporates three-term dependencies (WSDM-Int-3). WSDM-Int-3 extends WSDM-Int by including three-term features, similar to the existing two-term features. The weight for each three-term dependency is determined using a linear combination of the features in Table 3. Term and bi-term features are weighted as in WSDM-Int.

This extension adds three new parameters to the WSDM-Int model, one for each three-term weighting feature.

4. EXPERIMENTAL SETUP

To test the effectiveness of dependency retrieval models, we use three TREC collections, Robust04, GOV2, and ClueWeb-09-Cat-B. The details of these collections are provided in Table 4. Based on results presented at TREC 2009 and TREC 2010, ClueWeb-09-Cat-B is filtered to the set of documents with spam scores in the 60^{th} percentile, using the Fusion spam scores distributed with ClueWeb-09 [11]. Retrieval effectiveness results are reported for both the title and the description of each TREC topic. Robust-04 topics are collected from TREC Robust Track 2004. The GOV2 topics are accumulated over TREC Terabyte Tracks 2004, 2005, and 2006. The Clueweb-09-Cat-B topics are accumulated from TREC Web Tracks 2009, 2010, and 2011. The INQUERY stopwords [9] are removed from all topics. Indexed terms are stemmed using the Porter 2 stemmer.[2] All retrieval models are implemented using the Galago Search Engine [3].

Given the limited number of queries for each collection, 5-fold cross- validation is used to minimize over-fitting without reducing the number of learning instances. Topics for each collection are randomly divided into 5 folds. The parameters for each model are tuned on 4-of-5 folds. The final fold in each case is used to evaluate the optimal parameters. This process is repeated 5 times, once for each fold. For each fold, parameters are tuned using a coordinate ascent algorithm [18], using 10 random restarts. Mean average precision (MAP) is the optimized metric for all retrieval models. Throughout this paper each displayed evaluation statistic is the average of the five fold-level evaluation values.

For each collection, each type of query, and each retrieval model, we report the mean average precision (MAP), normalized discounted cumulative gain at rank 20 (nDCG@20), precision at rank 20 (P@20), and expected reciprocal rank at 20 (ERR@20). For each collection, each metric is computed over the joint results, as combined from the 5 test folds. Statistical differences between models are computed using the Fisher randomization test as suggested by Smucker et al. [28], where $\alpha = 0.05$.

Due to space constraints in this paper, the full details of the query folds, learned parameters, fold-level evaluation metrics, and measured p-values for statistical tests are published in a separate technical report [14].

5. RESULTS

5.1 Tuning retrieval models

Most of the models investigated in this study were originally proposed and tested over a variety of different collections and queries, and any parameters suggested in each corresponding study are unlikely to be optimal for other collections. In this section, we present a case study on tuning the SDM parameters. Metzler and Croft [19] suggest that the SDM parameters, $(\lambda_T, \lambda_O, \lambda_U)$, should be set to $(0.85, 0.1, 0.05)$. The smoothing parameter μ is assumed to be 2,500, the default smoothing parameter in the canonical implementation. [4]

The 5-fold cross-validation method used in this study learns 5 settings for each parameter, one setting per test fold. The average learned parameter settings for each collection, and type of query is shown in Figure 1. This graph suggests that, with the exception of μ, the SDM parameters are relatively stable. The optimal

[2]http:/ /snowball.tartarus.org/algorithms/english/stemmer.html
[3]http://www.lemurproject.org/galago.php
[4]Indri Search Engine, http://www.lemurproject.org/indri.php

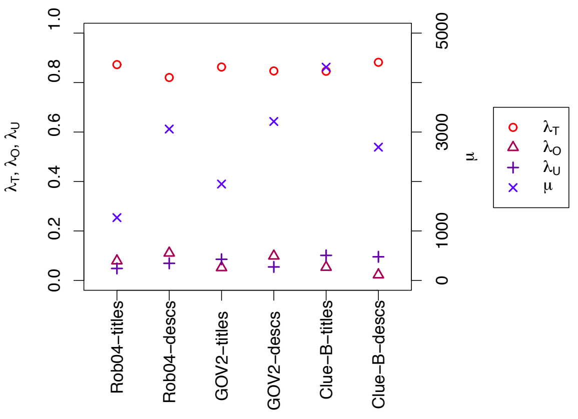

Figure 1: Average parameter settings across the 5 tuned folds for each collection, and each type of query. Left axis indicates values for each of the λ parameters, right axis indicates values for μ.

parameters are similar for all collections and query types, and the parameters are close to the suggested default parameters. Further, we observe very low standard deviation ($\sigma_p < 0.02$) across query folds, for each feature parameter, p, and each collection and query set. These observations imply that the learned parameters are stable, and that optimal parameters for one collection could be used effectively on another collection.

We next investigate the retrieval performance differences between default and tuned parameters. Table 5 shows the retrieval performance of the default parameter settings, and the tuned parameter settings. Results over the smallest collection, Robust-04, do not change significantly after tuning the model parameters. This is likely to be because the original paper presented results over some subsets of this collection. However, appropriate tuning of model parameters results in significant improvements in effectiveness for the larger GOV2 and Clueweb-09-Cat-B collections.

We make similar observations for all tested retrieval models that have suggested parameter settings. These observations demonstrate that even small changes in parameter values can have a significant effect on retrieval model effectiveness. Further, it is clear that if the suggested parameter values are naïvely used in a benchmark retrieval model, its performance may be significantly diminished. Reduced or low performance for a benchmark method is likely to produce erroneous conclusions about significant improvements for a proposed model. While this is not a new observation [2], it deserves restating.

5.2 Bi-term dependency model comparison

The section presents results from the systematic comparison of bi-term dependency models. This comparison allows us, for the first time, to determine the relative effectiveness of the different dependency models that have been proposed. Recall that each model is evaluated using 5-fold cross validation, with each fold tuned using a coordinate ascent algorithm [18], for each collection, and each type of query.

A summary of results is displayed in Table 6. This table shows significant differences, for the MAP metric, between each of the bi-term dependency models and SDM as the baseline model. Significant differences are evaluated using the Fisher randomization test and indicated in each table ($\alpha = 0.05$). Details of optimized results, as aggregated across query folds, for each model, each collection, and each type of query, are shown in Table 7. Significant improvement or degradation, relative to the query likelihood (QL), is indicated in this table.

Table 4: Collection details for collections used in this study. The ClueWeb-09-Cat-B collection has been filtered to the set of documents in the 60^{th} percentile of spam scores. The TREC Web Track 2012 competition used the ClueWeb-09-Cat-B document collection, but provides 50 new topics.

	Vocabulary	Document Count	Collection Length	Query Count
Robust-04	0.6 M	0.5 M	252 M	250
GOV2	35 M	25 M	22,332 M	150
ClueWeb-09-Cat-B	38 M	34 M	26,071 M	150

Table 5: Comparison of default to tuned parameter settings for SDM. Significance testing is performed between default and tuned results using the Fisher randomization test ($\alpha = 0.05$). Significant improvement over default parameters is indicated^{+}.

Collection	Model	MAP	nDCG@20	P@20
Robust-04, Title	SDM Default	0.264	0.424	0.374
	SDM Tuned	0.263	0.423	0.372
Robust-04, Desc.	SDM Default	0.255	0.404	0.345
	SDM Tuned	0.258	0.406	0.349
GOV2, Title	SDM Default	0.320	0.441	0.546
	SDM Tuned	0.326^{+}	0.449	0.557
GOV2, Desc.	SDM Default	0.273	0.413	0.506
	SDM Tuned	0.283^{+}	0.414	0.518^{+}
Clueweb-09-B, Title	SDM Default	0.103	0.224	0.320
	SDM Tuned	0.108^{+}	0.239^{+}	0.343^{+}
Clueweb-09-B, Desc.	SDM Default	0.076	0.180	0.226
	SDM Tuned	0.078	0.200^{+}	0.255^{+}

It is clear from this data that WSDM-Int is the most consistently effective bi-term dependency model. It significantly outperforms all other retrieval models in several settings. We also note that SDM is a very strong retrieval model, it significantly outperforms several other models on the Robust-04 and GOV2 collections, as is shown in Table 6. This data also shows that WSDM-Int significantly outperforms SDM in several settings.

All other bi-term retrieval models show some significant improved performance, relative to the QL baseline. Interestingly, the BM25-TP2 model does not perform well, even showing significant degradation of performance in some cases. This may be because this model does not control the contribution of bi- term features using a parameter, resulting in the score contribution of bi- terms is being overvalued, relative to the contribution of terms.

These results also confirm previously published findings; bi-term models can consistently improve information retrieval on the Robust-04 and GOV2 collections. Interestingly, the significant improvements observed on the Robust-04 and GOV2 collections, are much lower on the Clueweb-09-Cat-B collection. The relatively low performance improvements with dependency models using the current Clueweb-09 queries has also been observed at TREC and in recent publications [4, 23]. As more queries are developed for this corpus, we plan to study this issue further.

When comparing the performance of short and long queries, we observe that the use of bi-term proximity dependencies produces much larger improvements for description topics, than for title topics, for the Robust-04 and GOV2 collections. One obvious cause for this is that many more bi-term dependencies are extracted from the longer description topics. We also observe that for the Robust-04 collection, the effectiveness of the best performing model on description topics, (WSDM-Int), is significantly better than the effectiveness with title topics, using the MAP and ERR@20 metrics. For this collection, WSDM-Int is able to extract more informative features from the long queries compared to the associated short queries.

5.3 Many-term model comparison

Based on the results from the comparison of bi-term dependency models, we select SDM and WSDM-Int as benchmark methods against which to evaluate the benefit of many-term dependencies for retrieval effectiveness. We evaluate this by investigating three existing many-term dependency retrieval models; BM25-Spans, PLM, and PLM-2. We also study variants of the SDM and WSDM-Int retrieval models using many-term dependencies. We construct SDM variants that use dependent sets of three and four terms, as described in Section 3. Finally, we construct a variant of WSDM-Int, WSDM-Int-3 which includes three-term dependencies. Note that for the extended SDM and WSDM-Int models, all term dependencies are extracted sequentially from the query.

Similar to the comparison of bi-term models, each many-term dependency model is evaluated using 5-fold cross validation, where each fold is tuned using a coordinate ascent algorithm [18]. We focus on topic descriptions in this section, as title queries are frequently too short to extract dependent sets of more than two terms.

Note that PLM and PLM-2 are only tuned for the Robust-04 collection. To evaluate a given document for these retrieval models, for a particular query, each position in the document must be evaluated, and the maximum score is returned. In the worst case, every position in the collection must be scored independently. To reduce this overhead, we implement a simulated annealing algorithm to reduce the number of positions tested to locate the maximum scoring position in the document. This algorithm reduced the time to evaluate queries by a factor of 20, without any measurable change in retrieval effectiveness. However, even with this efficiency optimization, and the document-length estimation optimization presented by Lv and Zhai [16], tuning the parameters of this retrieval model using coordinate ascent over larger collections remains infeasible. In lieu of tuned optimal parameters, we report results for GOV2, and Clueweb-09-Cat-B using the optimal parameters from Robust-04.

Table 6: Significant differences, using the MAP metric, between bi-term dependency models and SDM as the baseline, computed using the Fisher randomization test ($\alpha = 0.05$). The first letter of each model is used to indicate a significant improvement over the paired model, '$-$' indicates no significant difference is observed.

	Robust-04		GOV2		Clueweb-09-B	
Models	Titles	Desc.	Titles	Desc.	Titles	Desc.
SDM / BM25-TP	–	S	–	S	–	–
SDM / BM25-TP-2	S	S	S	S	–	–
SDM / pDFR-BiL2	S	S	S	S	S	–
SDM / pDFR-PL2	–	S	S	S	P	–
SDM / WSDM-Int	W	W	–	W	W	–

Table 7: Comparison of dependency models over Robust-04, GOV2, and Clueweb-09-Cat-B collections. Significant improvement or degradation with respect to the query likelihood model (QL) is indicated ($^{+}/^{-}$).

	Robust-04 collection							
	Topic titles				Topic descriptions			
Model	MAP	nDCG@20	P@20	ERR@20	MAP	nDCG@20	P@20	ERR@20
QL	0.252	0.412	0.365	0.113	0.244	0.389	0.334	0.115
BM25	0.254	0.412	0.363	0.113	0.237^-	0.390	0.331	0.115
PL2	0.253	0.418^+	0.369	0.115^+	0.229^-	0.389	0.329	0.115
BM25-TP	0.262^+	0.418	0.371	0.114	0.243	0.394	0.336	0.114
BM25-TP-2	0.248	0.396	0.348^-	0.111	0.215^-	0.356^-	0.302^-	0.104^-
pDFR-BiL2	0.258^+	0.422^+	0.372	0.116	0.234^-	0.393	0.335	0.116
pDFR-PL2	0.260^+	0.422^+	0.375^+	0.115	0.235	0.393	0.333	0.116
SDM	0.263^+	0.423^+	0.375^+	0.115	0.258^+	0.406^+	0.349^+	0.116
WSDM-Int	0.269^+	0.432^+	0.382^+	0.119^+	0.278^+	0.428^+	0.365^+	0.123^+

	GOV2 collection							
	Topic titles				Topic descriptions			
Model	MAP	nDCG@20	P@20	ERR@20	MAP	nDCG@20	P@20	ERR@20
QL	0.298	0.413	0.511	0.164	0.257	0.378	0.472	0.149
BM25	0.299	0.435^+	0.530^+	0.174^+	0.261	0.401^+	0.484	0.161^+
PL2	0.300	0.415	0.516	0.165	0.258	0.390^+	0.479	0.155
BM25-TP	0.321^+	0.445^+	0.556^+	0.176^+	0.272^+	0.407^+	0.510^+	0.160^+
BM25-TP-2	0.273^-	0.382	0.480	0.157	0.250	0.392	0.495	0.157
pDFR-BiL2	0.313^+	0.437^+	0.536^+	0.174^+	0.266^+	0.394^+	0.486	0.155
pDFR-PL2	0.317^+	0.441^+	0.544^+	0.175^+	0.270^+	0.403^+	0.494^+	0.161^+
SDM	0.326^+	0.449^+	0.557^+	0.176^+	0.283^+	0.414^+	0.518^+	0.161^+
WSDM-Int	0.329^+	0.450^+	0.556^+	0.176^+	0.298^+	0.425^+	0.533^+	0.167^+

	Clueweb-09-Cat-B collection							
	Topic titles				Topic descriptions			
Model	MAP	nDCG@20	P@20	ERR@20	MAP	nDCG@20	P@20	ERR@20
QL	0.098	0.221	0.321	0.121	0.074	0.189	0.244	0.108
BM25	0.099	0.223	0.324	0.125	0.081^+	0.201	0.260	0.119^+
PL2	0.105^+	0.233^+	0.337^+	0.127^+	0.077^+	0.194	0.247	0.108
BM25-TP	0.109^+	0.242^+	0.349^+	0.128	0.084^+	0.201	0.258	0.112
BM25-TP-2	0.104	0.244^+	0.342^+	0.128	0.078	0.195	0.254	0.109
pDFR-BiL2	0.102^+	0.225	0.326	0.126^+	0.076	0.192	0.245	0.110
pDFR-PL2	0.111^+	0.248^+	0.358^+	0.127	0.080^+	0.193	0.244	0.107
SDM	0.108^+	0.239^+	0.343^+	0.128^+	0.078	0.200	0.255	0.114
WSDM-Int	0.113^+	0.245^+	0.354^+	0.130^+	0.083^+	0.199	0.255	0.118^+

The results from these experiments are displayed in Table 8. Significant differences are indicated in the table with respect to the SDM benchmark. We also show significance testing with respect to the WSDM-Int benchmark in Table 9.

We observe that many-term dependency models do not consistently improve retrieval performance over bi-term models. Small improvements can be observed in some cases. WSDM-Int-3, in particular, shows significant improvements over benchmark models for the Robust-04 collection. Investigation of the performance of WSDM-Int-3 for the GOV2 and Clueweb-09-B collections shows some evidence that the model is overfitting to the 4 folds of training data, thereby reducing performance on the test fold. However,

Table 8: Investigation of many-term proximity features over the Robust-04, and GOV2 collections. Significant improvements and degradation with respect to SDM are indicated ($^{+}/^{-}$).

	Robust-04, Topic desc.				GOV2, Topic desc.			
Model	MAP	nDCG@20	P@20	ERR@20	MAP	nDCG@20	P@20	ERR@20
SDM	0.258	0.406	0.349	0.116	0.283	0.414	0.518	0.161
WSDM-Int	0.278^{+}	0.428^{+}	0.365^{+}	0.123^{+}	0.298^{+}	0.425^{+}	0.533^{+}	0.167
BM25-Span	0.243^{-}	0.394^{-}	0.333^{-}	0.117	0.261^{-}	0.401	0.484^{-}	0.161
PLM	0.250^{-}	0.386^{-}	0.332^{-}	0.109^{-}	0.247^{-}	0.337^{-}	0.433^{-}	0.131^{-}
PLM-2	0.260	0.398	0.345	0.112^{-}	0.276	0.390^{-}	0.485^{-}	0.153
Uni+O234	0.258	0.409	0.351	0.118	0.279^{-}	0.407^{-}	0.511	0.157^{-}
Uni+O234+U2	0.259^{+}	0.408	0.351	0.117	0.283	0.409^{-}	0.516	0.157^{-}
Uni+O23+U23	0.258	0.411^{+}	0.353	0.118	0.281	0.409	0.516	0.157
Uni+O234+U234	0.259	0.409	0.353	0.117	0.282	0.410	0.511	0.161
WSDM-Int-3	0.280^{+}	0.428^{+}	0.364^{+}	0.123^{+}	0.297^{+}	0.425	0.531	0.167

	Clueweb-09-B, Topic desc.			
Model	MAP	nDCG@20	P@20	ERR@20
SDM	0.078	0.200	0.255	0.114
WSDM-Int	0.083	0.199	0.255	0.118
BM25-Span	0.085^{+}	0.205	0.261	0.118
PLM	0.064^{-}	0.153^{-}	0.198^{-}	0.088^{-}
PLM-2	0.075	0.190	0.239^{-}	0.108
Uni+O234	0.074^{-}	0.190^{-}	0.245^{-}	0.104^{-}
Uni+O234+U2	0.079	0.202	0.256	0.114
Uni+O23+U23	0.076	0.196	0.252	0.109^{-}
Uni+O234+U234	0.079	0.199	0.253	0.111
WSDM-Int-3	0.080	0.202	0.252	0.121^{+}

Table 9: Significance differences between pairs of dependency models, using MAP metric. Significance is computed using the Fisher randomization test ($\alpha = 0.05$). The first letter of each model is used to indicate a significant improvement over the paired model, '–' indicates no significant difference is observed.

	Robust-04	GOV2	Clueweb-09-B
Models	Desc.	Desc.	Desc.
WSDM-Int / BM25-Span	W	W	–
WSDM-Int / PLM	W	W	W
WSDM-Int / PLM-2	W	W	W
WSDM-Int / Uni+O234	W	W	W
WSDM-Int / Uni+O234+U2	W	W	–
WSDM-Int / Uni+O23+U23	W	W	–
WSDM-Int / Uni+O234+U234	W	W	–
WSDM-Int / WSDM-Int-3	W3	–	–

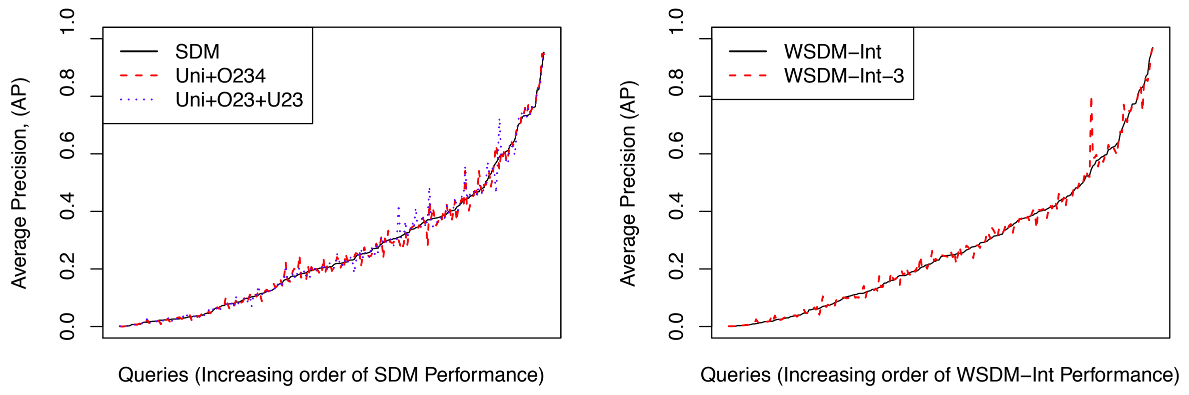

Figure 2: Per-query average precision (AP) for Robust-04, topic descriptions, using SDM and 2 related many-term proximity models.

in general, many-term features do not improve aggregate retrieval metrics with respect to the benchmark bi-term dependency models.

Figure 2 shows two per-query result graphs, one comparing SDM with two variant many-term models, and one comparing WSDM-Int to WSDM- Int-3. Data shown is only for the topic descriptions from the Robust-04 collection. Similar trends were observed for the other two collections.

This data shows that each of these models improves and degrades different queries, resulting in similar average performance. It also suggests that many- term proximity features may still be able to significantly improve retrieval performance. However, optimizing the use of many-term dependencies may require a more selective approach to the generation of sets of dependent terms, or to the selection of model features for an input query. Further, the availability of more training data, such as a large query click log, would be likely to make a significant difference to our results.

6. FUTURE WORK

Future work will include the relaxation of some of the restrictions we made to do comparisons between models. This will include studying the interactions between dependency models and methods of identifying groups of dependent terms from the query; methods of exploiting document structures; and the incorporation of information from external data sources. Each of these techniques have been demonstrated to improve retrieval over a variety of different baseline bag-of-words models, missing from the literature is a study of any relative improvements these techniques may provide to strong benchmark dependency models.

We have already started the investigation into the utility of external data sources. There is some evidence that external data sources can help significantly improve ad-hoc retrieval performance. For example, 6 of the 7 top performing models at the TREC 2012 Web Track use external data sources.

As originally proposed, WSDM [6] uses three external data sources; Wikipedia titles, the MSN query log, and Google n-grams. This model uses these features to determine term and window specific weights. By design, WSDM allows the inclusion of arbitrary external features into the weighting of each term and window, making this model appropriate for further investigation of any benefit of external data sources for dependency models.

Table 10 shows a comparison between WSDM, and one of the strongest performing dependency models, as determined in this paper, WSDM-Int. We test both models for each collection and query type, where both models are tuned using 5 fold cross validation. We observe that the external data sources used in WSDM can significantly improve the performance of this model. However, the improvements are relatively small. Future work will include the isolation and identification of the external features that provide the largest benefits for this model, and investigate alternative data sources.

7. CONCLUSIONS

In this study, we perform a systematic comparison of the retrieval effectiveness of state-of-the-art bi-term dependency models. We also present a comparison of many-term dependency models, using the best bi-term models as benchmarks. Additionally, we provide tuned parameters for a wide range of popular dependency models, for three standard test collections.

The retrieval models investigated in this study were selected to make the comparison as fair as possible. We restrict the comparison to models that use proximity-based dependencies between sequentially extracted sets of queried terms, that do not require external data sources, and do not require the use of pseudo-relevance feedback algorithms.

Our results support the well-established finding that bi-term dependency models can consistently outperform bag-of-words models. We observe that dependency models produce the largest improvements over bag-of-words models on longer queries. For the first time, we can also make some conclusions on the relative effectiveness of different dependency models. The best performing bi-term model, given the restrictions applied, is a variant of the weighted sequential dependence model. Due to our extensive parameter tuning on all models, the differences between models can be small, but they are significant. Given that all models make use of similar proximity features, it appears that it does make a difference exactly how those features are incorporated into the model. Our experiments also show that many-term dependency models do not consistently outperform bi-term models. However, these models do perform better on one of the collections (Robust-04) and per-query analysis shows that many-term proximity features have the potential to improve retrieval performance, if used in a more selective manner. Demonstrating the effectiveness of these features will, we believe, require larger query sets than are available in TREC collections.

Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF grant #CNS-0934322. We would also like to thank Michael Bendersky for several helpful technical discussions. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

References

[1] Gianni Amati and C. J. Van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, 20(4):357–389, October 2002.

[2] Timothy G. Armstrong, Alistair Moffat, William Webber, and Justin Zobel. Improvements that don't add up: ad-hoc retrieval results since 1998. In *Proc. of the 18th ACM CIKM*, pages 601–610, 2009.

[3] Michael Bendersky and W. Bruce Croft. Discovering key concepts in verbose queries. In *Proc. of the 31st ACM SIGIR*, pages 491–498, 2008.

[4] Michael Bendersky and W. Bruce Croft. Modeling higher-order term dependencies in information retrieval using query hypergraphs. In *Proc. of the 35th ACM SIGIR*, pages 941–950, 2012.

[5] Michael Bendersky, W. Bruce Croft, and David A. Smith. Two-stage query segmentation for information retrieval. In *Proc. of the 32nd ACM SIGIR*, pages 810–811, 2009.

[6] Michael Bendersky, Donald Metzler, and W. Bruce Croft. Learning concept importance using a weighted dependence model. In *Proc. of the 3rd ACM WSDM*, pages 31–40, 2010.

[7] Shane Bergsma and Qin Iris Wang. Learning noun phrase query segmentation. In *Proc. of EMNLP-CoNLL*, pages 819–826, 2007.

[8] Stefan Büttcher, Charles L. A. Clarke, and Brad Lushman. Term proximity scoring for ad-hoc retrieval on very large text collections. In *Proc. of the 29th ACM SIGIR*, pages 621–622, 2006.

[9] James P. Callan, W. Bruce Croft, and John Broglio. TREC and TIPSTER Experiments with INQUERY. In *Information*

Table 10: Comparison of the performance of WSDM-Int and WSDM [4]. $^+$ indicates a significant improvement over WSDM-Int.

Collection	Model	MAP	nDCG@20	P@20	ERR@20
Robust-04, Title	WSDM-Int	0.269	0.432	0.382	0.119
	WSDM	0.271^+	0.435^+	0.383	0.119
Robust-04, Desc.	WSDM-Int	0.278	0.428	0.365	0.123
	WSDM	0.283^+	0.431	0.366	0.125^+
GOV2, Title	WSDM-Int	0.329	0.450	0.556	0.176
	WSDM	0.331^+	0.453	0.563^+	0.176
GOV2, Desc.	WSDM-Int	0.298	0.425	0.533	0.167
	WSDM	0.303^+	0.426	0.536	0.165
Clueweb-09-B, Title	WSDM-Int	0.113	0.245	0.354	0.130
	WSDM	0.111	0.244	0.352	0.132
Clueweb-09-B, Desc.	WSDM-Int	0.083	0.199	0.255	0.118
	WSDM	0.088^+	0.219^+	0.283^+	0.125^+

Processing & Management, pages 31–3. Morgan Kaufmann, 1994.

[10] William S. Cooper. Some inconsistencies and misnomers in probabilistic information retrieval. In *Proc. of the 14th ACM SIGIR*, pages 57–61, 1991.

[11] Gordon V. Cormack, Mark D. Smucker, and Charles L. A. Clarke. Efficient and effective spam filtering and re-ranking for large web datasets. *Information retrieval*, 14(5):441–465, 2011.

[12] W Bruce Croft, Donald Metzler, and Trevor Strohman. *Search engines: Information retrieval in practice*. Addison-Wesley Reading, 2010.

[13] Jianfeng Gao, Jian-Yun Nie, Guangyuan Wu, and Guihong Cao. Dependence language model for information retrieval. In *Proc. of the 27th ACM SIGIR*, pages 170–177, 2004.

[14] Samuel Huston and W. Bruce Croft. Parameters learned in the comparison of retrieval models using term dependencies. Technical report, University of Massachusetts, 2014.

[15] Rosie Jones, Benjamin Rey, Omid Madani, and Wiley Greiner. Generating query substitutions. In *Proc. of the 15th WWW*, pages 387–396, 2006.

[16] Y. Lv and C.X. Zhai. Positional language models for information retrieval. In *Proc. of the 32nd ACM SIGIR*, pages 299–306. ACM, 2009.

[17] K. Tamsin Maxwell and W. Bruce Croft. Compact query term selection using topically related text. In *Proc. of the 36th ACM SIGIR*, pages 583–592, 2013.

[18] Donald Metzler. Using gradient descent to optimize language modeling smoothing parameters. In *Proc. of the 30th ACM SIGIR*, pages 687–688.

[19] Donald Metzler and W. Bruce Croft. A Markov random field model for term dependencies. In *Proc. of the 28th ACM SIGIR*, pages 472–479, 2005.

[20] Jae Hyun Park, W. Bruce Croft, and David A. Smith. A quasi-synchronous dependence model for information retrieval. In *Proc. of the 20th ACM CIKM*, pages 17–26, 2011.

[21] Jie Peng, Craig Macdonald, Ben He, Vassilis Plachouras, and Iadh Ounis. Incorporating term dependency in the dfr framework. In *Proc. of the 30th ACM SIGIR*, pages 843–844, 2007.

[22] Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *Proc. of the 21st ACM SIGIR*, pages 275–281, 1998.

[23] Fiana Raiber and Oren Kurland. Ranking document clusters using markov random fields. In *Proc. of the 36th ACM SIGIR*, pages 333–342, 2013.

[24] Yves Rasolofo and Jacques Savoy. Term proximity scoring for keyword-based retrieval systems. In *Proc. of the 25th ECIR*, pages 207–218, 2003.

[25] Knut Magne Risvik, Tomasz Mikolajewski, and Peter Boros. Query segmentation for web search. In *Proc. of the 12th WWW*, 2003.

[26] S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proc. of the 17th ACM SIGIR*, pages 232–241, 1994.

[27] Stephen E Robertson and K Sparck Jones. Relevance weighting of search terms. *Journal of the American Society for Information science*, 27(3):129–146, 1976.

[28] Mark D. Smucker, James Allan, and Ben Carterette. A comparison of statistical significance tests for information retrieval evaluation. In *Proc. of the 16th ACM CIKM*, pages 623–632, 2007.

[29] Fei Song and W. Bruce Croft. A general language model for information retrieval. In *Proc. of the 8th CIKM*, pages 316–321, 1999.

[30] Ruihua Song, Michael J. Taylor, Ji-Rong Wen, Hsiao-Wuen Hon, and Yong Yu. Viewing term proximity from a different perspective. In *Proc. 30th ECIR*, pages 346–357, 2008.

[31] Munirathnam Srikanth and Rohini Srihari. Incorporating query term dependencies in language models for document retrieval. In *Proc. of the 26th ACM SIGIR*, pages 405–406, 2003.

[32] Krysta M. Svore, Pallika H. Kanani, and Nazan Khan. How good is a span of terms?: exploiting proximity to improve web retrieval. In *Proc. of the 33rd ACM SIGIR*, pages 154–161, 2010.

[33] Tao Tao and ChengXiang Zhai. An exploration of proximity measures in information retrieval. In *Proc. of the 30th ACM SIGIR*, pages 295–302, 2007.