Latent Concept Expansion Using Markov Random Fields

Donald Metzler
metzler@cs.umass.edu

W. Bruce Croft
croft@cs.umass.edu

Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts
Amherst, MA 01003

ABSTRACT

Query expansion, in the form of pseudo-relevance feedback or relevance feedback, is a common technique used to improve retrieval effectiveness. Most previous approaches have ignored important issues, such as the role of features and the importance of modeling term dependencies. In this paper, we propose a robust query expansion technique based on the Markov random field model for information retrieval. The technique, called *latent concept expansion*, provides a mechanism for modeling term dependencies during expansion. Furthermore, the use of arbitrary features within the model provides a powerful framework for going beyond simple term occurrence features that are implicitly used by most other expansion techniques. We evaluate our technique against relevance models, a state-of-the-art language modeling query expansion technique. Our model demonstrates consistent and significant improvements in retrieval effectiveness across several TREC data sets. We also describe how our technique can be used to generate meaningful multi-term concepts for tasks such as query suggestion/reformulation.

Categories and Subject Descriptors

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

General Terms

Algorithms, Experimentation, Theory

Keywords

Information retrieval, query expansion, Markov random fields

1. INTRODUCTION

Users of information retrieval systems are required to express complex information needs in terms of Boolean expressions, a short list of keywords, a sentence, a question, or possibly a longer narrative. A great deal of information is lost during the process of translating from the information need to the actual query. For this reason, there has been a strong interest in query expansion techniques. Such techniques are used to augment the original query to produce a representation that better reflects the underlying information need.

Query expansion techniques have been well studied for various models in the past and have shown to significantly improve effectiveness in both the relevance feedback and pseudo-relevance feedback setting [12, 21, 28, 29].

Recently, a Markov random field (MRF) model for information retrieval was proposed that goes beyond the simplistic bag of words assumption that underlies BM25 and the (unigram) language modeling approach to information retrieval [20, 22]. The MRF model generalizes the unigram, bigram, and other various dependence models [14]. Most past term dependence models have failed to show consistent, significant improvements over unigram baselines, with few exceptions [8]. The MRF model, however, has been shown to be highly effective across a number of tasks, including ad hoc retrieval [14, 16], named-page finding [16], and Japanese language web search [6].

Until now, the model has been solely used for ranking documents in response to a given query. In this work, we show how the model can be extended and used for query expansion using a technique that we call *latent concept expansion* (LCE). There are three primary contributions of our work.

First, LCE provides a mechanism for combining term dependence with query expansion. Previous query expansion techniques are based on bag of words models. Therefore, by performing query expansion using the MRF model, we are able to study the dynamics between term dependence and query expansion.

Next, as we will show, the MRF model allows arbitrary features to be used within the model. Query expansion techniques in the past have implicitly only made use of term occurrence features. By using more robust feature sets, it is possible to produce better expansion terms that discriminate between relevant and non-relevant documents better.

Finally, our proposed approach seamlessly provides a mechanism for generating both single and multi-term concepts. Most previous techniques, by default, generate terms independently. There have been several approaches that make use of generalized concepts, however such approaches were somewhat heuristic and done outside of the model [19, 28]. Our approach is both formally motivated and a natural extension of the underlying model.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SIGIR'07, July 23–27, 2007, Amsterdam, The Netherlands.
Copyright 2007 ACM 978-1-59593-597-7/07/0007 ...$5.00.

The remainder of this paper is laid out as follows. In Section 2 we describe related query expansion approaches. Section 3 provides an overview of the MRF model and details our proposed latent concept expansion technique. In Section 4 we evaluate our proposed model and analyze the results. Finally, Section 5 concludes the paper and summarizes the major results.

2. RELATED WORK

One of the classic and most widely used approaches to query expansion is the Rocchio algorithm [21]. Rocchio's approach, which was developed within the vector space model, reweights the original query vector by moving the weights towards the set of relevant or pseudo-relevant documents and away from the non-relevant documents. Unfortunately, it is not possible to formally apply Rocchio's approach to a statistical retrieval model, such as language modeling for information retrieval.

A number of formalized query expansion techniques have been developed for the language modeling framework, including Zhai and Lafferty's model-based feedback and Lavrenko and Croft's relevance models [12, 29]. Both approaches attempt to use pseudo-relevant or relevant documents to estimate a better query model.

Model-based feedback finds the model that best describes the relevant documents while taking a background (noise) model into consideration. This separates the content model from the background model. The content model is then interpolated with the original query model to form the expanded query.

The other technique, relevance models, is more closely related to our work. Therefore, we go into the details of the model. Much like model-based feedback, relevance models estimate an improved query model. The only difference between the two approaches is that relevance models do not explicitly model the relevant or pseudo-relevant documents. Instead, they model a more generalized notion of relevance, as we now show.

Given a query Q, a relevance model is a multinomial distribution, $P(\cdot|Q)$, that encodes the likelihood of each term given the query as evidence. It is computed as:

$$\begin{aligned} P(w|Q) &= \int_D P(w|D)P(D|Q) \\ &\approx \frac{\sum_{D\in\mathcal{R}_Q} P(w|D)P(Q|D)P(D)}{\sum_w \sum_{D\in\mathcal{R}_Q} P(w|D)P(Q|D)P(D)} \end{aligned} \quad (1)$$

where $\mathcal{R}_Q$ is the set of documents that are relevant or pseudo-relevant to query Q. In the pseudo-relevant case, these are the top ranked documents for query Q. Furthermore, it is assumed that $P(D)$ is uniform over this set. These mild assumptions make computing the Bayesian posterior more practical.

After the model is estimated, documents are ranked by clipping the relevance model by choosing the k most likely terms from $P(\cdot|Q)$. This clipped distribution is then interpolated with with the original, maximum likelihood query model [1]. This can be thought of as expanding the original query by k weighted terms. Throughout the remainder of this work, we refer to this instantiation of relevance models as RM3.

There has been relatively little work done in the area of query expansion in the context of dependence models [9]. However, there have been several attempts to expand using multi-term concepts. Xu and Croft's local context analysis (LCA) method combined passage-level retrieval with concept expansion, where concepts were single terms and phrases [28]. Expansion concepts were chosen and weighted using a metric based on co-occurrence statistics. However, it is not clear based on the analysis done how much the phrases helped over the single terms alone.

Papka and Allan investigate using relevance feedback to perform multi-term concept expansion for document routing [19]. The concepts used in their work are more general than those used in LCA, and include InQuery query language structures, such as `#UW50`(white house), which corresponds to the concept "the terms white and house occur, in any order, within 50 terms of each other". Results showed that combining single term and large window multi-term concepts significantly improved effectiveness. However, it is unclear whether the same approach is also effective for *ad hoc* retrieval, due to the differences in the tasks.

3. MODEL

This section details our proposed latent concept expansion technique. As mentioned previously, the technique is an extension of the MRF model for information retrieval [14]. Therefore, we begin by providing an overview of the MRF model and our proposed extensions.

3.1 MRFs for IR

3.1.1 Basics

Markov random fields, which are undirected graphical models, provide a compact, robust way of modeling a joint distribution. Here, we are interested in modeling the joint distribution over a query $Q = q_1, \ldots, q_n$ and a document D. It is assumed the underlying distribution over pairs of documents and queries is a relevance distribution. That is, sampling from the distribution gives pairs of documents and queries, such that the document is relevant to the query.

A MRF is defined by a graph G and a set of non-negative potential functions over the cliques in G. The nodes in the graph represent the random variables and the edges define the independence semantics of the distribution. A MRF satisfies the Markov property, which states that a node is independent of all of its non-neighboring nodes given observed values for its neighbors.

Given a graph G, a set of potentials ψ_i, and a parameter vector Λ, the joint distribution over Q and D is given by:

$$P_{G,\Lambda}(Q,D) = \frac{1}{Z_\Lambda} \prod_{c\in C(G)} \psi(c;\Lambda)$$

where Z is a normalizing constant. We follow common convention and parameterize the potentials as $\psi_i(c;\Lambda) = \exp[\lambda_i f_i(c)]$, where $f_i(c)$ is a real-valued feature function.

3.1.2 Constructing G

Given a query Q, the graph G can be constructed in a number of ways. However, following previous work, we consider three simple variants [14]. These variants are *full independence*, where each query term is independent of each

other given a document, *sequential dependence*, which assumes a dependence exists between adjacent query terms, and *full dependence*, which makes no independence assumptions.

3.1.3 Parameterization

MRFs are commonly parameterized based on the maximal cliques of G. However, such a parameterization is too coarse for our needs. We need a parameterization that allows us to associate feature functions with cliques on a more fine grained level, while keeping the number of features, and thus the number of parameters, reasonable. Therefore, we allow cliques to share feature functions and parameters based on *clique sets*. That is, all of the cliques within a clique set are associated with the same feature function and share a single parameter. This effectively ties together the parameters of the features associated with each set, which significantly reduces the number of parameters while still providing a mechanism for fine-tuning on the level of clique sets.

We propose seven clique sets for use with information retrieval. The first three clique sets consist of cliques that contain one or more query terms and the document node. Features over these cliques should encode how well the terms in the clique configuration describe the document. These sets are:

- T_D – set of cliques containing the document node and exactly one query term.
- O_D – set of cliques containing the document node and two or more query terms that appear in sequential order within the query.
- U_D – set of cliques containing the document node and two or more query terms that appear in any order within the query.

Note that U_D is a superset of O_D. By tying the parameters among the cliques within each set we can control how much influence each type gets. This also avoids the problem of trying to determine how to estimate weights for each clique within the sets. Instead, we now must only estimate a single parameter per set.

Next, we consider cliques that only contain query term nodes. These cliques, which were not considered in [14], are defined in an analogous way to those just defined, except the the cliques are only made up of query term nodes and do not contain the document node. Feature functions over these cliques should capture how compatible query terms are to one another. These clique features may take on the form of language models that impose well-formedness of the terms. Therefore, we define following query-dependent clique sets:

- T_Q – set of cliques containing exactly one query term.
- O_Q – set of cliques containing two or more query terms that appear in sequential order within the query.
- U_Q – set of cliques containing two or more query terms that appear in any order within the query.

Finally, there is the clique that only contains the document node. Features over this node can be used as a type of document prior, encoding document-centric properties. This trivial clique set is then:

- D – clique set containing only the singleton node D

We note that our clique sets form a set cover over the cliques of G, but are not a partition, since some cliques appear in multiple clique sets.

After tying the parameters in our clique sets together and using the exponential potential function form, we end up with the following simplified form of the joint distribution:

$$\log P_{G,\Lambda}(Q,D) =$$
$$\underbrace{\lambda_{T_D} \sum_{c \in T_D} f_{T_D}(c) + \lambda_{O_D} \sum_{c \in O_D} f_{O_D}(c) + \lambda_{U_D} \sum_{c \in U_D} f_{U_D}(c) +}_{F_{DQ}(D,Q) \text{ - document and query dependent}}$$
$$\underbrace{\lambda_{T_Q} \sum_{c \in T_Q} f_{T_Q}(c) + \lambda_{O_Q} \sum_{c \in O_Q} f_{O_Q}(c) + \lambda_{U_Q} \sum_{c \in U_Q} f_{U_Q}(c) +}_{F_Q(Q) \text{ - query dependent}}$$
$$\underbrace{\lambda_D f_D(D)}_{F_D(D) \text{ - document dependent}} \quad - \quad \underbrace{\log Z_\Lambda}_{\text{document + query independent}}$$

where F_{DQ}, F_Q, and F_D are convenience functions defined by the document and query dependent, query dependent, and document dependent components of the joint distribution, respectively. These will be used to simplify and clarify expressions derived throughout the remainder of the paper.

3.1.4 Features

Any arbitrary feature function over clique configurations can be used in the model. The correct choice of features depends largely on the retrieval task and the evaluation metric. Therefore, there is likely not to be a single, universally applicable set of features.

To provide an idea of the range of features that can be used, we now briefly describe possible types of features that could be used. Possible query term dependent features include *tf*, *idf*, named entities, term proximity, and text style to name a few. Many types of document dependent features can be used, as well, including document length, PageRank, readability, and genre, among others.

Since it is not our goal here to find optimal features, we use a simple, fixed set of features that have been shown to be effective in previous work [14]. See Table 1 for a list of features used. These features attempt to capture term occurrence and term proximity. Better feature selection in the future will likely lead to improved effectiveness.

3.1.5 Ranking

Given a query Q, we wish to rank documents in descending order according to $P_{G,\Lambda}(D|Q)$. After dropping document independent expressions from $\log P_{G,\Lambda}(Q,D)$, we derive the following ranking function:

$$P_{G,\Lambda}(D|Q) \stackrel{rank}{=} F_{DQ}(D,Q) + F_D(D) \tag{2}$$

which is a simple weighted linear combination of feature functions that can be computed efficiently for reasonable graphs.

3.1.6 Parameter Estimation

Now that the model has been fully specified, the final step is to estimate the model parameters. Although MRFs are generative models, it is inappropriate to train them using

Feature	Value
$f_{T_D}(q_i, D)$	$\log\left[(1-\alpha)\frac{tf_{q_i,D}}{\|D\|} + \alpha\frac{cf_{q_i}}{\|C\|}\right]$
$f_{O_D}(q_i, q_{i+1}\ldots, q_{i+k}, D)$	$\log\left[(1-\beta)\frac{tf_{\#1(q_i\ldots q_{i+k}),D}}{\|D\|} + \beta\frac{cf_{\#1(q_i\ldots q_{i+k})}}{\|C\|}\right]$
$f_{U_D}(q_i, ..., q_j, D)$	$\log\left[(1-\beta)\frac{tf_{\#uw(q_i\ldots q_j),D}}{\|D\|} + \beta\frac{cf_{\#uw(q_i\ldots q_j)}}{\|C\|}\right]$
$f_{T_Q}(q_i)$	$-\log\left[\frac{cf_{q_i}}{\|C\|}\right]$
$f_{O_Q}(q_i, q_{i+1}\ldots, q_{i+k})$	$-\log\left[\frac{cf_{\#1(q_i\ldots q_{i+k})}}{\|C\|}\right]$
$f_{U_Q}(q_i, ..., q_j)$	$-\log\left[\frac{cf_{\#uw(q_i\ldots q_j)}}{\|C\|}\right]$
f_D	0

Table 1: Feature functions used in Markov random field model. Here, $tf_{w,D}$ is the number of times term w occurs in document D, $tf_{\#\mathbf{1}(q_i\ldots q_{i+k}),D}$ denotes the number of times the exact phrase $q_i\ldots q_{i+k}$ occurs in document D, $tf_{\#\mathbf{uw}(q_i\ldots q_j),D}$ is the number of times the terms $q_i,\ldots q_j$ appear ordered or unordered within a window of N terms, and $|D|$ is the length of document D. The cf and $|C|$ values are analogously defined on the collection level. Finally, α and β are model hyperparameters that control smoothing for single term and phrase features, respectively.

conventional likelihood-based approaches because of *metric divergence* [17]. That is, the maximum likelihood estimate is unlikely to be the estimate that maximizes our evaluation metric. For this reason, we discriminatively train our model to directly maximize the evaluation metric under consideration [14, 15, 25]. Since our parameter space is small, we make use of a simple hill climbing strategy, although other more sophisticated approaches are possible [10].

3.2 Latent Concept Expansion

In this section we describe how this extended MRF model can be used in a novel way to generate single and multi-term concepts that are topically related to some original query. As we will show, the concepts generated using our technique can be used for query expansion or other tasks, such as suggesting alternative query formulations.

We assume that when a user formulates their original query, they have some set of concepts in mind, but are only able to express a small number of them in the form of a query. We treat the concepts that the user has in mind, but did not explicitly express in the query, as latent concepts. These latent concepts can consist of a single term, multiple terms, or some combination of the two. It is, therefore, our goal to recover these latent concepts given some original query.

This can be accomplished within our framework by first expanding the original graph G to include the type of concept we are interested in generating. We call this expanded graph H. In Figure 1, the middle graph provides an example of how to construct an expanded graph that can generate single term concepts. Similarly, the graph on the right illustrates an expanded graph that generates two term concepts. Although these two examples make use of the sequential dependence assumption (i.e. dependencies between adjacent query terms), it is important to note that both the original query and the expansion concepts can use any independence structure.

After H is constructed, we compute $P_{H,\Lambda}(E|Q)$, a probability distribution over latent concepts, according to:

$$P_{H,\Lambda}(E|Q) = \frac{\sum_{D\in\mathcal{R}} P_{H,\Lambda}(Q,E,D)}{\sum_{D\in\mathcal{R}}\sum_{E} P_{H,\Lambda}(Q,E,D)}$$

where $\mathcal{R}$ is the universe of all possible documents and E is some latent concept that may consist of one or more terms. Since it is not practical to compute this summation, we must approximate it. We notice that $P_{H,\Lambda}(Q,E,D)$ is likely to be peaked around those documents D that are highly ranked according to query Q. Therefore, we approximate $P_{H,\Lambda}(E|Q)$ by only summing over a small subset of relevant or pseudo-relevant documents for query Q. This is computed as follows:

$$\begin{aligned} P_{H,\Lambda}(E|Q) &\approx \frac{\sum_{D\in\mathcal{R}_Q} P_{H,\Lambda}(Q,E,D)}{\sum_{D\in\mathcal{R}_Q}\sum_{E} P_{H,\Lambda}(Q,E,D)} \\ &\propto \sum_{D\in\mathcal{R}_Q} \exp\Big[F_{QD}(Q,D) + F_D(D) + F_{QD}(E,D) + F_Q(E)\Big] \end{aligned} \quad (3)$$

where $\mathcal{R}_Q$ is a set of relevant or pseudo-relevant documents for query Q and all clique sets are constructed using H. As we see, the likelihood contribution for each document in $\mathcal{R}_Q$ is a combination of the original query's score for the document (see Equation 2), concept E's score for the document, and E's document-independent score. Therefore, this equation can be interpreted as measuring how well Q and E account for the top ranked documents and the "goodness" of E, independent of the documents. For maximum robustness, we use a different set of parameters for $F_{QD}(Q,D)$ and $F_{QD}(E,D)$, which allows us to weight the term, ordered, and unordered window features differently for the original query and the candidate expansion concept.

3.2.1 Query Expansion

To use this framework for query expansion, we first choose an expansion graph H that encodes the latent concept structure we are interested in expanding the query using. We then select the k latent concepts with the highest likelihood given by Equation 3. A new graph G' is constructed by augmenting the original graph G with the k expansion concepts $E_1,\ldots,E_k$. Finally, documents are ranked according to $P_{G',\Lambda}(D|Q,E_1,\ldots,E_k)$ using Equation 2.

3.2.2 Comparison to Relevance Models

Inspecting Equations 1 and 3 reveals the close connection that exists between LCE and relevance models. Both

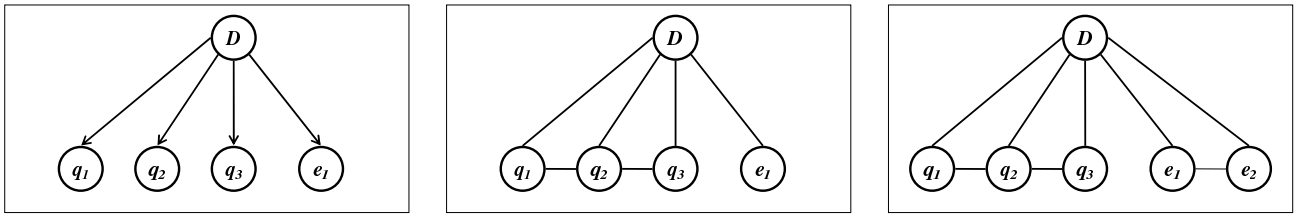

Figure 1: Graphical model representations of relevance modeling (left), latent concept expansion using single term concepts (middle), and latent concept expansion using two term concepts (right) for a three term query.

models essentially compute the likelihood of a term (or concept) in the same manner. It is easy to see that just as the MRF model can be viewed as a generalization of language modeling, so too can LCE be viewed as a generalization of relevance models.

There are important differences between MRFs/LCE and unigram language models/relevance models. See Figure 1 for graphical model representations of both models. Unigram language models and relevance models are based on the multinomial distribution. This distributional assumption locks the model into the bag of words representation and the implicit use of term occurrence features. However, the distribution underlying the MRF model allows us to move beyond both of these assumptions, by modeling both dependencies between query terms and allowing arbitrary features to be explicitly used.

Moving beyond the simplistic bag of words assumption in this way results in a general, robust model and, as we show in the next section, translates into significant improvements in retrieval effectiveness.

4. EXPERIMENTAL RESULTS

In order to better understand the strengths and weaknesses of our technique, we evaluate it on a wide range of data sets. Table 2 provides a summary of the TREC data sets considered. The WSJ, AP, and ROBUST collections are smaller and consist entirely of newswire articles, whereas WT10g and GOV2 are large web collections. For each data set, we split the available topics into a training and test set, where the training set is used solely for parameter estimation and the test set is used for evaluation purposes.

All experiments were carried out using a modified version of Indri, which is part of the Lemur Toolkit [18, 23]. All collections were stopped using a standard list of 418 common terms and stemmed using a Porter stemmer. In all cases, only the title portion of the TREC topics are used to construct queries. We construct G using the sequential dependence assumption for all data sets [14].

4.1 Ad Hoc Retrieval Results

We now investigate how well our model performs in practice in a pseudo-relevance feedback setting. We compare unigram language modeling (with Dirichlet smoothing), the MRF model (without expansion), relevance models, and LCE to better understand how each model performs across the various data sets.

For the unigram language model, the smoothing parameter was trained. For the MRF model, we train the model parameters (i.e. Λ) and model hyperparameters (i.e. α, β). For RM3 and LCE, we also train the number of pseudo-relevant feedback documents used and the number of expansion terms.

Name	Description	# Docs	Train Topics	Test Topics
WSJ	Wall St. Journal 87-92	173,252	51–150	151–200
AP	Assoc. Press 88-90	242,918	51–150	151–200
ROBUST	Robust 2004 data	528,155	301–450	601–700
WT10g	TREC Web collection	1,692,096	451–500	501–550
GOV2	2004 crawl of .gov domain	25,205,179	701–750	751–800

Table 2: Overview of TREC collections and topics.

4.1.1 Expansion with Single Term Concepts

We begin by evaluating how well our model performs when expanding using only single terms. Before we describe and analyze the results, we explicitly state how expansion term likelihoods are computed under this setup (i.e. using the sequential dependence assumption, expanding with single term concepts, and using our feature set). The expansion term likelihoods are computed as follows:

$$
\begin{aligned}
&P_{H,\Lambda}(e|Q) \propto \\
&\sum_{D \in \mathcal{R}_Q} \exp\Big[\lambda_{T_D} \sum_{w \in Q} \log\left[(1-\alpha)\frac{tf_{w,D}}{|D|} + \alpha\frac{cf_w}{|C|}\right] + \\
&\qquad \lambda_{O_D} \sum_{b \in Q} \log\left[(1-\beta)\frac{tf_{\#1(b),D}}{|D|} + \beta\frac{cf_{\#1(b)}}{|C|}\right] + \\
&\qquad \lambda_{U_D} \sum_{b \in Q} \log\left[(1-\beta)\frac{tf_{\#uw(b),D}}{|D|} + \beta\frac{cf_{\#uw(b)}}{|C|}\right] + \\
&\qquad\qquad \log\frac{\left((1-\alpha)\frac{tf_{e,D}}{|D|} + \alpha\frac{cf_e}{|C|}\right)^{\lambda'_{T_D}}}{\left(\frac{cf_e}{|C|}\right)^{\lambda'_{T_Q}}}\Big] \qquad (4)
\end{aligned}
$$

where $b \in Q$ denotes the set of bigrams in Q. This equation clearly shows how LCE differs from relevance models. When we set $\lambda_{T_D} = \lambda'_{T,D} = 1$ and all other parameters to 0, we obtain the exact formula that is used to compute term likelihoods in the relevance modeling framework. Therefore, LCE adds two very important factors to the equation. First, it adds the ordered and unordered window features that are applied to the original query. Second, it applies an intuitive $tf.idf$-like form to the candidate expansion term w. The *idf* factor, which is not present in relevance models, plays an important role in expansion term selection.

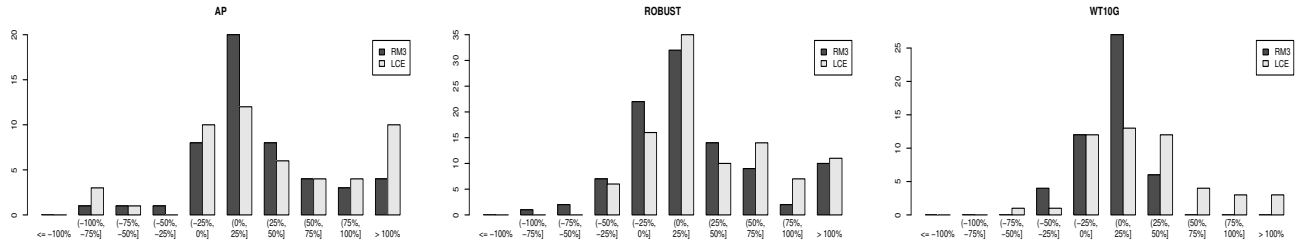

Figure 2: Histograms that demonstrate and compare the robustness of relevance models (RM3) and latent concept expansion (LCE) with respect to the query likelihood model (QL) for the AP, ROBUST, and WT10G data sets.

The results, evaluated using mean average precision, are given in Table 3. As we see, the MRF model, relevance models, and LCE always significantly outperform the unigram language model. In addition, LCE shows significant improvements over relevance models across all data sets. The relative improvements over relevance models is 6.9% for AP, 12.9% for WSJ, 6.5% for ROBUST, 16.7% for WT10G, and 7.3% for GOV2.

Furthermore, LCE shows small, but not significant, improvements over relevance modeling for metrics such as precision at 5, 10, and 20. However, both relevance modeling and LCE show statistically significant improvements in such metrics over the unigram language model.

Another interesting result is that the MRF model is statistically equivalent to relevance models on the two web data sets. In fact, the MRF model outperforms relevance models on the WT10g data set. This reiterates the importance of non-unigram, proximity-based features for content-based web search observed previously [14, 16].

Although our model has more free parameters than relevance models, there is surprisingly little overfitting. Instead, the model exhibits good generalization properties.

4.1.2 Expansion with Multi-Term Concepts

We also investigated expanding using both single and two word concepts. For each query, we expanded using a set of single term concepts and a set of two term concepts. The sets were chosen independently. Unfortunately, only negligible increases in mean average precision were observed.

This result may be due to the fact that strong correlations exist between the single term expansion concepts. We found that the two word concepts chosen often consisted of two highly correlated terms that are also chosen as single term concepts. For example, the two term concept "stock market" was chosen while the single term concepts "stock" and "market" were also chosen. Therefore, many two word concepts are unlikely to increase the discriminative power of the expanded query. This result suggests that concepts should be chosen according to some criteria that also takes novelty, diversity, or term correlations into account.

Another potential issue is the feature set used. Other feature sets may ultimately yield different results, especially if they reduce the correlation among the expansion concepts.

Therefore, our experiments yield no conclusive results with regard to expansion using multi-term concepts. Instead, the results introduce interesting open questions and directions for future exploration.

	LM	MRF	RM3	LCE
WSJ	.3258	$.3425^{\alpha}$	$.3493^{\alpha}$	$.3943^{\alpha\beta\gamma}$
AP	.2077	$.2147^{\alpha}$	$.2518^{\alpha\beta}$	$.2692^{\alpha\beta\gamma}$
ROBUST	.2920	$.3096^{\alpha}$	$.3382^{\alpha\beta}$	$.3601^{\alpha\beta\gamma}$
WT10g	.1861	$.2053^{\alpha}$	$.1944^{\alpha}$	$.2269^{\alpha\beta\gamma}$
GOV2	.3234	$.3520^{\alpha}$	$.3656^{\alpha}$	$.3924^{\alpha\beta\gamma}$

Table 3: Test set mean average precision for language modeling (LM), Markov random field (MRF), relevance models (RM3), and latent concept expansion (LCE). The superscripts α, β, and γ indicate statistically significant improvements ($p < 0.05$) over LM, MRF, and RM3, respectively.

4.2 Robustness

As we have shown, relevance models and latent concept expansion can significantly improve retrieval effectiveness over the baseline query likelihood model. In this section we analyze the robustness of these two methods. Here, we define robustness as the number queries whose effectiveness are improved/hurt (and by how much) as the result of applying these methods. A highly robust expansion technique will significantly improve many queries and only minimally hurt a few.

Figure 2 provides an analysis of the robustness of relevance modeling and latent concept expansion for the AP, ROBUST, and WT10G data sets. The analysis for the two data sets not shown is similar. The histograms provide, for various ranges of relative decreases/increases in mean average precision, the number of queries that were hurt/improved with respect to the query likelihood baseline.

As the results show, LCE exhibits strong robustness for each data set. For AP, relevance models improve 38 queries and hurt 11, whereas LCE improves 35 and hurts 14. Although relevance models improve the effectiveness of 3 more queries than LCE, the relative improvement exhibited by LCE is significantly larger. For the ROBUST data set, relevance models improve 67 queries and hurt 32, and LCE improves 77 and hurts 22. Finally, for the WT10G collection, relevance models improve 32 queries and hurt 16, and LCE improves 35 and hurts 14. As with AP, the amount of improvement exhibited by the LCE versus relevance models is significantly larger for both the ROBUST and WT10G data sets. In addition, when LCE does hurt performance, it is less likely to hurt as much as relevance modeling, which is a desirable property.

1 word concepts	2 word concepts	3 word concepts
telescope	hubble telescope	hubble space telescope
hubble	space telescope	hubble telescope space
space	hubble space	space telescope hubble
mirror	telescope mirror	space telescope NASA
NASA	telescope hubble	hubble telescope astronomy
launch	mirror telescope	NASA hubble space
astronomy	telescope NASA	space telescope mirror
shuttle	telescope space	telescope space NASA
test	hubble mirror	hubble telescope mission
new	NASA hubble	mirror mirror mirror
discovery	telescope astronomy	space telescope launch
time	telescope optical	space telescope discovery
universe	hubble optical	shuttle space telescope
optical	telescope discovery	hubble telescope flaw
light	telescope shuttle	two hubble space

Table 4: Fifteen most likely one, two, and three word concepts constructed using the top 25 documents retrieved for the query *hubble telescope achievements* on the ROBUST collection.

Overall, LCE improves effectiveness for 65%-80% of queries, depending on the data set. When used in combination with a highly accurate query performance prediction system, it may be possible to selectively expand queries and minimize the loss associated with sub-baseline performance.

4.3 Multi-Term Concept Generation

Although we found that expansion using multi-term concepts failed to produce conclusive improvements in effectiveness, there are other potential tasks that these concepts may be useful for, such as query suggestion/reformulation, summarization, and concept mining. For example, for a query suggestion task, the original query could be used to generate a set of latent concepts which correspond to alternative query formulations.

Although evaluating our model on these tasks is beyond the scope of this work, we wish to show an illustrative example of the types of concepts generated using our model. In Table 4, we present the most likely one, two, and three term concepts generated using LCE for the query *hubble telescope achievements* using the top 25 ranked documents from the ROBUST collection.

It is well known that generating multi-term concepts using a unigram-based model produces unsatisfactory results, since it fails to consider term dependencies. This is not the case when generating multi-term concepts using our model. Instead, a majority of the concepts generated are well-formed and meaningful. There are several cases where the concepts are less coherent, such as *mirror mirror mirror*. In this case, the likelihood of the term *mirror* appearing in a pseudo-relevant document outweighs the "language modeling" features (e.g. f_{O_Q}), which causes this non-coherent concept to have a high likelihood. Such examples are in the minority, however.

Not only are the concepts generated well-formed and meaningful, but they are also topically relevant to the original query. As we see, all of the concepts generated are on topic and in some way related to the Hubble telescope. It is interesting to see that the concept *hubble telescope flaw* is one of the most likely three term concepts, given that it is somewhat contradictory to the original query. Despite this contradiction, documents that discuss the telescope flaws are also likely to describe the successes, as well, and therefore this is likely to be a meaningful concept.

One important thing to note is that the concepts LCE generates are of a different nature than those that would be generated using a bigram relevance model. For example, a bigram model would be unlikely to generate the concept *telescope space NASA*, since none of the bigrams that make up the concept have high likelihood. However, since our model is based on a number of different features over various types of cliques, it is more general and robust than a bigram model.

Although we only provided the concepts generated for a single query, we note that the same analysis and conclusions generalize across other data sets, with coherent, topically related concepts being consistently generated using LCE.

4.4 Discussion

Our latent concept expansion technique captures two semi-orthogonal types of dependence. In information retrieval, there has been a long-term interest in understanding the role of term dependence. Out of this research, two broad types of dependencies have been identified.

The first type of dependence is *syntactic dependence*. This type of dependence covers phrases, term proximity, and term co-occurrence [2, 4, 5, 7, 26]. These methods capture the fact that queries implicitly or explicitly impose a certain set of positional dependencies.

The second type is *semantic dependence*. Examples of semantic dependence are relevance feedback, pseudo-relevance feedback, synonyms, and to some extent stemming [3]. These techniques have been explored on both the query and document side. On the query side, this is typically done using some form of query expansion, such as relevance models or LCE. On the document side, this is done as document expansion or document smoothing [11, 13, 24].

Although there may be some overlap between syntactic and semantic dependencies, they are mostly orthogonal. Our model uses both types of dependencies. The use of phrase and proximity features within the model captures syntactic dependencies, whereas LCE captures query-side semantic dependence. This explains why the initial improvement in effectiveness achieved by using the MRF model is not lost

after query expansion. If the same types of dependencies were capture by both syntactic and semantic dependencies, LCE would be expected to perform about equally as well as relevance models. Therefore, by modeling both types of dependencies we see an additive effect, rather than an absorbing effect.

An interesting area of future work is to determine whether or not modeling document-side semantic dependencies can add anything to the model. Previous results that have combined query- and document-side semantic dependencies have shown mixed results [13, 27].

5. CONCLUSIONS

In this paper we proposed a robust query expansion technique called latent concept expansion. The technique was shown to be a natural extension of the Markov random field model for information retrieval and a generalization of relevance models. LCE is novel in that it performs single or multi-term expansion within a framework that allows the modeling of term dependencies and the use of arbitrary features, whereas previous work has been based on the bag of words assumption and term occurrence features.

We showed that the technique can be used to produce high quality, well formed, topically relevant multi-term expansion concepts. The concepts generated can be used in an alternative query suggestion module. We also showed that the model is highly effective. In fact, it achieves significant improvements in mean average precision over relevance models across a selection of TREC data sets. It was also shown the MRF model itself, without any query expansion, outperforms relevance models on large web data sets. This reconfirms previous observations that modeling dependencies via the use of proximity features within the MRF has more of an impact on larger, noisier collections than smaller, well-behaved ones.

Finally, we reiterated the importance of choosing expansion terms that model relevance, rather than the relevant documents and showed how LCE captures both syntactic and query-side semantic dependencies. Future work will look at incorporating document-side dependencies, as well.

Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval, in part by NSF grant #CNS-0454018, in part by ARDA and NSF grant #CCF-0205575, and in part by Microsoft Live Labs. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the sponsor.

6. REFERENCES

[1] N. Abdul-Jaleel, J. Allan, W. B. Croft, F. Diaz, L. Larkey, X. Li, M. D. Smucker, and C. Wade. UMass at TREC 2004: Novelty and HARD. In *Online proceedings of the 2004 Text Retrieval Conf.*, 2004.

[2] C. L. A. Clarke and G. V. Cormack. Shortest-substring retrieval and ranking. *ACM Trans. Inf. Syst.*, 18(1):44–78, 2000.

[3] K. Collins-Thompson and J. Callan. Query expansion using random walk models. In *Proc. 14th Intl. Conf. on Information and Knowledge Management*, pages 704–711, 2005.

[4] W. B. Croft. Boolean queries and term dependencies in probabilistic retrieval models. *Journal of the American Society for Information Science*, 37(4):71–77, 1986.

[5] W. B. Croft, H. Turtle, and D. Lewis. The use of phrases and structured queries in information retrieval. In *Proc. 14th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 32–45, 1991.

[6] K. Eguchi. NTCIR-5 query expansion experiments using term dependence models. In *Proc. of the Fifth NTCIR Workshop Meeting on Evaluation of Information Access Technologies*, pages 494–501, 2005.

[7] J. Fagan. Automatic phrase indexing for document retrieval: An examination of syntactic and non-syntactic methods. In *Proc. tenth Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 91–101, 1987.

[8] J. Gao, J. Nie, G. Wu, and G. Cao. Dependence language model for information retrieval. In *Proc. 27th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 170–177, 2004.

[9] D. Harper and C. J. van Rijsbergen. An evaluation of feedback in document retrieval using co-occurrence data. *Journal of Documentation*, 34(3):189–216, 1978.

[10] T. Joachims. A support vector method for multivariate performance measures. In *Proc. of the International Conf. on Machine Learning*, pages 377–384, 2005.

[11] O. Kurland and L. Lee. Corpus structure, language models, and ad hoc information retrieval. In *Proc. 27th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 194–201, 2004.

[12] V. Lavrenko and W. B. Croft. Relevance-based language models. In *Proc. 24th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 120–127, 2001.

[13] X. Liu and W. B. Croft. Cluster-based retrieval using language models. In *Proc. 27th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 186–193, 2004.

[14] D. Metzler and W. B. Croft. A Markov random field model for term dependencies. In *Proc. 28th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 472–479, 2005.

[15] D. Metzler and W. B. Croft. Linear feature based models for information retrieval. *Information Retrieval*, to appear, 2006.

[16] D. Metzler, T. Strohman, Y. Zhou, and W. B. Croft. Indri at terabyte track 2005. In *Online proceedings of the 2005 Text Retrieval Conf.*, 2005.

[17] W. Morgan, W. Greiff, and J. Henderson. Direct maximization of average precision by hill-climbing with a comparison to a maximum entropy approach. Technical report, MITRE, 2004.

[18] P. Ogilvie and J. P. Callan. Experiments using the lemur toolkit. In *Proc. of the Text REtrieval Conf.*, 2001.

[19] R. Papka and J. Allan. Why bigger windows are better than smaller ones. Technical report, University of Massachusetts, Amherst, 1997.

[20] S. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. In *Online proceedings of the Third Text Retrieval Conf.*, pages 109–126, 1995.

[21] J. J. Rocchio. *Relevance Feedback in Information Retrieval*, pages 313–323. Prentice-Hall, 1971.

[22] F. Song and W. B. Croft. A general language model for information retrieval. In *Proc. eighth international conference on Information and knowledge management (CIKM 99)*, pages 316–321, 1999.

[23] T. Strohman, D. Metzler, H. Turtle, and W. B. Croft. Indri: A language model-based serach engine for complex queries. In *Proc. of the International Conf. on Intelligence Analysis*, 2004.

[24] T. Tao, X. Wang, Q. Mei, and C. Zhai. Language model information retrieval with document expansion. In *Proc. of HLT/NAACL*, pages 407–414, 2006.

[25] B. Taskar, C. Guestrin, and D. Koller. Max-margin markov networks. In *Proc. of Advances in Neural Information Processing Systems (NIPS 2003)*, 2003.

[26] C. J. van Rijsbergen. A theoretical basis for the use of cooccurrence data in information retrieval. *Journal of Documentation*, 33(2):106–119, 1977.

[27] X. Wei and W. B. Croft. LDA-based document models for ad-hoc retrieval. In *Proc. 29th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 178–185, 2006.

[28] J. Xu and W. B. Croft. Improving the effectiveness of information retrieval with local context analysis. *ACM Trans. Inf. Syst.*, 18(1):79–112, 2000.

[29] C. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proc. 10th Intl. Conf. on Information and Knowledge Management*, pages 403–410, 2001.