Parameterized Concept Weighting in Verbose Queries

Michael Bendersky
Dept. of Computer Science
U. of Massachusetts
Amherst, MA
bemike@cs.umass.edu

Donald Metzler
Information Sciences Institute
U. of Southern California
Marina del Rey, CA
metzler@isi.edu

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

ABSTRACT

The majority of the current information retrieval models weight the query concepts (e.g., terms or phrases) in an unsupervised manner, based solely on the collection statistics. In this paper, we go beyond the unsupervised estimation of concept weights, and propose a parameterized concept weighting model. In our model, the weight of each query concept is determined using a parameterized combination of diverse importance features. Unlike the existing supervised ranking methods, our model learns importance weights not only for the explicit query concepts, but also for the latent concepts that are associated with the query through pseudo-relevance feedback. The experimental results on both newswire and web TREC corpora show that our model consistently and significantly outperforms a wide range of state-of-the-art retrieval models. In addition, our model significantly reduces the number of latent concepts used for query expansion compared to the non-parameterized pseudo-relevance feedback based models.

Categories and Subject Descriptors

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

General Terms

Algorithms, Experimentation

Keywords

Parameterized concept weighting, query expansion

1. INTRODUCTION

Term weighting is a classical information retrieval problem that has been studied for decades. However, despite significant advances, a majority of the most widely used information retrieval models, including language modeling [29] and BM25 [30], still weight the importance of query concepts (e.g., terms or phrases) in a simple, unsupervised manner. Such unsupervised estimates of concept importance tend to be based solely on global collection statistics, as evidenced by the standard *inverse document frequency* (IDF) measure of a term's importance.

However, such unsupervised estimates of concept importance have two major shortcomings that can degrade retrieval effectiveness. First, these estimates are rigid and inflexible, due to their dependency on a single global statistic (e.g., IDF). Such dependency does not take into account the influence of a wide range of factors beyond document frequency on the importance of a query concept. Flexible query term weighting is particularly important for verbose queries that often contain a mix of key and complementary concepts [5, 14].

Second, most of the importance weighting research has been applied to single terms (i.e., unigrams). Relatively little research has been done to investigate the appropriateness of IDF and other unsupervised importance weights for multiple-term concepts, including bigrams, phrases, and other proximity expressions. In fact, recent work by Macdonald and Ounis suggests that global statistics do not play as important of a role for such concepts [22]. This research highlights the need for improved understanding of possible alternatives for estimating the importance of multiple-term concepts.

To overcome the shortcomings of unsupervised concept weighting, a number of researchers have recently proposed to incorporate flexible supervised concept importance weighting into the Markov Random Field model [18, 6, 23]. In particular, the weighted sequential dependence model (`WSD`) proposed by Bendersky et al. [6] models the importance weight of the query concepts (including query terms, exact phrases and proximity matches) as a linear combination of document-independent features, such as the number of times the concept occurred as a Wikipedia title, its frequency within a query log, and its global frequency within a large web crawl. The `WSD` model provides a flexible way of estimating importance and can be effectively optimized using existing learning to rank approaches.

The main shortcoming of the `WSD` model [6], and its variants [31, 37], is that the weighting is performed exclusively on the concepts that explicitly occur within the query and disregards the *latent* concepts associated with the information need underlying the query (e.g., the concepts distilled by state-of-the-art query expansion approaches such as relevance model [16] or latent concept expansion [24]). Therefore, the question of how to seamlessly and effectively in-

SIGIR'11, July 24–28, 2011, Beijing, China.

Query Terms	Query Bigrams	Expansion Terms
.1064 patrol	.0257 civil air	.0639 cadet
.1058 civil	.0236 air patrol	.0321 force
.1046 training	.0104 training participants	.0296 aerospace
.0758 participants	.0104 participants receive	.0280 cap

Table 1: Explicit and latent concepts with the highest importance weight for the query "What is the current role of the civil air patrol and what training do participants receive?".

tegrate these latent concepts within a supervised concept weighting model still remains open.

To address this question, in this paper, we propose a novel *parameterized query expansion* model. The proposed model provides an effective alternative to the standard unsupervised weighting for both single terms and multiple-term concepts. In addition, the model generalizes the current supervised concept weighting approaches [6, 18, 31, 35, 37] and provides a *unified* framework for weighting both explicit and latent query concepts.

As an illustrative example of the parameterized query expansion in action, consider the verbose query

> "What is the current role of the civil air patrol and what training do participants receive?"

Table 1 shows the most important explicit query concepts (i.e., the query terms and bigrams) and the most important latent concepts (i.e., the expansion terms) learned by our model. Note that the weights assigned by our model are different from the weights that would be assigned by IDF alone. For instance, while the term *air* has higher IDF than the term *training*, it is deemed less important for the query. In addition, while the term *air* is not important on its own, it is significant in the context of the bigram *air patrol*.

In the case of the query in Table 1, the parameterized query expansion model improves the retrieval effectiveness by 64% over the standard query-likelihood model [29], by 21% over the WSD model [6], and by 8% over the latent concept expansion model [24]. As the evaluation in Sec. 5 demonstrates, these gains in retrieval effectiveness are consistent across queries and collections.

This paper has three primary contributions. First, we describe a novel parameterized query expansion model. Parameterized query expansion provides a flexible framework for modeling the importance of both explicit and latent query concepts. As we show, this framework is a generalization and unification of current state-of-the-art concept weighting [6, 18, 31] and query expansion [24, 15] models. Second, we describe a novel two-stage optimization technique for parameterized query expansion. This technique leverages learning to rank approaches for effective estimation of the explicit and latent concepts importance weights. Finally, we carry out a detailed empirical evaluation that demonstrates the state-of-the-art effectiveness of the proposed model. Our evaluation shows that the approach is particularly beneficial for verbose queries, but is also highly effective for short keyword queries.

The remainder of this paper is laid out as follows. First, Sec. 2 details the process of concept weight parameterization. Then, Sec. 3 describes query expansion with parameterized concept weights. Related work is covered in Sec. 4. Experimental evaluation is provided in Sec. 5. Finally, Sec. 6 concludes the paper and describes directions for future work.

2. PARAMETERIZED CONCEPT WEIGHTING

Given a query Q, we assume that there exists a set of *concepts* related to the underlying information need. In this paper, we use a very broad definition of a concept. A concept is defined as *any syntactic expression that can be matched within a document.*

Clearly, there are a multitude of concept types that can potentially be associated with the information need: individual words, exact phrases, unordered phrases, etc. It is important to note that these concept types can be either explicitly present in the query (e.g., query terms or phrases), or latent (e.g., concepts that are associated with the information need via the process of query expansion).

Formally, we use $\mathcal{T}$ to denote the set of all possible concept types (explicit and latent) associated with the information need underlying the query Q. Then, to assign a score to document D in response to the query Q, we use a linear weighted combination of matches in document D of all concepts types in $\mathcal{T}$ as follows:

$$sc(Q, D) = \sum_{T \in \mathcal{T}} \sum_{\kappa \in T} \lambda_\kappa f(\kappa, D). \qquad (1)$$

This ranking function consists of two components: a concept matching function $f(\kappa, D)$ that computes how related κ is to D, and a concept importance weight λ_κ that indicates the importance of κ. We now discuss how these two components are computed.

2.1 Matching function $f(\kappa, D)$

The concept matching function $f(\kappa, D)$ assigns a score to the matches of concept κ in the document D. The function can take various forms, but in information retrieval applications it is commonly a monotonic function, i.e., its value increases with the number of times concept κ matches document D.

Throughout the remainder of this paper we assume that the matching function $f(\kappa,D)$ takes the form

$$f(\kappa, D) = \log \frac{tf_{\kappa,D} + \mu \frac{tf_{\kappa,\mathcal{C}}}{|\mathcal{C}|}}{|D| + \mu}. \qquad (2)$$

where $tf_{\kappa,D}$ and $tf_{\kappa,\mathcal{C}}$ are the number of concept occurrences in the document and the collection, respectively; μ is a free parameter; $|D|$ is the number of terms in D, and $|\mathcal{C}|$ is the total number of terms in the collection.

The matching function in Eq. 2 is exactly the log of the language modeling estimate for concept κ with Dirichlet smoothing [39]. We use the language modeling estimate as a concept matching function since it is convenient and efficient to compute and exhibits state-of-the-art retrieval performance in other concept-based retrieval models [23, 24]. However, Eq. 1 can also be implemented using other matching functions such as BM25 [30] or DFR [3].

2.2 Importance weight λ_κ

Parameter λ_κ assigns a document-independent importance weight to each concept κ associated with the information need. Based on Eq. 1, the score for document D increases as it matches more of the important concepts associated with the information need. Therefore, correct estimation of the concept importance weight λ_κ is a crucial aspect of the ranking function in Eq. 1. There are several possible approaches for determining concept importance.

One common approach is tying the weights λ_κ of all the concepts of the same type T. This approach is commonly used in the bag-of-words models [29, 3, 30], as well as models that incorporate multiple concept types [23, 28, 10]. Concept weight tying is equivalent to the assumption that all the concepts of the same type are equally important for expressing the query intent. Such an assumption can be potentially detrimental for retrieval performance, especially for complex verbose queries, which combine a large number of concepts of different types [5].

Accordingly, we would like to relax the uniform importance assumption and estimate the weights λ_κ. Clearly, separately estimating a single weight for each concept κ is infeasible, since the number of possible concepts is exponential in the size of the vocabulary.

Instead, we parameterize a concept of type T using a set of importance features Φ^T, which is associated with each concept of the type T. Thus, a concept weight, λ_κ, can be represented as a linear weighted combination of importance features

$$\lambda_\kappa = \sum_{\varphi \in \Phi^T} w_\varphi \varphi(\kappa). \qquad (3)$$

Although a linear form of the importance weighting function is used to simplify parameter estimation (see Sec. 3.3), there is no reason why more complex functional forms could not be used instead.

2.3 Parameterized Ranking Function

Having specified the concept matching function and the parameterized concept weight, we are now ready to derive the final parameterized form of our ranking function. Plugging the parameterized concept weight λ_κ from Eq. 3 into Eq. 1, we get

$$sc(Q, D) = \sum_{T \in \mathcal{T}} \sum_{\varphi \in \Phi^T} w_\varphi \sum_{\kappa \in T} \varphi(\kappa) f(\kappa, D). \qquad (4)$$

From Eq. 4 it is evident that $sc(Q, D)$ is in fact linear in w_φ. This observation simplifies the optimization of Eq. 4, since many current learning to rank techniques [19] can be readily applied to efficiently and effectively optimize a linear ranking function.

Recently, there has been some work on ranking with parameterized concept weights [5, 6, 18, 35]. There are two main limitations in the existing work that we address in this paper.

First, the previous work does not take the holistic view presented in this paper, and optimizes weights for a predetermined concept type. For instance, Bendersky and Croft [5] apply parameterized weights only to the noun phrase concepts, Lease [18] applies it only to the query terms, and Svore et al. [35] only to the query term spans [33].

Second, the aforementioned previous work focuses on the surface form of the query, rather than its underlying information need. Therefore, it is only applicable to the concepts that are explicitly present in the query, and not to the latent concepts that are obtained through query expansion. In contrast, in this paper we propose a novel *parameterized query expansion* model that applies parameterized concept weighting to both the explicit and the latent query concepts.

3. PARAMETERIZED QUERY EXPANSION

In this section, we provide a detailed description of the *parameterized query expansion* retrieval model. This model is an implementation of the general parameterized concept weighting model defined in Sec. 2. It jointly weights several types of both explicit and latent concepts associated with the query. To fully describe the parameterized query expansion model (also referred to as PQE), in Sec. 3.1 we detail the concept types used by the model; in Sec. 3.2 we describe the set of features that determine the concept importance; finally, in Sec. 3.3 we outline the process of optimizing the retrieval effectiveness of the parameterized ranking function.

3.1 Concept Types

In this section, we define the set of concept types $\mathcal{T}$ used by the parameterized query expansion retrieval model. Recall that the choice of $\mathcal{T}$ will determine the structure of the parameterized ranking function in Eq. 4. PQE draws from two sources of evidence for deriving the concepts in $\mathcal{T}$. The first source is the set of words $(q_1, q_2, \ldots, q_{|Q|})$ that appear in the query Q. The second source is the set of latent concepts – concepts that are associated with the query Q through the process of pseudo-relevance feedback. Overall, $\mathcal{T}$ consists of the following concept types.

(1) *QT-concepts.* The query term (QT) concepts are simply the individual query words q_i. This is the most common concept type in information retrieval, which is used both in bag-of-words models [29, 30] and models that incorporate multiple concept types [23, 27, 10].

(2) *PH-concepts.* The phrase (PH) concepts are combinations of query terms that are matched as *exact phrases* in the document. Exact phrase matching has often been used for improving the performance of retrieval methods [9, 38]. Most recently, it has been shown that using query bigrams for exact phrase matching is a simple and efficient method for improving the retrieval performance in large scale web collections [23, 27, 28]. Following this finding, we define the PH-concepts as adjacent query word pairs $(q_i q_{i+1})$.

(3) *PR-concepts.* Similarly to the PH-concepts, the proximity (PR) concepts are defined over adjacent query word pairs $(q_i q_{i+1})$. In order to match the document, both of the individual words in a PR-concept must occur in any order within a *window of fixed length.* In this paper, we fix the window size to 8 words, following some previous work on proximity matching [23, 24, 28].

(4) *ET-concepts.* The expansion (ET) concepts are defined as the top-K terms associated with the query through the process of pseudo-relevance feedback. There is an abundance of literature on query expansion using pseudo-relevance feedback, most recent of which includes, among many others, work by Lavrenko and Croft [16], Tao and Zhai [36], Cao et al. [8], and Lv an Zhai [21]. In this paper we use the *latent concept expansion* (LCE) technique, first proposed by Metzler and Croft [24]. This technique has several important advantages, including state-of-the art performance [24, 15] and the ability to leverage information about arbitrary concepts to improve the quality of query expansion.

To obtain the initial set of ET-concepts using LCE, we first rank documents in the collection using Eq. 4 including only the concept types manifested in the query itself (QT-concepts, PH-concepts and PR-concepts). Then, all the terms in the pseudo-relevant set of documents $\mathcal{R}$ (top ranked

Feature	Description
$\mathtt{GF}(\kappa)$	Frequency of κ in Google n-grams
$\mathtt{WF}(\kappa)$	Frequency of κ in Wikipedia titles
$\mathtt{QF}(\kappa)$	Frequency of κ in a search log
$\mathtt{CF}(\kappa)$	Frequency of κ in the collection
$\mathtt{DF}(\kappa)$	Document frequency of κ in the collection
$\mathtt{AP}(\kappa)$	A priori concept weight

Table 2: Concept importance features Φ^T.

documents) are weighted by

$$w_{LCE}(\kappa) = \sum_{D\in\mathcal{R}} \exp\Big(\gamma_1 sc(Q,D)+\gamma_2 f(\kappa,D)-\gamma_3 \log\frac{tf_{\kappa,\mathcal{C}}}{|\mathcal{C}|}\Big), \tag{5}$$

where γ_i's are free parameters. Finally, K terms with the highest w_{LCE} weights, are added to the set of ET-concepts.

As evident from Eq. 5, w_{LCE} combines three key features to rank a concept: document relevance (manifested by the document score $sc(Q,D)$), score of the concept in the pseudo-relevance set $\mathcal{R}$ (manifested by the matching function $f(\kappa,D)$), and the inverse collection frequency (ICF) of the concept ($-\log\frac{tf_{\kappa,\mathcal{C}}}{|\mathcal{C}|}$). The ICF factor dampens the weights of very common words, thereby improving the quality of the initial set of ET-concepts.

Latent concept expansion can be adopted to include any arbitrary concept type for query expansion. However, in this paper we limit the expansion to individual terms. First, this focus improves the overall efficiency of the PQE model. Second, previous work found no significant benefits when additional types of latent concepts (such as bigrams) were associated with the query in addition to terms alone [24].

3.2 Concept Importance Features

As described in Sec. 2, in order to derive the ranking function $sc(Q,D)$ (Eq. 4), we associate a separate set of importance features Φ^T with each concept type $T = (\mathtt{QT}, \mathtt{PH}, \mathtt{PR}, \mathtt{ET})$. As these features depend only on the concept itself, they can leverage the statistics of the underlying document collection as well as the statistics of external data sources to achieve a potentially more accurate concept weighting.

Following previous work [5, 6, 37], in this paper we use three such external data sources: (i) a large collection of web n-grams, (ii) a sample of a search log, and (iii) Wikipedia. Some of these data sources provide better coverage of terms, while others provide more focused sources of information for determining concept importance. Although there are numerous additional data sources that could be potentially used, we intentionally limit our attention to these three sources as they are available for research purposes, and can be used to reproduce the reported results.

The first source, the Google n-grams corpus[1], contains the frequency counts of English n-grams generated from approximately 1 trillion word tokens. We expect these counts to provide a more accurate frequency estimator, especially for smaller corpora, where some concept frequencies may be underestimated due to the collection size.

In addition, we use a large sample of a search log consisting of approximately 15 million queries[2]. We use this data source to estimate how often a concept occurs in user queries. Intuitively, we assume a positive correlation between an importance of a concept for retrieval and the frequency with which it occurs in queries formulated by the search engine users.

Finally, our third external data source is a snapshot of Wikipedia article titles[3]. Due to the large volume and the high diversity of topics covered by Wikipedia, we assume that important concepts will often appear in its article titles.

For each concept type, we derive five simple frequency features based on these three external sources, as well as the underlying document collection (see Table 2). Note that the parameterized concept weighting in Eq. 3 does not restrict us to this particular set of features, and any additional importance features can be associated with each concept type. However, in this paper, we limit our attention to these five features, since they can be efficiently computed and cached even for large-scale web collections [37], and are suitable for operational retrieval systems.

In addition to the frequency features, Table 2 lists a sixth feature, $\mathtt{AP}(\kappa)$, which is an a priori concept weight (a weight assigned to the concept by default). $\mathtt{AP}(\kappa)$ is set to 1 for query-based concept types $(\mathtt{QT}, \mathtt{PH}, \mathtt{PR})$, and to w_{LCE} (Eq. 5) for the ET-concepts.

The features in Table 2 are computed for each of the four concept types, resulting in 24 features overall. For each such feature φ, we need to estimate the parameter w_φ (see Eq. 4). This estimation process is described in the next section.

3.3 Concept Weight Optimization

To estimate the free parameters w_φ associated with the concept importance features in the ranking function in Eq. 4, we rely on a large and growing body of literature on the learning to rank methods for information retrieval (see Liu [19] for a survey). In this section, we first discuss the coordinate ascent (CA) algorithm [25], which we choose as a base optimization algorithm (Sec. 3.3.1). Then, in Sec. 3.3.2 we discuss the adaption of the CA algorithm for the parameterized query expansion.

3.3.1 Coordinate Ascent

As previously discussed, the ranking function in Eq. 4 is linear w.r.t. w_φ. Therefore, as a base algorithm for optimizing the parameters in Eq. 4 we make use of the coordinate ascent (CA) algorithm proposed by Metzler and Croft [25].

The CA algorithm iteratively optimizes a target metric (in our case, retrieval metric such as MAP) by performing a series of one-dimensional line searches. It repeatedly cycles through each of the parameters w_φ, holding all other parameters fixed while optimizing it. This process is performed iteratively over all parameters until the gain in the target metric is below a certain threshold. Although we use the CA algorithm primarily for its simplicity, efficiency and effectiveness, any other learning to rank approach that estimates the parameters for linear models such as RankSVM [13] or RankNet [7] can be adopted as well.

3.3.2 Optimization with Pseudo-Relevance Feedback

In contrast to most other learning to rank approaches, which usually consider only the concepts that explicitly occur in the query, the parameterized query expansion combines evidence from the query itself and the pseudo-relevance feedback in response to the query. Therefore, the setting of

[1] Available as LDC Catalog # LDC2006T13

[2] Available as a part of Microsoft 2006 RFP dataset.

[3] Available at `http://download.wikimedia.org/enwiki/`

(a) Training Phase

```
(a1) $\mathbf{Q} \leftarrow$ Train queries
(a2) $\mathcal{T} \leftarrow (\mathtt{QT}_{\mathbf{Q}}, \mathtt{PH}_{\mathbf{Q}}, \mathtt{PR}_{\mathbf{Q}})$
(a3) $\mathcal{W}'_{\Phi} \leftarrow \mathtt{CA}(\mathbf{Q}, \mathcal{T})$
(a4) $\mathcal{R}'_{\mathbf{Q}} \leftarrow \mathtt{Rank}(\mathbf{Q}, \mathcal{T}, \mathcal{W}'_{\Phi})$
(a5) $\mathtt{ET}_{\mathbf{Q}} \leftarrow \mathtt{LCE}(\mathbf{Q}, \mathcal{R}'_{\mathbf{Q}})$
(a6) $\mathcal{T} \leftarrow (\mathtt{QT}_{\mathbf{Q}}, \mathtt{PH}_{\mathbf{Q}}, \mathtt{PR}_{\mathbf{Q}}, \mathtt{ET}_{\mathbf{Q}})$
(a7) $\mathcal{W}_{\Phi} \leftarrow \mathtt{CA}(\mathbf{Q}, \mathcal{T})$
```

(b) Testing Phase

```
(b1) $\mathbf{Q} \leftarrow$ Test queries
(b2) $\mathcal{T} \leftarrow (\mathtt{QT}_{\mathbf{Q}}, \mathtt{PH}_{\mathbf{Q}}, \mathtt{PR}_{\mathbf{Q}})$
(b3) $\mathcal{R}'_{\mathbf{Q}} \leftarrow \mathtt{Rank}(\mathbf{Q}, \mathcal{T}, \mathcal{W}_{\Phi})$
(b4) $\mathtt{ET}_{\mathbf{Q}} \leftarrow \mathtt{LCE}(\mathbf{Q}, \mathcal{R}'_{\mathbf{Q}})$
(b5) $\mathcal{T} \leftarrow (\mathtt{QT}_{\mathbf{Q}}, \mathtt{PH}_{\mathbf{Q}}, \mathtt{PR}_{\mathbf{Q}}, \mathtt{ET}_{\mathbf{Q}})$
(b6) $\mathcal{R}_{\mathbf{Q}} \leftarrow \mathtt{Rank}(\mathbf{Q}, \mathcal{T}, \mathcal{W}_{\Phi})$
```

Figure 1: Algorithms for (a) training and (b) testing phases of the parameterized query expansion model.

the importance feature weights associated with the query-based concept types ($\mathtt{QT}, \mathtt{PH}, \mathtt{PR}$), will have a direct effect on the quality of the ET-concepts set, obtained through pseudo-relevance feedback.

To address this challenge, we propose a novel two-stage optimization technique for estimating the set of free parameters $\mathcal{W}_{\Phi}$ in the PQE retrieval model. While simple, this two-stage technique is effective for learning robust weights for both explicit and latent query concepts, as well as improving the quality of the set of ET-concepts.

The algorithm in Fig. 1 provides a schematic overview of this two-stage optimization. At the first stage of the training phase (Fig. 1 (a)), we include only the explicit concept types for optimizing the weights $\mathcal{W}_{\Phi}$ (line *a3*) and ranking with Eq. 4 (line *a4*). This initial ranking is used to obtain a large initial pool of ET-concepts[4] using latent concept expansion, as described in Eq. 5 (line *a5*).

At the second stage of the training phase, we include both explicit and latent concepts for ranking with Eq. 4 (line *a6*). A second round of the CA algorithm is then performed in order to re-estimate the weights $\mathcal{W}_{\Phi}$ for all concept types (line *a7*). To make the optimization process more efficient, at each iteration of the CA algorithm, we use only a small set of the top-K ET-concepts from the initial large pool[5]. At each iteration, the top-K ET-concepts are updated based on the current setting of the $\mathcal{W}_{\Phi}$.

The training phase concludes after the second round of the CA algorithm is completed. At this point, the weights $\mathcal{W}_{\Phi}$ are optimized (in terms of the target retrieval metric) for the training queries. We then use a held-out set of test queries to evaluate the performance of the optimized weights $\mathcal{W}_{\Phi}$ (Fig. 1 (b)).

4. RELATED WORK

Importance weighting of query concepts is one of the key challenges of information retrieval research. However, the majority of commonly used bag-of-words retrieval models (including, among many others, BM25 [30] and language modeling [11, 29, 39]) still use unsupervised term weighting based on global collection statistics, which resembles the term weighting proposed by Luhn [20] in 1958.

Name	# Docs	Topic Numbers
ROBUST04	528,155	301-450, 601-700
WT10g	1,692,096	451-550
GOV2	25,205,179	701-850

Table 3: Summary of TREC collections and topics used for evaluation in Sec. 5.

⟨*title*⟩	dam removal
⟨*desc*⟩	Where have dams been removed and what has been the environmental impact?

Figure 2: An example of ⟨*title*⟩ and ⟨*desc*⟩ portions of a TREC topic #752.

Recently, researchers began investigating techniques for supervised weighting of the terms and concepts in the query [5, 6, 18, 31, 35]. However, these investigations mostly focus only on assigning importance weights to a subset of predetermined concept types. For instance, Lease [18] focuses on term weighting, Bendersky and Croft [5] on noun phrases, and Svore et al. [35] on query term spans. In addition, these supervised techniques take into account only the explicit query concepts and disregard the latent concepts that can be associated with the query via expansion. The parameterized query expansion method proposed in this paper addresses these limitations.

Another field of research which is relevant to this paper is pseudo-relevance feedback. While there is a large number of successful pseudo-relevance feedback based retrieval models (e.g., [8, 16, 24, 21, 38]), most of them employ unsupervised weighting for both explicit and latent concepts. A notable exception is the work by Cao et al. [8] which uses binary classification to determine the importance of the expansion terms. Unlike Cao et al. [8], the proposed parameterized query expansion method takes a more holistic approach, and assigns importance weights to *both* explicit and latent concepts.

5. EVALUATION

This section describes the details of our experimental evaluation. First, in Sec. 5.1, we describe the experimental setup used for the evaluation. Then, in Sec. 5.2, we compare the performance of the parameterized query expansion (PQE) method to the performance of several standard non-parameterized retrieval methods. Further analysis, comparisons and in-depth discussion of the results are provided in Sec. 5.3.

5.1 Experimental Setup

The retrieval experiments described in this paper are implemented using Indri, an open-source search engine [34]. The structured query language implemented in the Indri search engine natively supports multiple concept types, including exact phrases and proximity matches. The Indri query language also supports custom term weighting schemes. As a result, Indri provides a flexible and convenient platform for evaluating the performance of our method.

Table 3 presents a summary of the TREC corpora used in our experiments. The corpora vary both by type (ROBUST04 is a newswire collection, while WT10g and GOV2 are web collections), number of documents, and number of available topics, thereby providing a diverse experimental setup for assessing the robustness of the proposed retrieval method.

During indexing and retrieval, both documents and queries are stemmed using the Porter stemmer. Stopword removal is performed on both documents and queries using the stan-

[4] We set the size of the large pool to 100 concepts.

[5] We limit the size of this small set to 10 concepts.

		Concept Types			
		QT	PH	PR	ET
Non-parameterized methods	QL	$\mathcal{N}$			
	SD	$\mathcal{N}$	$\mathcal{N}$	$\mathcal{N}$	
	RM	$\mathcal{N}$			$\mathcal{N}$
	LCE	$\mathcal{N}$	$\mathcal{N}$	$\mathcal{N}$	$\mathcal{N}$
Parameterized methods	WSD	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$	
	WRM	$\mathcal{P}$			$\mathcal{P}$
	PQE	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$

Table 4: Summary of the evaluated retrieval methods. Each cell indicates whether the weights for the concept type are parameterized according to Eq. 3 ($\mathcal{P}$) or not ($\mathcal{N}$). An empty cell indicates that the concept type is not used by the retrieval method.

dard INQUERY stopword list [2]. The free parameter μ in the concept matching function $f(\kappa,D)$ (see Eq. 2) is set to $2,500$, according to the default Indri configuration of the Dirichlet smoothing parameter.

The optimization of the PQE method is done using 3-fold cross-validation with mean average precision (MAP) as the target metric of the CA algorithm (see Sec. 3.3.2). The statistical significance of differences in the performance of PQE with respect to other retrieval methods is determined using a two-sided Fisher's randomization test with 50,000 permutations and $\alpha < 0.05$.

As was shown in previous work [4, 5, 6, 18], the impact of concept weighting techniques varies significantly across queries of different length. In general, more verbose queries are expected to benefit more from concept weighting, since they are more likely to contain concepts of varying importance. Thus, to test the performance of the proposed methods across multiple query lengths, we treat the ⟨*title*⟩ and the ⟨*desc*⟩ portions of TREC topics as two separate sets of queries in our experiments. The ⟨*title*⟩ and the ⟨*desc*⟩ query convey the same information need for the same topic, but differ in their structure. The ⟨*title*⟩ query is a short keyword query, while the ⟨*desc*⟩ query is a verbose natural language description of the information need. Fig. 2 shows an example of ⟨*title*⟩ and ⟨*desc*⟩ queries for TREC topic #752.

5.2 General Evaluation

Our initial evaluation compares the retrieval performance of the *parameterized query expansion* (PQE) retrieval method (described in Sec. 3) to the performance of several standard baseline methods that do not employ concept weight parameterization.

First, we compare the retrieval performance of the PQE method to the performance of the query likelihood (QL) [29] and sequential dependence model (SD) [23] retrieval methods. These baselines do not perform query expansion, and differ in the choice of the query-based concept types that they use. QL is a standard bag-of-words method. In contrast, the SD method uses, in addition to query terms, both PH-concepts and PR-concepts (which are described in Sec. 3.1). The SD method has consistently demonstrated state-of-the-art retrieval effectiveness in a variety of search tasks, and especially for search over large web collections [23]. Top performing submissions at several TREC tracks have used SD: Terabyte Track 2004-2006 [26], Million Query Track 2007-2008 [1] and Web Track 2009 [32].

Second, we compare the performance of PQE to the performance of two retrieval methods that perform pseudo-relevance feedback (PRF) for query expansion: the RM3 variant of the relevance model (RM) [16] and Latent Concept Expansion (LCE) [24]. Both of these methods are known to improve retrieval performance over methods that do not employ query expansion. Analogously to the QL and SD methods, the RM and LCE methods differ in their choice of the query-based concept types. RM is a bag-of-words model, while LCE uses both PH-concepts and PR-concepts. Both PRF-based methods use individual terms (i.e., unigrams) for query expansion.

Both RM and LCE exhibit highly competitive retrieval performance. Particularly, the LCE method is among the most effective PRF-based methods for large-scale web collections [23, 15]. Lease [17] has recently affirmed its effectiveness at the TREC Relevance Feedback track.

To ensure competitive baseline performance, the free parameters in the PRF-based methods – such as the number of documents used for pseudo-relevance feedback, the query weight in the RM method and the γ parameters in the LCE method (see Eq. 5) – are set using 3-fold cross-validation, analogously to the PQE method. To maintain reasonable efficiency, especially for the large web collection GOV2, we limit the number of expansion terms to 10 for all the PRF-based methods presented in Table 5.

Overall, the four baselines described above differ in their choice of concept types. In contrast to the PQE method, they do not parameterize the concept weights. Table 4 summarizes the choice of concepts and weight parameterization by these methods (as well as two additional methods that will be discussed in Sec. 5.3). For instance, we can see from Table 4 that LCE and PQE share the same concept types, but differ in the parameterization of the concept weights.

5.2.1 Baseline Comparisons

Table 5 compares the retrieval effectiveness of the four baselines to the retrieval effectiveness of PQE, both for ⟨*title*⟩ and ⟨*desc*⟩ queries. Effectiveness is measured using both an early precision metric (prec@20), and the mean average precision of the entire ranked list of 1,000 documents (MAP).

First, it is clear from Table 5 that methods that use multiple concept types (SD, LCE, PQE) are superior to the methods that use terms alone (QL, RM). This result holds for all the collections, for both prec@20 and MAP.

Second, the LCE method, which uses both multiple explicit query concept types and latent expansion concepts, outperforms the SD method, which uses the query concepts alone. This result is consistent with previous work [24], and demonstrates the positive effect of query expansion, even when multiple query concept types are used.

Finally, we compare the proposed method, PQE, to the four baselines. In all 12 comparisons (three collections, two metrics and two query types), our method outperforms all the baselines, in most cases to a statistically significant degree. There are two key elements that contribute to the success of the PQE retrieval method.

First, similarly to LCE, PQE combines multiple explicit concept types with expansion concepts. This combination leads to a very substantial improvement over the standard bag-of-words methods. For instance, for ⟨*desc*⟩ queries on the GOV2 collection, PQE achieves 24% and 17% improvement in MAP over QL and RM, respectively.

⟨*title*⟩	ROBUST04		WT10g		GOV2	
	prec@20	MAP	prec@20	MAP	prec@20	MAP
QL	34.86	24.43	23.99	19.39	50.41	29.56
SD	36.31^{q}	25.90^{q}	23.84	20.63	55.34^{q}	32.24^{q}
RM[10]	36.67^{q}	27.19^{qs}	24.04	19.71	51.15^{s}	30.07
LCE[10]	38.39^{qs}_{r}	28.93^{qs}_{r}	24.95	21.09^{qs}_{r}	55.10^{q}_{r}	33.63^{qs}_{r}
PQE[10]	$\mathbf{39.02}^{qs}_{r}$	$\mathbf{29.16}^{qs}_{r}$ *(+12.6/+0.80)*	$\mathbf{26.31}^{qs}_{rl}$	$\mathbf{21.19}^{qs}_{r}$ *(+2.7/+0.50)*	$\mathbf{56.66}^{q}_{r}$	$\mathbf{34.84}^{qs}_{rl}$ *(+8.1/+3.6)*

⟨*desc*⟩	ROBUST04		WT10g		GOV2	
	prec@20	MAP	prec@20	MAP	prec@20	MAP
QL	33.05	24.22	26.35	18.87	47.62	25.66
SD	34.94^{q}	25.65^{q}	27.45	19.82	50.67^{q}	27.94^{q}
RM[10]	35.04^{q}	25.92^{q}	27.15	20.49^{q}	48.42	27.11^{q}
LCE[10]	37.57^{qs}_{r}	28.10^{qs}_{r}	28.50	21.26	52.52^{qs}_{r}	30.66^{qs}_{r}
PQE[10]	$\mathbf{38.98}^{qs}_{rl}$	$\mathbf{29.40}^{qs}_{rl}$ *(+11.6/+4.6)*	$\mathbf{29.90}^{qs}_{rl}$	$\mathbf{22.17}^{qs}$ *(+11.8/+4.3)*	$\mathbf{54.70}^{qs}_{rl}$	$\mathbf{31.68}^{qs}_{rl}$ *(+13.4/+3.3)*

Table 5: Retrieval effectiveness comparison with all the baselines. Statistically significant differences are marked using the first letter in the title of the retrieval method under comparison. Best result per column is marked by boldface. The numbers in parenthesis indicate improvement over SD and LCE methods, respectively.

Second, unlike all the standard baselines in Table 5, the PQE method applies parameterized concept weighting to both explicit and latent concepts. The parameterized concept weighting leads to significant improvements over all the non-parameterized baselines, including those that use multiple concept types. In most cases, these improvements are statistically significant for both metrics, and they are consistent across different collections.

On average, for ⟨*desc*⟩ queries, there is a 12.2% gain in MAP over SD, and 4.1% gain over LCE. Note that when comparing PQE and LCE methods, the concept weight parameterization is the only factor that contributes to the effectiveness gain, since these two methods share the same concept types (as demonstrated in Table 4).

5.2.2 *Robustness*

In Table 5 we have shown that the PQE method significantly improves the overall performance compared to two state-of-the-art PRF-based methods (RM and LCE). In this section, we analyze the *robustness* of PQE, compared to these two methods. Following previous work [24], we define the robustness of the method as the number of queries improved or hurt (and by how much – in terms of MAP) as the result of the application of the method. A highly robust expansion technique will significantly improve many queries and only minimally hurt a few.

Fig. 3 provides an analysis of the robustness of RM, LCE and PQE for the ⟨*desc*⟩ queries[6]. The histograms in Fig. 3 show, for various ranges of relative decreases/increases in MAP, the number of queries that were hurt/improved with respect to the QL baseline.

Fig. 3 unequivocally demonstrates that PQE is more robust compared to the other two methods. For instance, for the GOV2 collection, PQE improves the performance of 75% of the queries w.r.t. QL, compared to 64% and 68% of the queries improved by RM and LCE respectively. Similar improvements are observed for the other two collections.

In addition, the PQE method is much less likely to significantly hurt the performance, compared to the other two methods. For instance, for the ROBUST04 collection, PQE decreases performance by more than 25% for only 24 (out of 250) queries, compared to 33 and 34 queries with such a decrease for the RM and LCE methods, respectively.

5.2.3 *Graded Relevance Judgments*

In Table 5 and Fig. 3, we have evaluated the retrieval methods using binary relevance judgments — documents are either assumed relevant or non-relevant. However, graded relevance judgments (i.e., categorical judgments with more than two degrees of relevance) are becoming more widely used, especially for evaluating web search tasks. Accordingly, in addition to the binary metrics, we use the normalized discounted cumulative gain (nDCG) metric, which takes into account multiple relevance grades, to evaluate the retrieval effectiveness for the GOV2 collection, which has graded relevance judgments available.

Table 6 shows the nDCG at the top 20 results, as well as the nDCG of the entire ranked list for the PQE method along with the two most effective baselines from Table 5 (SD and LCE). The results in Table 6 are in agreement with the results for the binary metrics in Table 5.

PQE is the most effective among the three methods, in most cases to a statistically significant degree. It is interesting to note that while it is often the case that query expansion methods do not have a significant positive effect on early precision [21, 24], PQE shows significant improvements over the SD method for prec@20 and nDCG@20 metrics for both ⟨*title*⟩ and ⟨*desc*⟩ queries.

5.2.4 ⟨*title*⟩ *and* ⟨*desc*⟩ *Queries*

While most of the previous concept weighting techniques specifically target verbose natural language queries [5, 12, 14, 18], the parameterized concept weighting described in Sec. 2 is more general, and can be applied to any type of query. Table 5 and Table 6 show that PQE outperforms the other methods for both ⟨*title*⟩ and ⟨*desc*⟩ queries.

Intuitively, however, we expect that the parameterized concept weighting will be more beneficial for longer, more complex queries, which may contain more concepts of varying importance. Overall, the results in Table 5 and Table 6

[6] The robustness of these methods for the ⟨*title*⟩ queries is similar, and is omitted due to space constraints.

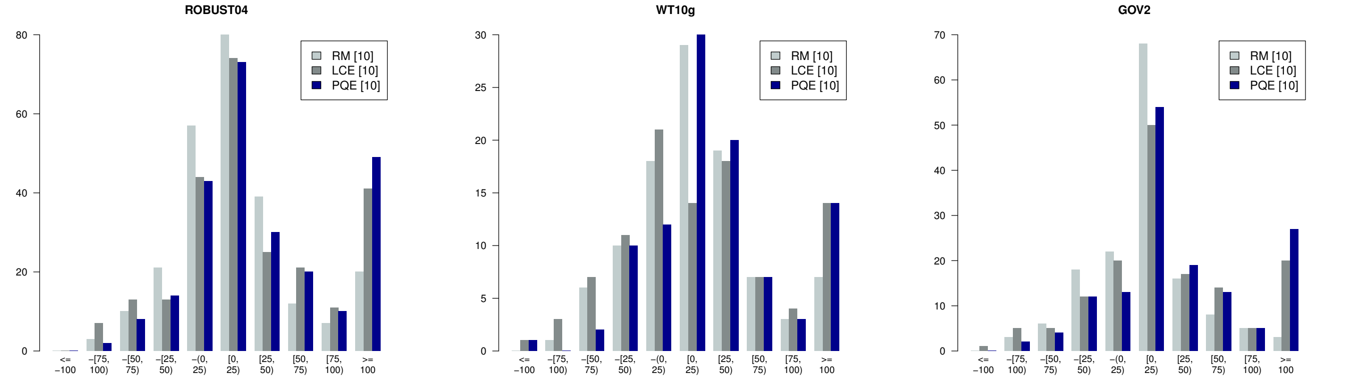

Figure 3: Robustness of RM[10], LCE[10] and PQE[10] methods for the ⟨*desc*⟩ queries w.r.t. the QL method.

⟨*title*⟩	GOV2	
	nDCG@20	nDCG@1000
`SD`	44.54	60.48
`LCE[10]`	43.70	61.29
`PQE[10]`	**45.20**$_l$ *(+1.5/+3.4)*	**62.32**^s_l *(+3.0/+1.7)*

⟨*desc*⟩	GOV2	
	nDCG@20	nDCG@1000
`SD`	41.15	54.45
`LCE[10]`	41.66	56.41
`PQE[10]`	**43.30**^s_l *(+5.2/+3.9)*	**57.60**^s_l *(+5.8/+2.1)*

Table 6: Retrieval effectiveness evaluation (nDCG@k) for the GOV2 collection with graded relevance judgments. Statistically significant differences are marked using the first letter in the title of the retrieval method under comparison. Best result per column is marked by boldface. The numbers in parenthesis indicate improvement over SD and LCE methods, respectively.

confirm this intuition. The gains attained by the parameterized concept weighting for ⟨*desc*⟩ queries are, on average, higher than those attained for ⟨*title*⟩ queries. For instance, the average gain in MAP of the `PQE` method over the `LCE` method is 1.7% for the ⟨*title*⟩ queries, compared to 4.1% for the ⟨*desc*⟩ queries.

It is important to note, however, that the effectiveness gains achieved by the `PQE` method are consistent, and in many cases statistically significant, for both ⟨*title*⟩ and ⟨*desc*⟩ queries. This showcases the applicability of the parameterized concept weighting employed by the `PQE` method for a variety of search scenarios, apart from verbose natural language queries.

5.3 Further analysis

In the remainder of this section, we provide a deeper analysis of the various aspects of the `PQE` method. In Sec. 5.3.1 we compare the performance of `PQE` to the performance of some previously proposed parameterized retrieval methods. In Sec. 5.3.2 we compare the effectiveness of `PQE` with a small number of expansion concepts to the effectiveness of non-parameterized PRF-based methods that use an increasingly large number of expansion concepts. Finally, in Sec. 5.3.3 we compare `PQE` with two recently published methods for query expansion that employ concept weighting.

5.3.1 Parameterized Retrieval Methods

In Sec. 5.2 we have compared the performance of the `PQE` method to the performance of four standard retrieval methods that do not perform any parameterized concept weighting. In this section, we compare `PQE` with two additional retrieval methods, both of which employ parameterized concept weighting. These parameterized retrieval methods differ in their choice of the concept types.

The first method is `WSD`, proposed by Bendersky et al. [6]. This method is a parameterized version of the standard `SD` method. The second method, `WRM` is the parameterized version of the `RM` method. It is conceptually similar to the `PQE` method, however it only uses unigram concepts (`QT` and `ET`-concepts).

Table 4 summarizes the concepts and the parameterization of the `WSD` and `WRM` methods. We compare the effectiveness of these two methods to the effectiveness of the `PQE` method in Table 7.

First, we note that the three parameterized retrieval methods `WSD`, `WRM`, and `PQE` outperform their non-parameterized counterparts (`SD`, `RM` and `LCE`, respectively). In most cases these performance gains are statistically significant. For instance, `WSD` attains a 5.5% gain over `SD`, and `WRM` attains a 10.3% gain over `RM` for the ⟨*desc*⟩ queries. Moreover, analogously to the `PQE` method, these gains, while consistent for all queries, are, on average, larger for the ⟨*desc*⟩ queries.

Second, we note that `PQE` is, overall, the best-performing parameterized retrieval method. The only exception is the WT10g collection, where `WSD` and `PQE` are statistically indistinguishable. On average, `PQE` attains 4.7% gain over `WSD` and 3.5% gain over `WRM` for the ⟨*desc*⟩ queries (and much higher gains for the ROBUST04 and GOV2 collections, specifically).

5.3.2 Number of Expansion Concepts

In Table 5, we have limited ourselves to the efficient setting of using solely the top ten `ET`-concepts for query expansion. In this section, we study the effect of increasing

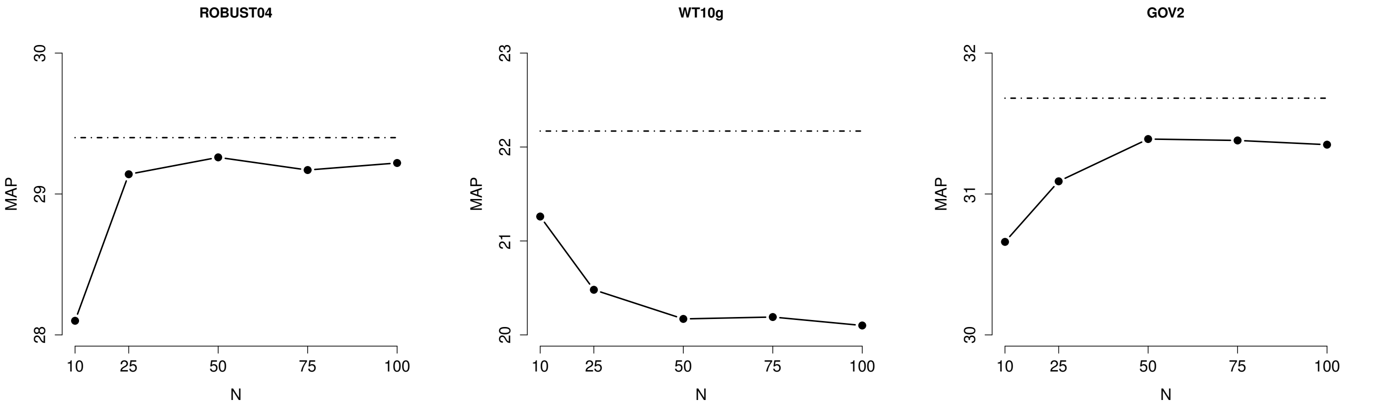

Figure 4: Effect of increasing the number of expansion concepts (N) on the retrieval effectiveness (MAP) of the ⟨*desc*⟩ queries. Solid line — the effectiveness of LCE[N]. Dotted line — the effectiveness of PQE[10].

⟨*title*⟩	ROBUST04	WT10g	GOV2
`WSD`	26.16	**21.48**	33.13
`WRM[10]`	27.35	19.86	31.08
`PQE[10]`	$\mathbf{29.16}^{s}_{r}$	21.19_{r}	$\mathbf{34.84}^{s}_{r}$

⟨*desc*⟩	ROBUST04	WT10g	GOV2
`WSD`	27.49	**22.60**	29.46
`WRM[10]`	27.77	22.46	29.89
`PQE[10]`	$\mathbf{29.40}^{s}_{r}$	22.17	$\mathbf{31.68}^{s}_{r}$

Table 7: Retrieval effectiveness (MAP) of the parameterized retrieval methods. Statistically significant differences are marked using the second letter in the title of the retrieval method under comparison. Best result per column is marked by boldface.

the number of expansion concepts. Particularly, we are interested in addressing the question of whether the addition of expansion concepts in the non-parameterized PRF-based methods (e.g., `LCE`) can compensate for their lack of accurate concept weighting.

Fig. 4 shows the effect of increasing the number of expansion concepts used by the `LCE` retrieval method on its retrieval effectiveness for the ⟨*desc*⟩ queries[7] for all test collections. Fig. 4 demonstrates that adding expansion terms improves the effectiveness of `LCE` in some cases (but worsens it in the case of WT10g). However, adding more `ET`-concepts to `LCE` is still inferior to the fixed setting of using the top ten `ET`-concepts in the `PQE` method, while significantly increasing the query latency.

		Topics	MAP	Source
(a)	`MIX+SOFT-10`	351-400	21.25	*Table 11* [8]
	`PQE[10]`		**21.97**	
(b)	`PRM1`	801-850	33.22	*Table 2* [21]
	`PQE[10]`		**37.41**	

Table 8: Additional comparisons with: (a) Cao et al. [8]; (b) Lv and Zhai [21]. Best result per comparison is marked by boldface.

Fig. 4 demonstrates the importance of parameterized concept weighting for creating both effective and efficient retrieval methods that can scale to large web collections. Compared to the non-parameterized retrieval methods, `PQE` provides a more accurate estimate of concept importance. Moreover, since the features that are used to determine the importance of a concept can be pre-computed and cached (see Sec. 3.2), `PQE` effectively and efficiently filters out the less important expansion concepts and minimizes the query execution time.

5.3.3 Additional Comparisons

In this section, we compare the performance of the `PQE` retrieval method to the performance of two recently proposed query expansion methods that employ concept weighting and proximity information. The first method was proposed by Cao et al. [8], and uses binary classification to weight expansion terms. The second method was proposed by Lv and Zhai [21], and leverages term proximities for expansion term weighting. While less general than the approach proposed here, these two methods also focus on concept weighting, and hence we briefly compare their performance to `PQE`.

For comparison, we use the MAP results reported in the papers by Cao et al. [8] and Lv and Zhai [21], for a subset of topics overlapping with our evaluation. The reported results are for the ⟨*title*⟩ queries only, since these queries are also used in the papers under consideration. Table 8 reports the comparison between the `PQE` method and these two methods. While we cannot draw statistical significance conclusions, since we have no information on individual query performance, we can see from Table 8 that `PQE` is the best performing method in both comparisons.

In all the cases in Table 8 similar query and document processing was applied (Porter stemming, INQUERY stopwords removal, setting of smoothing parameters and number of expansion terms), and similar baselines were reported. Hence, we can confidently attribute the performance gains to the effectiveness of our method, even when compared to other state-of-the-art query expansion methods that use concept weighting and proximity information.

[7] The results for the ⟨*title*⟩ queries are similar, and are omitted due to space constraints.

6. CONCLUSIONS

In this paper, we introduced a novel framework for query expansion with parameterized concept weighting. Parameterized query expansion generalizes and unifies several of the current state-of-the-art concept weighting and query expansion approaches.

Unlike many common retrieval models that use unsupervised concept weighting based on a single global statistic, parameterized query expansion leverages a number of publicly available sources such as Wikipedia and a large collection of web n-grams, to achieve a more accurate concept importance weighting. This importance weighting is applied to both explicit query concepts (terms, exact phrases and proximity matches) as well as latent concepts, which are associated with the query using pseudo-relevance feedback.

An empirical evaluation on newswire and web TREC corpora unequivocally demonstrates the state-of-the-art effectiveness of the parameterized query expansion. Our method consistently outperforms a number of strong baseline methods, which use term dependencies and pseudo-relevance feedback with a larger number of latent concepts. It also achieves significant gains over methods that use parameterized concept weighting, but do not perform query expansion. The highest effectiveness gains are demonstrated for verbose natural language queries, but parameterized query expansion is beneficial for the keyword queries as well.

Overall, our findings demonstrate that the parameterized query expansion is an effective and flexible framework that can seamlessly incorporate multiple concept types. Accordingly, in future work, we intend to introduce additional types of concepts into the parameterized query expansion framework, including multiple-term expansion concepts, named entities, and non-adjacent query term pairs.

7. ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval and in part by ARRA NSF IIS-9014442. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

8. REFERENCES

[1] J. Allan, J. Aslam, B. Carterette, V. Pavlu, and E. Kanoulas. Million Query Track 2008 overview. In *Proc. of TREC*, 2008.

[2] J. Allan, M. E. Connell, W. B. Croft, F. F. Feng, D. Fisher, and X. Li. INQUERY and TREC-9. In *Proc. of TREC-9*, pages 551–562, 2000.

[3] G. Amati and C. J. Van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems (TOIS)*, 20(4):357–389, October 2002.

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

[5] M. Bendersky and W. B. Croft. Discovering key concepts in verbose queries. In *Proc. of SIGIR*, pages 491–498, 2008.

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

[7] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proc. of ICML*, pages 89–96, 2005.

[8] G. Cao, J.-Y. Nie, J. Gao, and S. Robertson. Selecting good expansion terms for pseudo-relevance feedback. In *Proc. of SIGIR*, pages 243–250, 2008.

[9] J. Fagan. Automatic phrase indexing for document retrieval. In *Proc. of SIGIR*, pages 91–101, 1987.

[10] J. Gao, H. Qi, X. Xia, and J.-Y. Nie. Linear discriminant model for information retrieval. In *Proc. of SIGIR*, pages 290–297, 2005.

[11] D. Hiemstra. Term-specific smoothing for the language modeling approach to information retrieval: the importance of a query term. In *In Proc. of SIGIR*, pages 35–41, 2002.

[12] S. Huston and W. B. Croft. Evaluating verbose query processing techniques. In *Proc. of SIGIR*, pages 291–298, 2010.

[13] T. Joachims. Optimizing search engines using clickthrough data. In *Proc. of KDD*, pages 133–142, 2002.

[14] G. Kumaran and V. R. Carvalho. Reducing long queries using query quality predictors. In *Proc. of SIGIR*, pages 564–571, 2009.

[15] H. Lang, D. Metzler, B. Wang, and J.-T. Li. Improved latent concept expansion using hierarchical markov random fields. In *Proc. of CIKM*, pages 249–258, 2010.

[16] V. Lavrenko and W. B. Croft. Relevance Models in Information Retrieval. In W. B. Croft and J. Lafferty, editors, *Language modeling for Information Retrieval*, pages 11–56. Kluwer, 2003.

[17] M. Lease. Incorporating relevance and pseudo-relevance feedback in the markov random field model. In *Proc. of TREC*, 2008.

[18] M. Lease. An improved markov random field model for supporting verbose queries. In *Proc. of SIGIR*, pages 476–483, 2009.

[19] T.-Y. Liu. Learning to Rank for Information Retrieval. *Foundations and Trends in Information Retrieval*, 3(3), 2009.

[20] H. Luhn. The Automatic Creation of Literature Abstracts. *IBM Journal of Research and Development*, 2(2):159, 1958.

[21] Y. Lv and C. Zhai. Positional relevance model for pseudo-relevance feedback. In *Proc. of SIGIR*, pages 579–586, 2010.

[22] C. Macdonald and I. Ounis. Global statistics in proximity weighting models. In *Proc. of the SIGIR Web N-Gram Workshop*, 2010.

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

[24] D. Metzler and W. B. Croft. Latent concept expansion using markov random fields. *Proc. of SIGIR*, pages 311–318, 2007.

[25] D. Metzler and W. B. Croft. Linear feature-based models for information retrieval. *Information Retrieval*, 10(3):257–274, 2007.

[26] D. Metzler, T. Strohman, and W. B. Croft. Indri at TREC 2005: Terabyte track. In *Proc. of TREC*, 2005.

[27] G. Mishne and M. de Rijke. Boosting Web Retrieval Through Query Operations. In *Proc. of ECIR*, pages 502–516, 2005.

[28] J. Peng, C. Macdonald, B. He, V. Plachouras, and I. Ounis. Incorporating term dependency in the DFR framework. In *Proc. of SIGIR*, pages 843–844, 2007.

[29] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proc. of SIGIR*, pages 275–281, 1998.

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

[31] L. Shi and J.-Y. Nie. Using various term dependencies according to their utilities. In *Proc. of CIKM*, pages 1493–1496, 2010.

[32] M. D. Smucker, C. Clarke, and G. V. Cormack. Experiments with ClueWeb09: Relevance feedback and web tracks. In *Proc. of TREC*, 2009.

[33] R. Song, M. Taylor, J.-R. Wen, H.-W. Hon, and Y. Yu. Viewing term proximity from a different perspective. In *Proc. of ECIR*, pages 346–357, 2008.

[34] T. Strohman, D. Metzler, H. Turtle, and W. B. Croft. Indri: A language model-based search engine for complex queries. In *Proc. of IA*, 2004.

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

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

[37] L. Wang, D. Metzler, and J. Lin. Ranking under temporal constraints. In *Proc. of CIKM*, pages 79–88, 2010.

[38] J. Xu and W. B. Croft. Query expansion using local and global document analysis. *Proc. of SIGIR*, pages 4–11, 1996.

[39] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems*, 22(2):179–214, 2004.