Compact Query Term Selection Using Topically Related Text

K. Tamsin Maxwell
University of Edinburgh
School of Informatics
Edinburgh EH8 9AB, UK
t.maxwell@ed.ac.uk

W. Bruce Croft
University of Massachusetts
Dept. of Computer Science
Amherst, MA 01003, USA
croft@cs.umass.edu

ABSTRACT

Many recent and highly effective retrieval models for long queries use query reformulation methods that jointly optimize term weights and term selection. These methods learn using word context and global context but typically fail to capture query context. In this paper, we present a novel term ranking algorithm, PhRank, that extends work on Markov chain frameworks for query expansion to select compact and focused terms from within a query itself. This focuses queries so that one to five terms in an unweighted model achieve better retrieval effectiveness than weighted term selection models that use up to 30 terms. PhRank terms are also typically compact and contain 1-2 words compared to competing models that use query subsets up to 7 words long. PhRank captures query context with an affinity graph constructed using word co-occurrence in pseudo-relevant documents. A random walk of the graph is used for term ranking in combination with discrimination weights. Empirical evaluation using newswire and web collections demonstrates that performance of reformulated queries is significantly improved for long queries and at least as good for short, keyword queries compared to highly competitive information retrieval (IR) models.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Query Formulation

Keywords

Random Walk; Verbose Queries; Query Reformulation

1. INTRODUCTION

Query reformulation is a rich area of information retrieval (IR) research, including techniques for query expansion, dependency analysis, query segmentation and term selection. For short queries, IR effectiveness is improved by smoothing and query expansion using techniques such as pseudo relevance feedback [14, 23, 34]. Conversely, long or verbose queries contain words that are peripheral or shared across many topics so expansion is prone to query drift. Reformulation instead focuses on term weighting [18, 5], term selection [3, 2] and query reduction [15]. The selection of informative terms, defined as one or many-word units, becomes critical as the number of potentially noisy terms increases.

Techniques for term selection and term weighting automatically emphasize the 'essence' of a query. Several successful techniques jointly optimize weights and term selection using both global statistics and local syntactic features [3, 33]. However, these features can fail to detect or differentiate *informative* terms, where an informative term represents the essential aspects of query meaning given a collection of documents. Global statistics are strong indicators of term importance [4] but do not reflect local query context. There is also evidence that they do not lead to significant improvement in query effectiveness [20]. Syntactic features precisely identify word relations but do not identify all the informative relations [22]. The ubiquity of global statistics and syntactic features in current methods for term selection suggests a continuing need for improved understanding of alternatives ways to estimate term informativeness [20].

In this paper, we present PhRank (phrase rank), an algorithm that uses pseudo relevance feedback for in-query term selection rather than expansion to terms not in a query. Compact and focused terms are selected from a list of candidates by ranking terms using a Markov chain framework and picking the top-ranked candidates. Candidate terms are all combinations of 1-3 words in a query that are not stopwords. Term scores are computed using the average word score for words in a term, combined with global discrimination weights. Word scores are computed using a random walk of a word co-occurrence graph constructed from pseudo relevant documents, combined with word salience weights for the query and global contexts. This approach selects terms that achieve significant gains in both recall and precision compared to the most effective techniques for query reformulation that do not use term weighting, expansion, or stratified dependencies [4]. This is achieved by focusing on a limited number of word relationships with a core concept.

PhRank has three advantages compared to previous work. First, to our knowledge PhRank is the first method to use pseudo relevance feedback for in-query term selection. Feedback was applied initially to estimate weights for independent query words without selection [8] and is predominantly used for query expansion. Previous approaches to in-query term selection use highly localized *word context*, in the form

SIGIR'13, July 28–August 1, 2013, Dublin, Ireland.
Copyright 2013 ACM 978-1-4503-2034-4/13/07 ...$15.00.

Query: *Locations of volcanic activity which occurred within the present day boundaries of the U.S. and its territories.*

PhRank	Sequential Dependence	Key Concept	Subset Distribution
volcanic	locations volcanic	present day boundaries	volcanic day boundaries
volcanic boundaries	volcanic activity	volcanic activity	day boundaries territories
volcanic territories	activity which		volcanic activity occurred day boundaries
volcanic activity	which occurred		present day boundaries
volcanic occurred	occurred within		volcanic boundaries territories
	within present		volcanic activity occurred
	present day		activity occurred day boundaries
	day boundaries		volcanic activity occurred boundaries
	boundaries us		volcanic present day boundaries
	us territories		volcanic occurred boundaries
			+ 20 bigrams (*if weights collapsed*)

Table 1: Terms selected by four highly effective query reformulation models for TREC GOV2 topic #756.

of syntactic relations and co-occurrence, and *global context* in the retrieval collection. They do not consider *query context*, such as a general query topic identified from pseudo relevant documents. The intuition behind a random walk of a query context graph is that it reinforces words that capture query 'essence' more strongly than words that are peripheral to query meaning. For this reason, informative terms are more readily apparent if query context is considered.

Second, PhRank achieves significant performance gains with a small number of compact terms while retaining the flexibility to select more and longer terms if required. Other approaches use a robust, but less effective, distribution over many imprecise but approximately relevant terms. Alternatively, they take a relatively inflexible, high-risk approach that prefers a few exact terms and is prone to mistakes. For example, Table 1 shows the terms selected for TREC topic #756 by three top performing IR models. The sequential dependence (SD) model is straightforward and *robust* [24]. The key concept (KC) model [3] aims at a highly *succinct* representation but is hampered by a requirement that terms are predefined syntactic units (noun phrase length). The subset distribution (SDist) model [33] optimizes over many term and weight variables and is *highly effective* but is biased towards longer terms of 3-6 words. PhRank demonstrates that for a majority of queries, a few precise terms, in addition to a standard query likelihood representation, are more effective than term distributions. They also result in queries that have up to 90% fewer terms, and these terms are typically only 1-2 words long.

Finally, an affinity graph captures aspects of both syntactic and non-syntactic word associations in an integrated manner. A co-occurrence affinity graph shares the same structure as a global dependency graph in which edges are defined by linguistic relations. Specifically, the most connected vertices are high frequency functional words and less frequent content-bearing words tend towards the edges of the graph [10, 11]. By consequence, the semantic significance of a word is correlated with the degree of the corresponding vertex. We infer that the shared structure of dependency and affinity graphs captures aspects of both syntactic and non-syntactic word associations. Moreover, an affinity graph can be used to estimate the semantic significance of words.

To summarize, unlike existing models of term selection, PhRank integrates three characteristics that we believe are important to accurately identify the most informative terms: query context, compactness, and integration of syntactic and semantic knowledge. We show that consolidating these characteristics delivers up to 14% performance improvement compared to highly competitive methods for TREC description topics and is comparable to the state-of-the-art for TREC keyword (title) queries.

The rest of this paper is organized as follows. In Section 2 we review related work and its connection to PhRank. Section 3 defines the problem of term selection and its key characteristics. In Section 4 we formally describe the PhRank algorithm. Section 5 presents the evaluation framework. In Section 6 we discuss the results of empirical experiments, and Section 7 concludes the paper.

2. RELATED WORK

Markov chain frameworks and spreading activation networks for a network of words are well-studied in IR with origins in associative word networks [7]. They include research on webpage authority [26], e.g. PageRank, as well as query expansion [17, 6, 23, 14]. However, they are novel for *unexpanded* term selection.

The Markov chain framework uses the stationary distribution of a random walk over an affinity graph G to estimate the importance of vertices in the graph. Vertices can represent words, in which case edges represent word associations. If the random walk is ergodic, affinity scores at vertices converge to a stationary distribution that can be used to establish a ranking, e.g. over words.

A random walk describes a succession of random or semi-random steps between vertices v_i and v_j in G. Let ℓ_{ij} be the transition probability (or edge weight) between v_i and v_j. The path of the walk is determined by a square probability matrix $H = (h_{ij})$ with size n, where n is the number of unique vertices in G. The probability $h_{ij} = \ell_{ij}$ if v_i and v_j are connected, and $h_{ij} = 0$ otherwise. Affinity scores are computed recursively. Let π_j^t be the affinity score associated with v_j at time t. Then π_j^{t+1} is the sum of scores for each v_i connected to v_j, weighted by the possibility of choosing v_j as the next step on the path from v_i:

$$\pi_j^{t+1} = \sum_i \pi_i h_{ij} \tag{1}$$

It is usual to introduce some minimal likelihood that a path from v_i at time t will randomly step to some v_j at time $t+1$ that may be unconnected to v_i. Otherwise, clusters of vertices interfere with the propagation of weight through the graph. This likelihood is often defined to be the uniform probability vector $u = 1/n$, although any other vector can be chosen [14]. A corresponding factor reflects the likelihood that a path will follow the structure of edges in G. A damping factor α controls the balance between them:

$$\pi^{t+1} = \alpha\pi^{t}H + (1-\alpha)u \qquad (2)$$

The Markov chain framework has has been used in a principled way to smooth and expand queries in a language modeling framework [34], but application in query reformulation has been limited to selection of individual words that do not appear in the original query. By contrast, PhRank ranks terms containing one or more words that do appear in the original query. Moreover, while expansion techniques can exacerbate problems with unrelated terms, PhRank reduces the problem of query drift through improved term selection.

Markov chain processes have also been applied in text summarization for keyphrase extraction. This is a task similar to term detection for automated indexing. TextRank [25], SingleRank [32] and ExpandRank [32] use a random walk to identify salient sequences of nouns and adjectives. They improve over earlier unsupervised methods for this task but achieve only 30-40% task accuracy and may be outperformed by a *tf.idf* metric [13]. ExpandRank supplements text with pseudo relevant documents but does not improve performance compared to SingleRank [13]. PhRank is similar to these algorithms but is more flexible and better suited to IR. It uses multiple sources of co-occurrence evidence and the discriminative ability of terms in a collection. It also produces an unbiased ranking over terms of mixed lengths, does not rely on syntactic word categories such as nouns, and permits terms to contain words with long distance dependencies.

Other related work focuses on techniques for identification of dependent terms [28], key concepts [3], or sub-queries [16, 33]. This includes techniques for the removal of stop structure [15]; reduction of narrative queries to word sequences associated with part of speech blocks [19]; selection of candidate sub-queries using noun phrases [3]; query term ranking using dependency tree relations [27]; and optimized ranking over possible subqueries [33]. There is also a significant body of work on learning individual term weights [18, 5]. Much of this work incorporates syntactic and statistical features in machine learning.

3. PRINCIPLES FOR TERM SELECTION

We hypothesize that the following principles define word and term informativeness. These principles motivate the PhRank algorithm detailed in the next Section.

An informative word:

1. **Is informative relative to a query:** An informative word should accurately represent the meaning of a query. However, queries do not provide much context with which to determine meaning. Pseudo relevance used in PhRank is an established means of enhancing a query representation [29].

2. **Is related to other informative words:** The *Association Hypothesis* [31] states that, "if one index term is good at discriminating relevant from non-relevant documents, then any closely associated index term is also likely to be good at this". PhRank uses a Markov chain framework in which the value assigned to a word i is determined by the value of other words connected to i, and the number of connections to i.

An informative term:

3. **Contains informative words:** Consider a base case in which a term has only one word. It is obvious that this term must also display the properties of an informative word. We deduce that all terms must contain informative words. PhRank considers the informativeness of individual words when ranking terms.

4. **Is discriminative in the retrieval collection:** A term that occurs many times within a small number of documents gives a pronounced relevance signal. PhRank weights terms with a normalized *tf.idf* inspired weight.

4. THE PHRANK ALGORITHM

PhRank captures query context with an affinity graph constructed from stopped, stemmed pseudo-relevant documents. Vertices in the graph represent unique stemmed words (or simply, *stems*). Edges connect stems that are adjacent in the processed pseudo relevant set. Graph transition probabilities (edge weights) are computed using a weighted linear combination of stem co-occurrence, the certainty that the document in which they co-occur is relevant, and the salience of sequential bigram factors in the pseudo relevant set. The edge weights thus represent the tendency for two stemmed words w_i and $w_{j\neq i}$ to appear in close proximity in documents that that reflect a query topic.

Stems in the affinity graph are scored using a random walk algorithm. Following convergence, stem scores are weighted by a *tf.idf* style weight that further captures salience in the pseudo relevant set. This aims to compensate for potential undesirable properties of the random walk. Finally, term scores are computed using the average score for stemmed words in a term, weighted by term salience in the retrieval collection. The m highest scoring terms are employed to reformulate Q. Pseudo code for the algorithm is shown in Figure 1. The rest of this section describes the algorithm in more detail, including three heuristic weights (factors r, s and z). A number of choices for these factors could have been made and specific choices are analyzed in Section 6.1.

1) Graph construction (principle 1):

Let a query $Q = \{w_1, ...w_n\}$ and C be a document collection. The top k documents retrieved from C using Q are assumed to describe a similar topic to Q. We define C to be the retrieval collection plus English Wikipedia. We use Wikipedia since it improves IR results for query expansion using a random walk [6], but also explore the effectiveness of using the retrieval collection alone. The top k documents in C, together with Q itself encoded as a short document d_0, comprise *neighboring* documents in the neighborhood set $N = \{d_0, ...d_k\}$.

Documents in N are stopped using a minimal list of 18 words [21] and stemmed using the Krovetz stemmer. This improves co-occurrence counts for content-bearing stems and reduces the size of an affinity graph G constructed from the processed documents. Stoplisting with a longer list hurt IR effectiveness. Edges in G connect stemmed words i and j at vertices v_i and v_j if i and j are adjacent in N. Documents in N with only one word (e.g. some queries) are discarded to ensure that all vertices have at least one connecting edge.

2) Edge weights (principle 1):

Transition probabilities (edge weights) ℓ_{ij} are based on a weighted linear combination of the number of times i and

```
k = 5
resourceList = [ C, wikipedia ]

for q in queryList:

  N = set()
  for rsc in resourceList:
    N.add( retrieve_top_k( q, rsc ) )
  N = retrieve_top_k( q, N )
  N.add( q )

  # one word type per row and column
  G = arrayStruct()
  for ( doc, docRel ) in N:
    doc.stopStem()
    G.grow( buildGraph( doc, docRel ) )

  G.idfWeightEdge()            # bigram wt r
  G.normalize()
  G.iterate()
  G.weightVertex()              # word wt s

  T = q.terms
  for term in q:
    term.wt = G.score( term )
    term.wt *= term.globalWt( C ) # term wt z
  T.sortByWeight()
```

```
def buildGraph( doc, docRel ):
  docG = index( doc )
  docG.linearWt( uw2, uw10 )
  docG.weight( docRel )

  return docG


def score( term ):
  S = 0

  for w in term.wordSplit():
    S += self.affinityScore( term )

  return S /= term.length()


def globalWt( C ):
  l = self.length()
  wt = C.tfidf( self ) * l^l

  return wt
```

Figure 1: Pseudocode for the PhRank algorithm.

j co-occur in windows W of size 2 and 10. This is motivated by the idea that different degrees of proximity provide rich evidence for word relatedness in IR [25, 32, 24]. Edge weights are defined by:

$$\ell_{ij} = r * \sum_{d_k \in N} p(d_k|Q)(\lambda c_{ij_{W_2}} + (1-\lambda) c_{ij_{W_{10}}})$$

where $p(d_k|Q)$ is the probability of the document in which the stems i and j co-occur given Q, and $c_{ij_{W_2}}$ and $c_{ij_{W_{10}}}$ are the counts of stem co-occurrence in windows of size 2 and 10 in N. λ is set to 0.6. We set the relevance of d_0 to Q to be high but reasonable (-4 for Indri log likelihood scores). The exact setting has very little effect on term ranking.

Factor r is a *tf.idf* style weight that confirms the importance of a connection between i and j in N. G includes many stemmed words, so unweighted affinity scores can be influenced by co-occurrences with highly frequent, but possibly uninformative, stems such as '*make*'. Factor r minimizes this effect. Since the *tf* component is already accounted for by $\lambda c_{ij_{W_2}} + (1-\lambda) c_{ij_{W_{10}}}$, we reduce r to the *idf* component:

$$r_{ij} = log_2 \frac{\sum_{ij \in N} c_{ij_{W_2}}}{1 + c_{ij_{W_2}}}$$

3) Random Walk (principle 2):

A random walk of G follows the standard Markov chain framework presented in Section 2. Edge weights are normalized to sum to one and π_j is the affinity score of the stem associated with v_j. π_j indicates the importance of a stem in the query context. Iteration of the walk ceases when the difference in score at any vertex does not exceed 0.0001. This translates to around 15 iterations but may be optimized for efficiency. The damping factor $\alpha = 0.85$ is equivalent to a walk along five connected edges in G before the algorithm randomly skips to a possibly unrelated vertex. The average sentence length in English is around 11-15 words so this equates to skipping at or near the boundary of a sentence around one half of the time.

4) Vertex weights (principle 3):

Following the random walk, stemmed words in G are further weighted to capture both the *exhaustiveness* with which they represent a query, and their global *saliency* in the collection [30]. Exhaustivity indicates whether a word w_1 is a sufficient representation of the query. If w_1 appears many times in N then it is less likely that a term x containing w_1 will benefit from additional words $w_2 ... w_n$. For example, the term *geysers* quite exhaustively represents the TREC query #840, '*Give the definition, locations, or characteristics of geysers*'. A term containing additional words, e.g. *definition geysers*, is not more informative. However, common stems, such as '*make*', tend to have high affinity scores because they co-occur with many words.

Factor s balances exhaustivity with global saliency to identify stems that are poor discriminators been relevant and non-relevant documents. Specifically, $s_{w_n} = w_n f_{avg} * idf_{w_n}$, where $w_n f_{avg}$ is the frequency of a word w_n in N, averaged over $k+1$ documents (the average frequency) and normalized by the maximum average frequency of any term in N. As usual, idf_{w_n} is the inverse document frequency of w_n in the collection, so $idf_{w_n} = log_2 \frac{|C|}{1+df_{w_n}}$ where $|C|$ is the vocabulary of stemmed words in the collection C, and df_{w_n} is the number of documents in C containing w_n.

An advantage of factor s is that it enables PhRank to be independent of an IR model. A model may treat the component words of terms as independent or dependent. Factor s helps to ensure that the selected terms are informative irrespective of this representation.

5) Term ranking (principles 3, 4):

To avoid a bias towards longer terms, a term x is scored by averaging the affinity scores for its component words $\{w_1, [...w_n]\}$. Term rank is determined by the average score multiplied by a factor z_x that represents the degree to which the term is discriminative in a collection:

$$z_x = f_{x_e} * idf_{x_e} * l_x$$

Let x_e be a proximity expression such that the component words of x appear in an unordered window of size $W = 4$ per word. Thus, a term with two words appears in an 8-word window, and a term with three words appears in a 12-word window. The frequency of x_e in C is f_{x_e} and idf_{x_e} is defined analogously to idf_{w_n} above. l_x is an exponential weighting factor proposed for the normalization of ngram frequencies during query segmentation [12]. This factor favors longer ngrams that tend to occur less frequently in text. Multiplication of ngram counts by l_x enables comparison of counts for terms of varying length. Let $|x|$ be the number of words in x, then $l_x = |x|^{|x|}$.

In summary, the PhRank algorithm describes how informative a term x is for Q compared to other terms. This is computed using the function:

$$f(x,Q) \overset{rank}{=} z_x * \frac{\sum_{w_n \in x} \pi_{w_n}}{n} \tag{3}$$

4.1 Diversity filter

PhRank often assigns a high rank to multi-word terms that contain only one highly informative word. This is due to use of an average word affinity score, and is desirable because informative terms can contain uninformative words. For example, given a query about '*the destruction of Pan Am Flight 103 over Lockerbie, Scotland*' (TREC #409), the term '*pan flight 103*' is informative even if the polysemous word '*pan*' is uninformative by itself. However, this can result in low diversity of top ranked terms used in query reformulation. To increase diversity, we apply a simple, heuristic filtering technique with top-down constraints.

Given a ranked list, all terms with a score of zero are discarded. The lowest ranked term, x_n, is checked against the list of terms with a better rank $x_{m<n}$. Let A be the set of component words in x_n and B be the set of component words in any single term $x_{m<n}$. If $\exists A : A \subset B \vee A \supset B$ then we discard x_n *iff* every component word of x_n is contained in at least one $x_{m \neq n}$ that is in the retained list of ranked words at the time B is evaluated. For example, if x_n ='*birth rate*' and we find some $x_{m<n}$ ='*birth rate china*' then we discard x_n on the assumption that the longer term better represents the information need. If x_n ='*declining birth rate*' and we find some $x_{m<n}$ ='*birth rate*' and some $x_{m<n}$ ='*declining birth*' then we discard x_n on the assumption that the shorter terms better represent the information need and the longer term is redundant. Note that the top-ranked term is always retained. This process is adequate to increase diversity in the ranked list and ensures that no vital information is lost, but clearly presents an opportunity for further improvement.

5. EVALUATION FRAMEWORK

This section describes comparative models and query reformulations used to assess the degree to which PhRank queries are robust, precise and succinct, and represent word dependency. The main point of comparison is a *robust* and highly effective IR model (SD) that uses term selection and is employed as a baseline in related work [3, 28, 33]. We also compare against the model with the highest mean average *precision* of which we are aware that is relevant to a discussion of term selection with query expansion (sDist). Finally, since compact queries are a feature of PhRank, we compare against a succinct yet competitive model that selects only two terms (KC). We note that superior IR effectiveness is possible with term weighting, but we focus on results using *unweighted terms* to more clearly demonstrate the effect of term selection alone. We also report results for query likelihood (QL) for reference even though this model uses no term selection. This is because the other models reported include a query likelihood component. We do not compare against models that use pseudo relevance feedback for expansion. Pseudo relevance feedback without expansion is a novel feature of our work that contributes to PhRank performance.

5.1 Robustness

Evaluation across three TREC collections using both description topics and title queries requires a strong, robust baseline. We use a sequential dependence (SD) variant of the Markov random field (MRF) model [24]. SD uses a linear combination of three cliques of terms, where each clique is prioritized by a weight λ_c. The first clique contains individual words (query likelihood QL), $\lambda_1 = 0.85$. The second clique contains query bigrams that match document bigrams in 2-word ordered windows ('*#1*'), $\lambda_2 = 0.1$. The third clique uses the same query bigrams as clique 2 with an 8-word unordered window ('*#uw8*'), $\lambda_3 = 0.05$. For example, the query '*new york city*' in Indri[1] query language is:

```
   #weight(
$\lambda_1$ #combine(new york city)
$\lambda_2$ #combine(#1(new york) #1(york city))
$\lambda_3$ #combine(#uw8(new york) #uw8(york city)))
```

Because it is very simple to generate SD queries, this model is regularly used as a baseline. Highly effective weighted variants have also been developed [4, 33, 28]. We compare SD with a PhRank model (PR-.F) that uses the same query format, except the second and third cliques contain PhRank terms instead of query bigrams. In addition, because PhRank terms may be 1-3 words long, we adjust the unordered window operator in the manner proposed for the full dependence variant of the MRF model [24]. Namely, the window size is 4 multiplied by the number of words in a term. Note that for a term with only one word i, the operators *#1(i)* and *#uw8(i)* equate to a search for the word i in a document. So, if two terms '*york*' and '*new york city*' are selected by PhRank, the PR-.F model has the form:

```
   #weight(
$\lambda_1$ #combine(new york city)
$\lambda_2$ #combine( york #1(new york city))
$\lambda_3$ #combine( york #uw12(new york city)))
```

PR-.F uses five terms for description topics and feature analysis experiments, and three terms for title queries (or less, if the required number of terms is not available after rank filtering).

5.2 Precision

Highly competitive performance compared to SD can be achieved by jointly optimizing possible subqueries and subquery weights using syntactic and statistical features. Among the models of which we are aware, the subset distribution model (sDist) [33] achieves the highest mean average precision on long queries using term selection with no higher order dependencies [4]. However, it is not entirely fair to compare sDist with PR-.F since sDist uses heavily optimized weights for ten subqueries. A subquery in sDist is a linear combination of a standard SD query and one selected term treated as a bag-of-words. This compares with the flat λ weights used in SD and PR-.F. Despite this, sDist is the most effective model we can use for stringent comparison that ensures real progress has been made. We therefore include sDist in our evaluation even though queries for Robust04 are not available from the authors.

5.3 Succinctness

Queries formulated with PhRank have few terms and a maximum of three words per term. To evaluate highly succinct queries we compare against Key Concepts (KC) [3]. KC is another succinct weighted linear feature model that combines two cliques. The first clique ($\lambda_1 = 0.8$) contains a bag-of-words query representation of the original query,

[1] http://www.lemurproject.org/

and the second clique ($\lambda_2 = 0.2$) combines a weighted bag-of-words representation for each of two selected terms. The top terms are selected from the set of query noun phrases using a decision tree with frequency-based features [3]. The model reduces to a weighted representation of the original query with word independence. If '*city*' and '*new york*' are the top two terms, it takes the following form, where δ is the decision tree confidence score associated with a term:

#weight(
λ_1 #combine(new york city)
λ_2 #weight(δ #combine(new york) δ city))

To compare against KC, we present a model (PR-zF2) that takes the same form but does not benefit from term weights δ. We use the two top terms selected by PhRank.

5.4 Word dependence

Assumptions of word dependence are an important issue in IR. To clarify the dependence assumptions made by PhRank we refer to four models of *phrase belief* presented by [9] (Figure 2, a-d). These models show how belief in a document $d_c \in C$ flows to belief in a query Q in an inference network, and thus how words and terms can be dependent. In PhRank, we do not perform inference, but by analogy these models aid interpretation of PhRank features.

Of the four models in Figure 2, the more general dependence assumption (d) is used by PhRank to score words, and term ranks are computed using an independence assumption (b). Even if component words of terms are not connected in G, weight is propagated through the graph such that word dependencies affect evidence for a term. PhRank factors z, s and r reflect Figure 2 models a, b and c respectively.

We speculate that optimal term selection occurs when 1) a high rank is assigned to terms that are important under all four interpretations of phrase belief according to evidence in N, 2) the rank of terms that have less evidence under one or more interpretation decreases gracefully, and 3) the ranking meets the principles of term selection proposed in Section 3.

6. EXPERIMENTS

We examine the performance of PhRank in three ways. First, we compare versions of the algorithm in which we omit specific features. Second, we compare performance of queries reformulated using PhRank top ranked terms against highly effective models for both TREC description topics and title queries. Third, we compare on a query by query basis the robustness and performance error for PhRank versus a distributed approach to term selection (SD).

We evaluate on three TREC collections using version 4.12 of Indri with Dirichlet smoothing, $\mu = 2500$. The Robust04, WT10G and GOV2 newswire and open web text collections have queries that vary substantially in length and known difficulty. Together they provide a diverse platform for experiments (Table 2). Topic 672 is excluded from the Robust04 evaluation as the collection contains no relevant documents. All collections and queries are stopped and stemmed using the INQUERY stoplist and Krovetz stemmer. Queries are further stopped to exclude 18 TREC stopwords such as '*describe*' [1]. Candidate terms are all units of 1-3 words in the power set $\mathcal{P}(q)$ of content-bearing words in Q. IR models are defined in Section 5. Pseudo relevant documents are retrieved using a sequential dependence model. Models '.F' exclude the feature represented by '.' and models '.T' include the feature. Thus, model rTsTzT includes all features.

Name	# Docs	Topic Numbers
ROBUST04	528,155	301-450, 601-700 (-672)
WT10G	1,692,096	451-550
GOV2	25,205,179	701-850

Table 2: TREC collections and topics

	ROBUST04		WT10G		GOV2	
	MAP	R-Pr	MAP	R-Pr	MAP	R-Pr
Description topics						
rTsTzT	26.65	30.05	0.00	0.00	28.83	34.55
zF	**27.32**	**30.32**	**23.68**	**26.71**	28.64	34.13
sF	26.03	29.61	21.00	25.10	27.93	33.67
rF	26.67	30.02	22.44	25.70	**28.93**	**34.65**
Title queries						
rTsTzT	24.87	29.04	21.78	**25.73**	31.49	37.26
zF	26.14	30.13	20.85	24.72	30.73	36.26
sF	25.90	30.03	20.72	24.30	31.30	36.91
rF	**26.32**	**30.25**	**21.81**	25.70	**31.59**	**37.42**

Table 3: Feature analysis results. Description topics perform best with omission of the global term weight z (zF). Title queries perform best with the omission of bigram salience weight r (rF).

6.1 Feature analysis

In this section we explore the impact of PhRank feature removal on IR effectiveness assessed using model PR-.F. Results in Table 3 show that PhRank is highly effective in selecting informative terms for a query. However, not all the features proposed consistently improve term ranking for IR. Description topics are most effective when factor z is omitted, and title queries are most effective when r is omitted.

1) Factor r: words dependent on term

Factor r imperfectly captures belief in component words dependent on belief in a term (Figure 2c). It uses global bigram statistics to scale edge weights in G. During a random walk, this affects the affinity scores for individual stemmed words. However, bigram statistics are only an approximate measure of term unity. More problematically, r relies on words in a term being co-occurrent in N. Highly informative terms are likely to have their component words connected in G, but this is not guaranteed. For terms with more than two words, edge weights in G also must be factored. Perhaps due to these limitations, r had minimal impact on IR effectiveness for title queries and could be omitted to improve algorithm efficiency.

However, we note that r is useful for description topics. We speculate that this is because the query words for description topics may be peripheral to the core information need. Spurious adjacent word dependencies in Q tend to appear in the pseudo relevant set because bigrams feature in the IR model employed for initial retrieval. Thus, if word co-occurrence in Q reflects query meaning, as typically occurs with title queries, the edges and weights used to initialise G are likely to be adequate. If word co-occurrence is spurious, the initialisation may be suboptimal. Factor r ameliorates misleading initial edge weights for description topics.

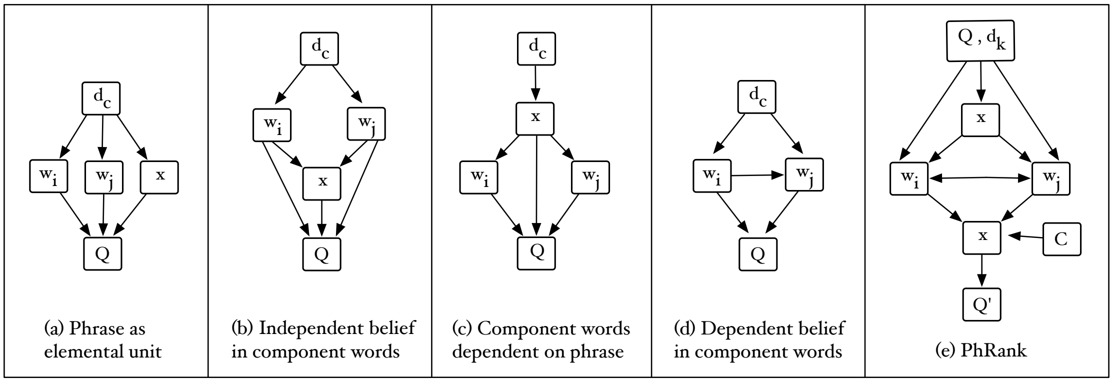
d_c
w_i
w_j
x
Q
(a) Phrase as elemental unit
(b) Independent belief in component words
(c) Component words dependent on phrase
(d) Dependent belief in component words
Q , d_k
C
Q'
(e) PhRank

Figure 2: Four models of phrase belief proposed by [9] (a-d). Word dependence in PhRank can be understood as a hybrid with features of all these models (e) for term $x = \{w_i, wj\}$ and documents $d_k \in N$.

2) Factor s: word independence
Factor s contributes to belief in a term dependent on belief in individual words (Figure 2b). It weights each vertex in an affinity graph by its salience in the query context N balanced by its salience in the document collection. Omission of s substantially hurt IR effectiveness. Among all the features tested it had the most impact on overall performance, perhaps because independent belief in words is the most important factor in IR effectiveness [24]. In addition, work with random walk algorithms for query expansion has found that words co-occurring with high frequency are of low value if they are not semantically close to the query [6]. We suggest that salience in N as captured by s represents semantic closeness to the query, and salience in the collection helps to identify high frequency co-occurrent words.

3) Factor z: term as elemental unit
Factor z represents belief in a term independent of belief in its component words (Figure 2a). It resembles a standard *tf.idf* weight and reflects the principle that a term should be discriminative in the retrieval collection. Given the established effectiveness of *tf.idf* weighting, it is surprising that omission of z improves IR effectiveness for description topics. However, it is based on observations of a term in an unordered proximity window in the retrieval collection. The way such observations are made implies a dependence assumption that may not provide an accurate estimate of term salience. In addition, it has recently been suggested that global statistics rarely improve retrieval performance and that local, document level evidence is sufficient [20].

We also note that both r and z account for the discrimination ability of multi-word units in the collection: r applies to bigrams and z applies to words in unordered windows. This encoding is partially redundant, so description queries may not require z because they use r, and title queries may require z because they do not use r. We remove z for our final runs for description queries, and retain it for title queries.

4) Factor k: pseudo relevant documents
Results in Table 4 show that the most improvement in IR effectiveness is achieved with 2 to 5 pseudo relevant documents. Higher k decreases effectiveness due to the introduction of non-relevant information. However, PhRank is quite robust to variation in k due to the weighting of co-occurrence relations by document relevance. Even with construction of the affinity graph from the original query only (¬PRF), PhRank performs better than sDist and comparably with SD. This suggests that most important information is retained by the term selection process.

	ROBUST04		WT10G		GOV2	
	MAP	R-Pr	MAP	R-Pr	MAP	R-Pr
¬PRF	26.44	29.59	21.88	25.36	27.85	33.28
k2	26.86	30.05	22.76	25.42	28.81	**34.38**
k5	**27.32**	**30.32**	**23.68**	**26.71**	28.64	34.13
k10	27.29	30.05	22.33	25.02	28.64	34.16
k50	27.09	30.11	23.04	26.21	**28.82**	34.27
k100	26.80	29.82	22.78	26.11	28.34	33.91

Table 4: IR effectiveness for description topics using k pseudo relevant documents. Best IR effectiveness is achieved using the top few documents only.

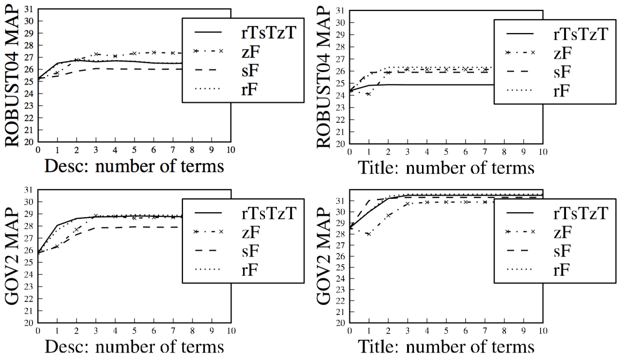

Figure 3: IR effectiveness with feature analysis and variable threshold. In many cases PhRank achieves performance gains with two terms, and is robust to variance in the number of terms selected.

6.2 Retrieval performance

We present our best results for runs using description topics and title queries on three TREC collections.

6.2.1 Robustness

For description topics, the results in Table 5 show highly significant or significant improvement in mean average precision (MAP) and R-precision compared to the SD baseline for

(a) TREC description topics

	ROBUST04		WT10G		GOV2	
	MAP	R-Pr	MAP	R-Pr	MAP	R-Pr
Robust and precise						
QL	25.25	28.69	19.55	22.77	25.77	31.26
SD	26.57	30.02	20.63	24.31	28.00	33.30
sDist	–	–	21.14	24.93	27.64	33.50
PR-zF	**27.32**	**30.32**	**23.68‡**	**26.71‡**	**28.64†**	**34.13‡**
PR¬W	27.19†	30.12	22.90†	26.57	28.18	33.77
Succinct						
KC	25.62	28.89	20.15	22.58	26.88	32.73
PR-zF2	25.91	28.92	22.02†	25.69‡	27.04	32.75
PR¬W2	25.76	28.33	21.43	25.40†	26.05	31.75

(b) TREC title queries

	ROBUST04		WT10G		GOV2	
	MAP	R-Pr	MAP	R-Pr	MAP	R-Pr
QL	24.37	28.52	19.48	23.08	28.55	34.41
SD	26.16	30.25	20.97	23.75	31.25	36.88
PR-rF	26.32	30.25	**21.81‡**	**25.70‡**	**31.59**	**37.42**
PR¬W	**26.44**	**30.40**	21.76†	25.57‡	31.50	37.14

Table 5: Retrieval results for description topics and title queries. PhRank significantly outperforms a highly effective baseline for description topics and is strongly competitive for title queries. † shows significant ($p < .05$) and ‡ highly significant ($p < .01$) results compared to SD and KC respectively as determined by a sign test.

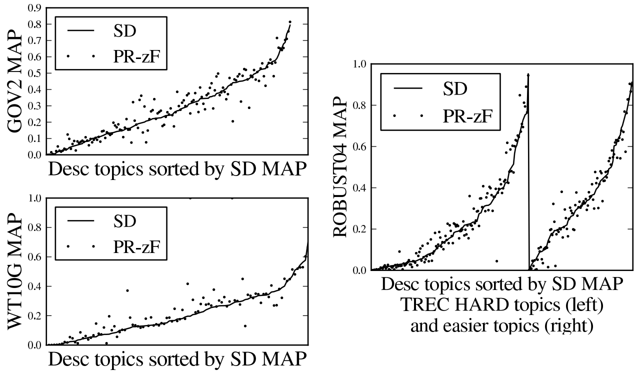

Figure 4: MAP difference compared to SD baseline per query for description topics. PhRank does slightly better on harder queries. The strategy of focus around one concept usually helps, but can significantly hurt some queries.

GOV2 and WT10G. Substantial improvements in precision on Robust04 are just short of significance. For title queries, improvement is highly significant for WT10G and comparable to the baseline for other collections. Increased precision occurs for top ranked documents (top 5 and 10) as well as being a general trend in the results. Exclusion of Wikipedia has a small negative effect as shown by PR¬W and PR¬W2 corresponding to PR-.F and PR-zF2 respectively.

To assess the quality of the ranked list of terms without a measure of ground truth for term informativeness, we explore the impact of varying the number of terms included in query reformulations. The results in Figure 3 show that the quality of top terms output by PhRank are stable as more terms are selected. Further, a large part of the gain in precision is attributed to the top two terms.

To investigate performance further, for each collection we manually reviewed the ranked term lists for queries that perform significantly better or worse than SD (>100% change in MAP), and 10 queries with comparable performance. Across all queries observed, there is a strong tendency for PhRank to single out one word, or a pair of words, as the main concept of the query, and rank all terms that contain the main concept highly. Remaining terms are ranked according to the contributions of their additional words. This high risk, high reward strategy negatively affects the robustness of PhRank on a query by query basis as shown in Figure 4 for description topics. Title queries exhibit similar behavior.

For example, one of the best performing queries for GOV2 is #756 as shown in Table 1. For this query, identification of '*volcano*' as the main concept greatly helped IR. On the other hand, the same strategy for query #780, one of the worst performing queries for GOV2 (see Table 6), selected '*earth*' as the main concept that should be included in all top terms. This resulted in terms that were representative of the query, but not well distributed.

Nevertheless, Figure 4 shows consistent improvement for queries that are known to be harder (Robust04 HARD track) or easier (high baseline MAP). It is more likely that PhRank selects an appropriate main concept for easy queries because the pseudo relevant documents are of high quality. Difficult queries are less clearly defined and often benefit from the strong directional focus provided by PhRank terms.

In comparison, models like SD and sDist, take a more robust approach to term selection with a distribution of possibly relevant terms. This presents a very different term selection strategy, so one potential avenue for improvement is interpolation of PhRank term selection with bigrams in SD. However, the robustness of a distributed term selection approach can come with a tradeoff in overall effectiveness. Initial interpolation experiments with a weighted linear combination of SD and PhRank terms did not appear to yield any benefit over PhRank terms alone.

Alternatively, the properties of G may be turned to advantage. It has been observed that a Markov field framework selects more general and robust query expansion terms than competing methods [6]. A combination of query expansion and term selection using a Markov field framework may balance complementary high reward and robust query reformulation strategies and result in significant overall gains.

6.2.2 Precision

Results in Table 5 show significant improvement in MAP and R-precision for PhRank compared to sDist for both GOV2 and WT10G. PhRank terms are significantly more precise on average than the highest precision models for unweighted term selection. Unfortunately, the focus on one aspect of query meaning has unpredictable effects and some queries are significantly hurt by a high precision strategy.

There are two potential causes for negative results. First, PhRank may be picking a suboptimal concept. This does occur, particularly in the presence of polysemous or highly co-occurrent words in the query, or irrelevant documents in N. This is demonstrated with the high rank for '*earth*' in query #780 (Table 6). In the case of highly co-occurrent

Q: *How much of planet Earth is arable at present? Area must have plenty of water, sun and soil to support plant life.*

PhRank terms	SD terms	
earth	planet earth	water sun
earth arable	earth arable	sun soil
planet earth	arable present	soil support
earth life	present area	support plant
earth water	area water	plant life

Table 6: TREC query #780: poor performance for PhRank compared to SD.

	Term Length						
	1	2	3	4	5	6	7
KC	37%	40%	15%	6%	1%	<1%	<1%
PhRank	22%	54%	24%				

Table 7: Percentage of PhRank and KC terms with various lengths.

words, these have a higher in-degree in G so they tend to accumulate weight during a random walk. A reduction in the number of iterations may help address this problem.

Irrelevant documents in G also hurt performance. The adequacy of an affinity graph G constructed using N is highly reliant on the quality of the initial query, the precision of the document similarity metric, and the adequacy of the collection being searched. If non-relevant documents occur in N there will be reduced connectivity in G, and this has an undesirable impact on the balance of word affinity scores. One solution to this problem may be to merge ranked lists computed by PhRank using different resources. The mistakes made by different instances of PhRank for the same query are likely to be less consistent than accurate assessments of term informativeness.

Second, more than one focus can occur, particularly in long queries. For example, there are two focal concepts of query #336: '*A relevant document would discuss the frequency of vicious black bear attacks worldwide and the possible causes for this savage behavior*'. The two core concepts are '*black bear*' and '*savage behavior*' but PhRank largely misses the importance of black bears. Instead, its top ranked terms for this query are {*savage, savage behavior, bear savage, vicious savage, attacks savage*}. This has a negative effect on IR effectiveness.

6.2.3 Succinctness

Results show that the performance of the top two PhRank terms in the same query structure as KC but *with no term weighting* performs comparably to KC *with term weighting.* The length of the terms is similar in both models, with around 75% of terms having a length of 1-2 words. This suggests that improved performance of unweighted PR-zF2 queries is more likely to be due to differences in the strategy for term selection than differences in term length. Note that KC shares the distributed approach to term selection with SD and sDist. KC selects two distinct concepts, whereas the top two terms selected by PhRank typically overlap.

More generally, it is observed that the succinct terms selected by PhRank are also novel. Table 8 shows that although PhRank and KC have the same number of 1-2 word terms overall, they display less than half of their potential overlap (we account for fewer terms in KC in this figure). Moreover, around 50% of PhRank terms contain two words, but only around half of them are also selected by SD.

PhRank (1-3 words)	**SD** (2 words)	**sDist** (3-6 words)	**KC** (1-7 words)
ROBUST04	23%	-	12%
WT10G	28%	11%	18%
GOV2	27%	15%	16%

Table 8: Percentage of PhRank terms selected by other models. Low figures show that PhRank is detecting novel terms with long-range dependencies.

Terms that are three words long dominate sDist (69% of all terms) yet less than half of the terms with three words in PhRank are also found in sDist queries. One likely explanation for these findings is that PhRank is not limited by syntactic or adjacency relations that are used in the other models. It detects distant word dependencies because repeat co-occurrences of word combinations reflect the associations in which they take part.

We hypothesize that distant associations may be present in queries because users condense information by relying on the ability of a search engine to infer links between words. The frequency of such textual economy was assessed in a sample of 100 queries randomly selected from Robust04 and GOV2. We assume that title queries capture a succinct information need and have informative associations between query words. By aligning description and title vocabulary, we discovered that 22% of description topics contain at least one informative word association that cannot be detected using any form of syntax or word adjacency, and a further 11% of topics contain at least one association that can only be detected using dependency relations.

7. CONCLUSION

We have presented PhRank, a novel term ranking algorithm that extends work on Markov chain frameworks for query expansion to select focused and succinct terms from within a query. PhRank captures query context with an affinity graph constructed using word co-occurrence in pseudo-relevant documents. A random walk of the graph is used for term ranking in combination with discrimination weights.

We showed that PhRank focuses on a limited number of words associated with a core query concept. Overall, this is more effective for both description topics and title queries than a distributed approach to term selection, and can generate queries with up to 90% fewer terms. However, this term selection strategy is risky and less robust than competing methods. For all collections, around 26% of queries have more than 5% decrease in MAP compared to SD (significant change is around 3-6%).

The two main issues affecting robustness are the handling queries with multiple concepts, and variation in the quality of pseudo relevance feedback. The first issue may be addressed by a diversity constraint on top ranked terms that adjusts the number of selected terms permitted to include the highest scoring query word. Improved sensitivity of the ranking algorithm may also improve results. For example, the present implementation does not consider the degree of connection between two words in the affinity graph when scoring terms. A third approach might apply a non-linear interpolation of SD and PhRank that backs off to distributed terms where required. Adaptive methods for the selection of k can address challenges with the depth of coverage in a collection or occasions when evidence for multiple concepts is widely dispersed.

Finally, the high precision strategy of term selection might be combined with the more conservative and robust expansion terms generated with a Markov chain approach to query expansion. On this point, we note that although weighted variants of an affinity graph have been proposed before, our concrete suggestion for a vertex weight s based on word salience in pseudo-relevant documents improves the informativeness of affinity scores and may benefit other techniques that use a Markov chain framework.

More generally, the work in this paper may be applicable to lexical feature selection methods for other areas of IR, including text-based image and multimedia retrieval or matching of search advertisements. Efficiency considerations surrounding the time to construct an affinity graph may be ameliorated by off-line indexing to precompute a language model for each document in a collection.

8. ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval. 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] J. Allan, L. Ballesteros, J. P. Callan, W. B. Croft, and Z. Lu. Recent experiments with INQUERY. In *Fourth Text REtrieval Conference (TREC-4)*. 1995.

[2] N. Balasubramanian, G. Kumaran, and V. R. Carvalho. Exploring reductions for long web queries. In *Proc. of SIGIR 2010*, pages 571–578, New York, NY, USA, 2010. ACM.

[3] M. Bendersky and W. B. Croft. Discovering key concepts in verbose queries. In *Proc. of SIGIR 2008*, pages 491–498, New York, NY, USA, 2008. ACM.

[4] M. Bendersky and W. B. Croft. Modeling higher-order term dependencies in information retrieval using query hypergraphs. In *Proc. of SIGIR 2012*, pages 941–950, New York, NY, USA, 2012. ACM.

[5] M. Bendersky, D. Metzler, and W. B. Croft. Learning concept importance using a weighted dependence model. In *Proc. of WSDM 2010*, pages 31–40, New York, NY, USA, 2010. ACM.

[6] K. Collins-Thompson and J. Callan. Query expansion using random walk models. In *Proc. of CIKM 2005*, pages 704–711, New York, NY, USA, 2005. ACM.

[7] F. Crestani. Application of spreading activation techniques in information retrieval. *Artif. Intell. Rev.*, 11(6):453–482, Dec. 1997.

[8] W. B. Croft and D. Harper. Using probabilistic models of document retrieval without relevance information. *Journal of Documentation*, 35(4):285 – 295, 1979.

[9] W. B. Croft, H. R. Turtle, and D. D. Lewis. The use of phrases and structured queries in information retrieval. In *Proc. of SIGIR 1991*, pages 32–45, New York, NY, USA, 1991. ACM.

[10] R. Ferrer i Cancho and R. Solé. The small world of human language. In *Proc. of the Royal Society of London B*, volume 268, 2001.

[11] R. Ferrer i Cancho, R. V. Solé, and R. Köhler. Patterns in syntactic dependency networks. *Physical Review E*, 69(5), 2004.

[12] M. Hagen, M. Potthast, B. Stein, and C. Bräutigam. Query segmentation revisited. In *Proc. of WWW 2011*, pages 97–106, New York, NY, USA, 2011. ACM.

[13] K. S. Hasan and V. Ng. Conundrums in unsupervised keyphrase extraction: making sense of the state-of-the-art. In *Proc. of COLING 2010: posters*, pages 365–373, Stroudsburg, PA, USA, 2010. ACL.

[14] Y. Huang, L. Sun, and J.-Y. Nie. Query model refinement using word graphs. In *Proc. of CIKM 2010*, pages 1453–1456, New York, NY, USA, 2010. ACM.

[15] S. Huston and W. B. Croft. Evaluating verbose query processing techniques. In *Proc. of SIGIR 2010*, pages 291–298, New York, NY, USA, 2010. ACM.

[16] G. Kumaran and V. R. Carvalho. Reducing long queries using query quality predictors. In *Proc. of SIGIR 2009*, pages 564–571, New York, NY, USA, 2009. ACM.

[17] J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proc. of SIGIR 2001*, pages 111–119, New York, NY, USA, 2001. ACM.

[18] M. Lease, J. Allan, and W. B. Croft. Regression rank: Learning to meet the opportunity of descriptive queries. In *Proc. of ECIR 2009*, pages 90–101, Berlin, Heidelberg, 2009. Springer-Verlag.

[19] C. Lioma and I. Ounis. A syntactically-based query reformulation technique for information retrieval. *Inf. Process. Manage.*, 44:143–162, January 2008.

[20] C. Macdonald and I. Ounis. Global statistics in proximity weighting models. In *Proceedings of Web N-gram 2010 Workshop at SIGIR*, 2010.

[21] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[22] K. Maxwell. *Term selection in information retrieval (forthcoming)*. PhD thesis, University of Edinburgh, 2013.

[23] Q. Mei, D. Zhang, and C. Zhai. A general optimization framework for smoothing language models on graph structures. In *Proc. of SIGIR 2008*, pages 611–618, New York, NY, USA, 2008. ACM.

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

[25] R. Mihalcea and P. Tarau. Textrank: Bringing order into texts. In *Proc. of EMNLP 2004*, pages 404–411, 2004.

[26] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford University, Stanford, CA, 1999.

[27] J. H. Park and W. B. Croft. Query term ranking based on dependency parsing of verbose queries. In *Proc. of SIGIR 2010*, pages 829–830, New York, NY, USA, 2010. ACM.

[28] J. H. Park, W. B. Croft, and D. A. Smith. A quasi-synchronous dependence model for information retrieval. In *Proc. of CIKM 2011*, pages 17–26, New York, NY, USA, 2011. ACM.

[29] J. Rocchio. Relevance feedback in information retrieval. In *The SMART Retrieval System - Experiments in Automatic Document Processing*. Prentice Hall, 1971.

[30] K. Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *J. of Documentation*, 28(1):11–21, 1972.

[31] C. J. van Rijsbergen. Information retrieval: theory and practice. In *Proceedings of the Joint IBM/University of Newcastle upon Tyne Seminar on Data Base Systems*, pages 1–14, 1979.

[32] X. Wan and J. Xiao. Single document keyphrase extraction using neighborhood knowledge. In *Proc. of AAAI 2008*, volume 2, pages 855–860. AAAI Press, 2008.

[33] X. Xue, S. Huston, and W. B. Croft. Improving verbose queries using subset distribution. In *Proc. of CIKM 2010*, pages 1059–1068, New York, NY, USA, 2010. ACM.

[34] C. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proc. of CIKM 2001*, pages 403–410, New York, NY, USA, 2001. ACM.