Towards Concept-Based Translation Models Using Search Logs for Query Expansion

Jianfeng Gao
Microsoft Research
One Microsoft Way
Redmond, WA 98052, USA
jfgao@microsoft.com

Jian-Yun Nie
University of Montreal
CP. 6128, succursale Centre-ville
Montreal, Quebec H3C 3J7, Canada
nie@iro.umontreal.ca

ABSTRACT

Query logs have been successfully used to improve Web search. One of the directions exploits user clickthrough data to extract related terms to a query to perform query expansion (QE). However, term relations have been created between isolated terms without considering their context, giving rise to the problem of term ambiguity. To solve this problem, we propose several ways to place terms in their contexts. On the one hand, contiguous terms can form a phrase; and on the other hand, terms at proximity can provide less strict but useful contextual constraints mutually. Relations extracted between such more constrained groups of terms are expected to be less noisy than those between single terms. In this paper, the constrained groups of terms are called *concepts*. We exploit user query logs to build statistical translation models between concepts, which are then used for QE.

We perform experiments on the Web search task using a real world data set. Results show that the concept-based statistical translation model trained on clickthrough data outperforms significantly other state-of-the-art QE systems.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; I.2.6 [**Artificial Intelligence**]: *Learning*

General Terms

Algorithms, Experimentation

Keywords

Query Expansion, Search Logs, Clickthrough Data, Translation Model, Phrase Model, Concept Model, Web Search

1. INTRODUCTION

One of the fundamental challenges in Web search is term mismatch i.e., a concept is often expressed using different vocabularies and language styles in Web documents and search queries. Query expansion (QE) is an effective strategy to address the issue. It expands a query issued by a user with additional related terms, called *expansion terms*, so that more relevant documents can be retrieved.

CIKM'12, Oct 29 – Nov 2, 2012, Maui Hawaii, USA.

QE is a long-standing research topic in information retrieval. The QE methods based on automatic relevance feedback (e.g., explicit feedback and pseudo relevance feedback or PRF) have been proved to be useful for improving the performance of information retrieval (IR) on TREC datasets [3, 8, 26, 30, 34, 41, 43]. However, these methods cannot be applied directly to a commercial Web search engine because the relevant documents are not always available and generating pseudo-relevant documents requires multi-phase retrieval, which is prohibitively expensive.

Recently, a number of log-based QE methods have been proposed. A typical example is the correlation model proposed by Cui et al. [9, 10], which is considered as the state-of-the-art. Similar to automatic relevance feedback, the method derives expansion terms from (pseudo-)relevant documents, which are not generated by initial retrieval but by previous user clicks extracted from search logs. This enabled them to pre-compute the term correlations offline. Comparing to other QE methods that use pre-computed term correlations either from document collections [24] or human-compiled thesauri [20, 32], the log-based method is superior in that it explicitly captures the correlation between query terms and document terms, and thus can bridge the lexical gap between them more effectively. The log-based method has two other important advantages [10]. First, since search logs retrain query-document pairs clicked by millions of users, the term correlations reflect the preference of the majority of users. Second, the term correlations evolve along with the accumulation of user logs. Hence, the QE process can reflect updated user interests at a specific time. These properties make log-based QE a promising technology of improving the Web search performance of commercial search engines.

However, as pointed out by Riezler et al. [35, 36], Cui et al.'s correlation-based method suffers low precision of QE due to two reasons. First, the correlation model is affected by data sparsity because the estimate is purely based on frequency counting. A possible solution to this problem is to replace the correlation model with a statistical translation model. Second, the method treats queries and documents as bag of words, and does not explicitly capture context information[1]. For example, the word "book" can appear in various contexts with different meanings. Some early work tried to address the problem by identifying phrases according to linguistic criteria [e.g., 24, 41]. Phrases are formed by contiguous words such as "school book" or "hotel booking". While this approach can successfully recognize some phrases, we notice that user queries in Web search do not follow strict linguistic rules.

[1] Cui et al. studied the impact of phrases in [10]. However, the phrases are not used in QE, but as additional indexes.

In many cases, they are simply bags of words without a strict syntactic structure. A typical example is "book paris hotel inexpensive". While the intent seems clear for a human being, applying strict linguistic rules on such a query may result in wrong phrases (e.g. "book paris") that are not useful or even harmful, and miss useful groups of terms (e.g. "hotel inexpensive"). Less strict approaches using bigrams [29] can solve the problem to some extent, but will still miss the useful connections between non-contiguous words such as "book … hotel". The experiments by Shi and Nie [37] showed that legitimate phrases are indeed not always useful for IR (because they do not add more information on top of single terms) while non-legitimate word groups sometimes do (e.g. "book … hotel"). While Shi and Nie considered various connections (dependencies) between query words within documents during the retrieval process, we consider them in the process of extracting term relations, so that the extracted relations are between groups of terms rather than single words.

In this paper we extend the previous log-based QE method of [9, 10] in two directions. First, we formulate QE as the problem of translating a source language of queries into a target language of documents, represented as titles. This allows us to adapt the established modeling techniques developed for statistical machine translation (SMT) to QE. Specifically, we replace the correlation model with a statistical translation model, which is trained on pairs of user queries and the titles of clicked documents using the Expectation Maximization (EM) algorithm [7, 12]. Second, we generalize the word-based translation model to a concept-based model, where the concept is defined as an individual term, a contiguous phrase or a pair of terms in proximity. We represent both queries and titles of documents as bag of concepts rather than bag of words so that term dependencies are explicitly captured and utilized for QE. Both single-term and multi-term concepts can be generated using the same model to expand an original query. Different from previous studies on word-based or phrase-based SMT [7, 25, 31], our translation model also considers less strict concepts formed by terms in proximity, which have proven to be useful for IR.

Another important problem in IR is term weighting, especially on how useful a group of terms is for IR. Shi and Nie used relevance judgments to learn the usefulness of a group of terms. In this paper, we exploit clickthrough data for it. The intuition is that, if many users using the same query clicked on documents in which a form of concept (e.g. phrase) frequently appears, we can assume that this form of concept encodes well the user's search intent, thus is assigned a high weight. In this paper the above intuition is used during the training of translation model in a principled way.

To investigate the relative contribution of translation models and concepts to QE, we have developed a series of translation models that are based on terms, phrases and concepts, respectively. We evaluated these models on the Web search task using a real world data set. Results show that the translation models significantly outperform the correlation model, and that a new QE system using a concept-based translation model achieves the best results, outperforming significantly other state-of-the-art QE methods that are used for comparison in our experiments.

In the rest of this paper, Section 2 reviews related work. Section 3 describes in detail the new log-based QE method that uses statistical translation models and concepts. Section 4 presents experiments. Section 5 concludes the paper.

2. RELATED WORK

Many QE methods have been proposed for IR and Web search: One can find a review in [2]. This section only discusses briefly the QE methods that use search logs and concepts.

2.1 Log-based Methods

There is growing interest in exploiting search logs for QE in Web search [e.g., 9, 15, 21, 35, 40]. Below, we review two log-based QE methods that are closest to ours. Both methods are considered state-of-the-art and have been frequently used for comparison in related studies. The term correlation model of Cui et al. [9, 10] is, to our knowledge, the first to explore query-document relations for direct extraction of expansion terms for Web search. More specifically, the correlation between a query term q and a document term w is computed as the conditional probability defined as

$$P(w|q) = \sum_{D \in \mathbf{D}_q} P(w|D)P(D|q), \quad (1)$$

where $\mathbf{D}_q$ is the set of documents clicked for the queries containing the term q and is collected from search logs, $P(w|D)$ is a normalized *tf-idf* weight of the document term in D, and $P(D|q)$ is the relative occurrence of D among all the documents clicked for the queries containing q. Equation (1) calculates the expansion probability for a single query term. Given a query Q, n expansion terms w with the highest scores are selected and added into the original query to form an expanded query for retrieval. The score of w is calculated as the cohesion weight (CoWeight) with respect to the whole query Q by aggregating the expansion probabilities for each query term $P(w|q)$ as follows:

$$CoWeight(w|Q) = \ln\left(\prod_{q \in Q} P(w|q) + 1\right). \quad (2)$$

Cui et al. showed that the correlation-based QE method outperforms state-of-the-art QE methods that do not use log data, e.g., the local context analysis method (LCA) [41]. In addition, unlike LCA, Cui et al.'s method allows to pre-compute term correlations offline by collecting counts from search logs. However, Riezler et al. [35] argued that the correlation-based method does not explicitly use context information in QE and is susceptible to noise. Riezler et al. thus developed a new log-based QE system by retraining a standard phrase-based SMT system using query-snippet pairs extracted from clickthrough data [35, 36]. Phrases are contiguous words that can be "translated" into other continuous words. The SMT-based QE system can produce cleaner, more relevant expansion terms because rich context information useful for filtering noisy expansions is captured. Furthermore, in the SMT-based system all component models are properly smoothed to alleviate sparse data problems while correlation model relies only on frequency counting. However, in [35], the SMT system is used as a black box in their experiments. Recall that techniques for SMT are developed specifically for translation purposes. Although QE bears a strong resemblance to SMT, it also has important differences due to the fact that related terms or concepts obey less strictly to linguistic rules. While our method also uses the principle of SMT, it allows for flexibilities necessary for IR. We will show that the adapted statistical translation models outperform significantly the correlation model and the black-box SMT system.

Notice that using statistical translation models for IR is not new [e.g., 5, 23, 42]. The effectiveness of the statistical transla-

tion-based approach to Web search has been demonstrated empirically in recent studies where word- and phrase-based translation models are trained on large amounts of clickthrough data [e.g., 14, 15]. The original contribution of our work is that it further extends these recent studies and constructs IR-oriented translation models capturing more flexible dependencies.

2.2 Concept-based Methods

Representing queries and documents as a set of related concepts rather than bag of individual words is a long-standing research topic in IR. The related studies can be grouped into two categories. The first focuses on how to identify concepts in queries and documents. In other words, concept identification is defined as a separate phase from the subsequent QE and document retrieval [e.g., 28, 38]. The second introduces concepts as hidden variables in the QE model as a means to capture term dependencies. Our work belongs to the latter, and is closely related to latent concept expansion (LCE) [30]. LCE is a generalization of the relevance model [26], which we will review first. In relevance models the candidate expansion terms w for a given query Q are ranked according to

$$P(w|Q) = \sum_{D \in \mathbf{D}_Q} P(w|D)P(D|Q), \quad (3)$$

where $\mathbf{D}_Q$ is the set of pseudo-feedback documents retrieved with Q. LCE assumes that a user query encodes a set of latent concepts, which can consist of a single term, multiple terms, or some combination of them. The goal of QE using LCE is to recover these latent concepts given the original query. LCE first constructs a Markov Random Field (MRF) model consisting of the original query Q, the relevant document D, and the expansion concept $\mathbf{e}$, and then picks n expansion concepts with the highest likelihood according to

$$P(\mathbf{e}|Q) \propto \sum_{D \in \mathbf{D}_Q} P(\mathbf{e}, Q, D), \quad (4)$$

where the joint probability $P(\mathbf{e}, Q, D)$, under the MRF framework, is proportion to a weighted combination of a set of potential functions, each of which is defined over a clique in MRF based on a specific term dependence assumption. Experiments on the TREC datasets demonstrate that LCE provides a mechanism to model term dependencies, and achieve superior retrieval results comparing to other state-of-the-art QE methods, including relevance models. Although LCE can generate both single and multi-term concepts, the latter were not found to be useful for QE [30]. A similar result is also reported on LCA [41]: phrase concepts do not help much over single terms. An important reason is that many concepts may have already been well described by single terms for IR purposes. However, from a multi-term phrase or concept, one can determine the related terms more precisely. It may be useful to use multi-term concepts in translation models, which will be confirmed in our experiments.

The latent semantic models, such as LSA [11], PLSA [19] and LDA [6, 39], can also be viewed as QE methods. Instead of expanding the original queries, these methods map different terms occurring in a similar context into the same latent semantic cluster. Thus, a query and a document, represented as vectors in the lower-dimensional semantic space, can still have a high similarity even if they do not share any term. Latent semantic models are also tested in our experiments for comparison.

In this paper we propose a new log-based QE method by combining the strengths of two previous approaches: statistical translation models and LCE. We will show that a statistical translation model based on concepts provides a mechanism to model correlations between query terms and document terms in a principled manner. Unlike reported in the previous studies, we find that both single-term and multi-term concepts generated using our method bring significant improvement in Web search, and combining them yields the best result.

3. MODEL

We view search queries and Web documents as two different languages, and cast QE as a means to bridge the language gap by translating queries to documents, represented by their titles. In this section we will describe three translation models that are based on terms, phrases, and concepts, respectively. We will also describe the way these models are learned from query-title pairs extracted from clickthrough data.

3.1 Word Model

The word model takes the form of IBM Model 1, which is one of the most commonly used lexicon selection models for machine translation [6]. Let $Q = q_1 \dots q_{|Q|}$ be a query and w a candidate expansion term, the translation probability from Q to w is defined as

$$P(w|Q) = \sum_{q \in Q} P(w|q)P(q|Q) \quad (5)$$

where $P(q|Q)$ is the unsmoothed unigram probability of term q in Q. The word translation probabilities $P(w|q)$ are estimated on the query-title pairs derived from the clickthrough data by assuming that the title terms of the clicked (thus possibly relevant) documents are likely to be the desired expansion terms of the query. Our training method follows the standard procedure of training statistical word alignment models proposed by [7]. Formally, we optimize the model parameters θ by maximizing the probability of generating titles from queries over the training data:

$$\theta^* = \text{argmax}_\theta \prod_{i=1}^{N} P(D_i|Q_i, \theta), \quad (6)$$

where both D and Q are viewed as bag of words. The translation probability $P(D|Q, \theta)$ takes the form of IBM Model 1 as

$$P(D|Q, \theta) = \frac{\varepsilon}{(I+1)^J} \prod_{w \in \mathbf{d}} \sum_{q \in \mathbf{q}} P(w|q, \theta). \quad (7)$$

where ε is a constant, J is the length of D, and I is the length of title Q. To find the optimal word translation probabilities of Model 1, we used the EM algorithm, where the number of iterations is determined empirically on held-out data.

3.2 Phrase Model

The word model is context-independent. The phrase model is intended to capture inter-term dependencies for selecting expansion terms. The model $P(w|Q)$ is based on the following generative story where w is generated from Q in three steps:

1. Select a segmentation S of Q according to $P(S|Q)$,
2. Select a query phrase $\mathbf{q}$ according to $P(\mathbf{q}|S)$, and
3. Select a translation (i.e., expansion term) w according to $P(w|\mathbf{q})$.

Summing over all possible segmentations and query phrases, the phrase model is parameterized as

$$P(w|Q) = \sum_{S} \sum_{\mathbf{q} \in S} P(S|Q) P(\mathbf{q}|S) P(w|\mathbf{q}). \quad (8)$$

Here, S is a sequence of phrases. We further assume a uniform probability over all possible segmentations of phrases (contiguous sequences of words). Since for a query of length $|Q|$, there are in total $2^{|Q|-1}$ possible segmentations, we then have $P(S|Q) = \frac{1}{2^{|Q|-1}}$. $P(\mathbf{q}|S)$ is estimated by maximum likelihood estimation (MLE) as

$$P(\mathbf{q}|S) = \frac{C(\mathbf{q}, S)}{|S|}, \quad (9)$$

where $C(\mathbf{q}, S)$ is the count of $\mathbf{q}$ in S, and $|S|$ is the total number of phrases in S (i.e., length of S). $P(w|\mathbf{q})$ in Equation (8) is the translation probability estimated on query-title pairs.

Let $P(\mathbf{q}|Q) = \sum_S P(\mathbf{q}|S) P(S|Q)$, Equation (8) can be rewritten as

$$P(w|Q) = \sum_{\mathbf{q} \in Q} P(w|\mathbf{q}) P(\mathbf{q}|Q), \quad (10)$$

which is of the same form as the word translation model of Equation (5), except on the choice of $\mathbf{q}$. The inter-term dependency information, captured in phrases, is useful for generating more desirable expansion terms. For example, given a query "deal with stuffy nose", expansion terms such as "remedy" and "cold" are more likely to be generated using a phrase model of Equation (10) than using a word model of Equation (5) because although none of the query terms has a high translation probability to generate either of the expansion terms, the phrase translation probabilities, $P(remedy|deal\ with)$ and $P(cold|stuffy\ nose)$, are likely to be high.

In Equation (10), $P(\mathbf{q}|Q)$ is also called the *expected count* of $\mathbf{q}$. Assuming a uniform probability over S, the expected count can be computed easily. For example, the expected count of "stuffy nose" in the query "deal with stuffy nose" is the ratio of the number of S where "stuffy nose" is treated as a phrase and the number of all possible S, i.e. $P(stuffy\ nose|Q) = \frac{2^1}{2^3} = 0.25$ (note that we consider that the stopword "with" is removed).

Now we describe the way $P(w|\mathbf{q})$ is estimated. The method is similar to the one we used to train the word translation model, described in Section 3.1. The only difference is that both queries and titles are represented as vectors. The query vector is a list of $(\mathbf{q}, E(\mathbf{q}|Q))$ pairs, where $\mathbf{q}$ is a query phrase (i.e., any n-gram substring of Q) and $E(\mathbf{q}|Q)$ is the expected count of $\mathbf{q}$ given Q. The title vector is a list of $(w, C(w, D))$ pairs where w is a title term and $C(w, D)$ is the count of w in D. Similar to the case of word translation model, we use the EM algorithm to learn the phrase model.

The phrase model could be improved on several aspects. For example, a better phrase model could be developed by using a high-quality query segmentation system. Instead of assuming a uniform probability over all possible segmentations, we might use the N-best segmentations produced by the system and compute the expected counts more precisely according to the probabilities or scores assigned by the system. However, in our pilot experiments we tried several in-house query segmentation systems, but found none of them outperform significantly the simple model described above. This may be due to the short length and the flexible structure and word order of queries.

3.3 Concept Model

The concept model is a generalization of the word model, where the expansion terms and the query terms are generalized to concepts. The model is of the form

$$P(\mathbf{e}^D|Q) = \sum_{\mathbf{e}^Q \in Q} P(\mathbf{e}^D|\mathbf{e}^Q) P(\mathbf{e}^Q|Q) \quad (11)$$

where $\mathbf{e}^Q$ is a query concept, which can be one of the following three types: a *term concept* which consists of a single query term, a *bigram concept* which is a contiguous term bigram (q_i, q_{i+1}), and a *proximity concept* which consists of an unordered term pair (q_i, q_j) where the two terms occur within a pre-defined window size in Q, and $\mathbf{e}^D$ is a candidate expansion concept, defined similarly as that of $\mathbf{e}^Q$. The above types of concept are found useful in [37].

Now we describe the way the two probability terms in the right hand side of Equation (11) are computed. Similar to the word model, $P(\mathbf{e}^Q|Q)$ is the unsmoothed unigram probability of the concept $\mathbf{e}^Q$ in Q. For example[2], assuming that the window size is the whole query, for a query of length *J,* there are *J* term concepts, $J-1$ bigram concepts, and $J(J-1)$ proximity concepts. Then, the query can have at most $J^2 + J - 1$ unique concepts, in which case $P(\mathbf{e}^Q|Q) = \frac{1}{J^2+J-1}$.

The translation probabilities $P(\mathbf{e}^D|\mathbf{e}^Q)$ are estimated on the query-title pairs derived from the clickthrough data. The method is similar to the one we used to train the phrase model: We represent both queries and titles as vectors. The query vector is a list of $(\mathbf{e}^Q, C(\mathbf{e}^Q, Q))$ pairs, where $\mathbf{e}^Q$ is a query concept and $C(\mathbf{e}^Q|Q)$ is the number of $\mathbf{e}^Q$ that can be derived from Q. The title vector is a list of $(\mathbf{e}^D; C(\mathbf{e}^D, D))$ pairs where $\mathbf{e}^D$ is a title concept and $C(\mathbf{e}^D, D)$ is the count of $\mathbf{e}^D$ in D. Similarly, we performed the EM algorithm.

It is instructive to compare the concept model and the phrase model. On the one hand, the two models are similar in that both try to capture context information by identifying multi-term groups (concepts or phrases) based on the intuition, which states that if many users using the same query click on a set of documents in which a form of multi-term group frequently appears, then this form of multi-term group is likely to encode well the user's search intent. The intuition is incorporated in the EM training of these models. For example, the trained translation models only retain those query concepts or phrases that can translate or map, frequently enough, to a term or concept in titles of the documents which are clicked for the query.

On the other hand, the context model and the phrase model differ in ways the context information is captured. The concept model views a query as a bag of concepts whilst the phrase model views a query as an ordered sequence of words. The EM-based training of phrase model can be viewed as a special case of unsupervised query segmentation, where a contiguous term *n*-gram in a query that is more likely to translate to a title term as a unit than

[2] As pointed out by the reviewers of the paper, for simplicity, we implicitly assume a uniform distribution of different types of concepts. However, the assumption may be suboptimal because, for example, a bigram concept is intuitively more important than an unordered proximity concept. In future work we will investigate the impact of the assumption.

as its individual terms is extracted as a phrase in the trained translation model. Comparing to previous work on query segmentation [e.g., 38], the phrase model tends to segment the queries in a way that good expansion terms can be generated. Long-span term dependencies that are beyond adjacent terms can be captured by setting a maximum length of phrase (at the price of a higher complexity), as we will discuss in Section 4.1. The concept model captures non-local term dependencies using word pairs, in which aspect our model is analogous to the cross-lingual trigger model proposed in [18]. We will show that the concept model can effectively incorporate long-distance dependencies that go beyond local context of phrases without suffering much the data sparseness problem.

Notice that although the concept model and the phrase model would have the same model form if proximity concepts are not allowed in the former and the phrases limited to unigrams and bigrams in the latter, the values of the parameters of the two models would still be different due to the different vector representations they use for EM training. In Section 4 we will demonstrate empirically the impact of these different modeling techniques on QE.

We could come up with a large number of variants of the concept model described in Section 3.3, for example, by defining a new concept consisting of an ordered word pair. The model presented above is the one that achieves the best results in our experiments.

4. EXPERIMENTS

In this study the effectiveness of a QE method is evaluated by issuing a set of queries which are expanded using the method to a search engine and then measuring the Web search performance. Better QE methods are supposed to lead to better Web search results using the correspondingly expanded query set.

Due to the characteristics of our QE methods, we cannot conduct experiments on standard test collections such as the TREC data because they do not contain related user logs we need. Therefore, following previous studies of log-based QE [e.g., 9, 10, 35, 36], we use the proprietary datasets that have been developed for building a commercial search engine, and demonstrate the effectiveness of our methods by comparing them against several state-of-the-art QE methods that are originally developed using TREC data [41, 26, 30]. We also reproduce on our datasets the results of two previous state-of-the-art log-based QE methods [10, 35] and the methods that are based on latent semantic models [14]. Thus, this paper in a sense provides a much-needed comparative study of a set of well-known QE methods that are developed on different settings and whose previously published results are not directly comparable.

Our relevance judgment set consists of 20,000 English queries. On average, each query is associated with 15 Web documents (URLs). Each query-document pair has a relevance label. The label is human generated and is on a 5-level relevance scale, 0 to 4, with 4 meaning document D is the most relevant to query Q and 0 meaning D is not relevant to Q.

The relevance judgment set is constructed as follows. First, the queries are sampled from a year of search engine logs. Adult, spam, and bot queries are all removed. Queries are "de-duped" so that only unique queries remain. To reflex a natural query distribution, we do not try to control the quality of these queries. For example, in our query sets, there are around 20% misspelled queries, and around 20% navigational queries and 10% transactional queries, etc. Second, for each query, we collect Web documents to be judged by issuing the query to several popular search engines (e.g., Google, Bing) and fetching top-10 retrieval results from each. Finally, the query-document pairs are judged by a group of well-trained assessors. In this study all the queries are preprocessed as follows. The text is white-space tokenized and lowercased, numbers are retained, and no stemming/inflection treatment is performed. We split the judgment set into two non-overlapping datasets, namely training and test sets, respectively. Each dataset contain 10,000 queries.

The clicked query-document pairs used for translation model training are extracted from one year query log files using a procedure similar to [16]. In our experiments we use a randomly sampled subset of 100 million pairs that do not overlap with the queries and documents in the training and test sets.

Our Web document collection consists of around 2.5 billion Web pages. In the retrieval experiments we use the index based on the content fields (i.e., body and title text) of each Web page.

The performance of Web search is evaluated by mean *Normalized Discounted Cumulative Gain* (NDCG) [22]. We report NDCG scores at truncation levels 1, 3, and 10. We also performed a significance test, i.e., a t-test with a significance level of 0.05. A significant difference should be read as significant at the 95% level.

4.1 Expansion with Single Terms

We begin by evaluating how well our statistical translation models perform when expanding using only single terms. Before we present experimental results, we describe the way the expansion terms are generated and their term weights are computed using the models described in Section 3.

4.1.1 Systems

Following [10], the log-based QE system used in our experiment takes the following steps to expand an input query Q to a new query Q':

1. Extract all query terms q (eliminating stopwords) from Q.
2. Find all documents that have clicks on a query that contains one or more of these query terms.
3. For each title term w in these documents, calculate its evidence of being selected as an expansion term according to the whole query via a scoring function $Score(w, Q)$.
4. Select n title terms with the highest score and formulate the new query Q' by adding these terms into Q.
5. Use Q' to retrieve Web documents.

The QE methods compared in this section differ in the models used to assign $Score(w, Q)$ in Step (3). For example, using statistical translation models, the score is a translation probability $P(w|Q)$ assigned by the word model, the phrase model, or the concept model.

We also use the translation models to assign the weights of the expansion terms. The weight for an expansion term w is computed as

$$Weight(w; Q) = \max_{q:C(q,w)>0} \frac{P(w|Q)}{P(q|Q)} \quad (12)$$

where $C(q, w)$ is the number of query-title pairs in training data where q occurs in the query part and w occurs in the title part, and $P(q|Q)$ is the translation probability from Q to one of its original

#	QE Models	NDCG@1	NDCG@3	NDCG@10
1	**NoQE**	0.2786	0.3429	0.4193
2	**LCA (PRF)**	0.2801	0.3478	0.4260
3	**TC**	0.2992	0.3626	0.4384
4	**SMT**	0.3003	0.3618	0.4362
5	**S2Net**	0.2993	0.3628	0.4367
6	**WM**	0.3054	0.3672	0.4400
7	**PM_2**	0.3203	0.3797	0.4505
8	**PM_8**	0.3203	0.3806	0.4496
9	**$CM_{T\text{-}B}$**	0.3190	0.3778	0.4473
10	**$CM_{T\text{-}P2}$**	0.3178	0.3760	0.4459
11	**$CM_{T\text{-}B\text{-}P3}$**	0.3214	0.3783	0.4474
12	**$CM_{T\text{-}B\text{-}P5}$**	0.3245	0.3815	0.4500
13	**$CM_{T\text{-}B\text{-}P8}$**	0.3249	0.3817	0.4500
14	**M-$CM_{T\text{-}B\text{-}P8}$**	0.3253	0.3830	0.4500

Table 1: Document ranking results using BM25 with different QE methods.

#	QE Models	NDCG@1	NDCG@3	NDCG@10
1	**NoQE**	0.2803	0.3430	0.4199
2	**RM (PRF)**	0.2842	0.3506	0.4278
3	**TC**	0.2992	0.3621	0.4382
4	**SMT**	0.3002	0.3617	0.4363
5	**BLTM-PR**	0.2983	0.3607	0.4331
6	**WM**	0.3117	0.3717	0.4434
7	**PM_2**	0.3261	0.3832	0.4522
8	**PM_8**	0.3263	0.3836	0.4523
9	**$CM_{T\text{-}B}$**	0.3208	0.3786	0.4472
10	**$CM_{T\text{-}P2}$**	0.3204	0.3771	0.4469
11	**$CM_{T\text{-}B\text{-}P3}$**	0.3219	0.3790	0.4480
12	**$CM_{T\text{-}B\text{-}P5}$**	0.3271	0.3842	0.4534
13	**$CM_{T\text{-}B\text{-}P8}$**	0.3270	0.3844	0.4534
14	**M-$CM_{T\text{-}B\text{-}P8}$**	0.3271	0.3843	0.4533

Table 2: Document ranking results using unigram language model (Jelinek-Mercer smoothing) with different QE methods.

query term q. Notice that $P(q|Q)$ is used as a normalization factor, and we define $Weight(q;Q) = 1$ for all the original query terms q.

We use two document ranking models, BM25 and the unigram language model with Jelinek-Mercer smoothing (LM-JM) [44], to perform Web document retrieval. Notice that in this series of experiments, multi-term phrases or multi-term concepts are only used in QE (i.e., in phrase models and concept models, respectively), but not in retrieval. Therefore, only single terms of the documents are used as indexes. In retrieval, the document ranking models treat both queries and documents as bag of words.

4.1.2 Results

Tables 1 and 2 show Web document ranking results using different QE methods, evaluated on the test set described above.

NoQE (Row 1) is the baseline that uses the raw input queries without expansion. Rows 2 to 5 are the QE methods proposed previously, used in our experiments for comparison. Rows 6 to 14 are the QE methods using different statistical translation models presented in Section 3.

LCA (Row 2 in Table 1), local context analysis [41], is the state-of-the-art PRF methods on the framework of vector space model. In our experiments the number of expansion terms and the number of top-ranked documents used for QE are optimized on the training set, and the smoothing factor δ is set to 0.1, as suggested by [41]. **RM** (Row 2 in Table 2), relevance model [26], is one of the state-of-the-art PRF methods developed for the language modeling framework. Similar to LCA, the number of expansion terms and the number of top-ranked documents used for QE are optimized on the training set.

TC (Row 3) is the QE method based on our implementation of the term correlation model [10]. For each query Q, we added $n = |Q| \times 10$ expansion terms, where $|Q|$ is the length of the query. We see that both TC and PRF methods improve the effectiveness of Web search significantly, and the log-based method outperforms significantly the RPF method that do not use query logs. The results confirm the conclusion of [10].

SMT (Row 4) is a SMT-based QE system. Following Riezler et al. [35], the system is an implementation of a standard phrase-based SMT system with a set of features derived from a translation model and a language model, combined under the log linear model framework [25, 31]. Different from Riezler et al.'s system where the translation model is trained on query-snippet pairs and the language model on queries, in our implementation the translation model is trained on query-title pairs and the language model on titles. To apply the system to QE, expansion terms of a query are taken from those terms in the 10-best translations of the query that have not been seen in the original query string. The results show that the SMT-based QE system is also effective (Row 4 vs. Row 1). However, it does not outperform significantly TC in NDCG at all levels (Row 4 vs. Row 3). This result differs from what is reported in Riezler et al. [35]. A possible reason is that Riezler et al. used the training data of different types (they used query-snippet pairs while we used query-title pairs) and of different sizes (their training data consists of 3 billion pairs while ours only 100 million pairs).

Both **TC** and **SMT**, considered state-of-the-art QE methods, have been frequently used for comparison in related studies. It has been proved in many previous studies that the number of expansion terms sometimes has a significant impact on the QE results. The numbers of expansion terms used for **TC** and **SMT**, described above, are optimized on training data. For a fair comparison, all the proposed translation models (Rows 5 to 11 in Tables 1 and 2) use the same number of expansion terms as **TC** does, i.e., for a query Q, we added $n = |Q| \times 10$ expansion terms. We will come back to this problem in Section 4.1.3.

The latent semantic models can also be viewed as QE methods. Instead of expanding the original queries, they deal with term mismatch by mapping terms into latent semantic clusters. Gao et al. [14] compared a number of latent semantic models trained on clickthrough data on the task of Web search. In our experiments we compare our QE methods with the best latent semantic models reported in [14]. For a fair comparison, these latent semantic models are trained using the same clickthrough data described above.

S2Net (Row 5 in Table 1) [14] is a linear projection model that maps a sparse, high-dimensional term vector into a dense, low-dimensional space through a simple matrix multiplication. **S2Net** is a significant extension to latent semantic analysis (LSA) [11] in that the projection matrix is discriminatively learned using pairs of queries and relevant/irrelevant titles, extracted from clickthrough data, in such a way that the learned model assigns higher

similarity scores to relevant titles compared to irrelevant ones for the same query. In our experiments we ranked documents based on a weighted linear combination of two BM25 scores, computed respectively in the original term space and in the projected semantic space.

BLTM-PR (Row 5 in Table 2) [14] is the bilingual topic model with posterior regularization. **BLTM-PR** is an extension to probabilistic LSA [19] and latent Dirichlet allocation (LDA) [6] that views a query and its paired titles as having a shared topic distribution. In our experiments, MAP inference was used to learn the model parameters. We also used posterior regularization [13] to constrain the paired query and title to have similar fractions of terms assigned to each topic. **BLTM-PR** is a generative model, and can be incorporated naturally into the language modeling framework of document ranking. Given a learned topic-word distribution $\boldsymbol{\phi}_z$, we fold in unseen documents to learn their document-topic distributions $\boldsymbol{\theta}_D$. Then, for a given query Q, **BLTM-PR** ranks the documents as:

$$P(Q|D) = \prod_{q \in Q} \sum_{z} P(q|\boldsymbol{\phi}_z) P(z|\boldsymbol{\theta}_D)$$

In our experiments we linearly interpolated **BLTM-PR** and LM-JM at the term level for document ranking.

Our results confirm the effectiveness of the two latent semantic models. Their performances are significantly better than the baseline models without QE (Row 5 vs. Row 1), and are statistically on par with the baseline log-based QE methods which are considered state-of-the-art (Row 5 to Rows 3 and 4). Although the latent sematic models cannot beat the improved log-based QE methods in Rows 6 to 13, it is interesting to investigate how to best combine latent semantic models and QE, assuming that the two approaches use different strategies to bridge the lexical gap between search queries and Web documents, and thus are complementary. We leave it to future work.

WM (Row 6) uses the word model described in Section 3.1 for QE. The models used in **WM** and **TC** (Row 3) are context-independent and differ mainly in training methods. For the sake of comparison, in our experiment the word model is EM-trained with the term correlation model as initial point. Riezler hypothesize that translation model is superior to correlation model because the EM training captures the hidden alignment information when mapping document terms to query terms, leading to a better smoothed probability distribution. We observe that **WM** outperforms **TC** significantly as shown in Row 6 vs. Row 3. This result confirms Riezler's hypothesis.

PM$_n$ (Rows 7 and 8) are the QE systems using phrase models, where the subscript n specifies the maximum length of query phrase that is allowed. On the one hand, we see that incorporating context information is useful for QE. The phrase models outperform significantly the word model (Rows 7 and 8 vs. Row 6). On the other hand, although we set the maximum phrase length to 8 (Row 8), less than 5% of the translation pairs contain phrase(s) longer than 3 in the resulting phrase model, which is pruned to a manageable size by dropping translation pairs whose translation probabilities are lower than a pre-set threshold. As a result, simply using longer phrases does not always lead to significant improvement due to the data sparseness problem (Row 8 vs. Row 7).

CM$_f$ (Rows 9 to 13) are the QE systems using different concept models, where the subscript f specifies the concept types that define the model. For example, f = **T-B-P3** in Row 11 indicates that the model consists of term concepts (**T**), bigram concepts (**B**), and proximity concepts consisting of term pairs within a window of size 3 (**P3**). The results suggest several conclusions. First, we see that **PM**$_2$ significantly outperforms **CM**$_{\text{T-B}}$, demonstrating that using expected counts (as in **PM**$_2$) for EM training is more effective than using raw counts (as in **CM**$_{\text{T-B}}$). Second, word order is useful information for generating better expansion terms. This is demonstrated by the fact that although **CM**$_{\text{T-P2}}$ and **CM**$_{\text{T-B}}$ cover the term dependencies within the same span, the latter, in which word order is captured via bigram features, significantly outperforms the former. Third, long-distance context that goes beyond local context of phrases is still useful to improve the effectiveness of QE. The use of proximity concepts based on word pairs turns out to be a simple and robust way of capturing such information without suffering the data sparseness problem. Unlike **PM**, the improvement of using word pairs within larger window sizes is visible until the window size is increased to 8 (Rows 9 to 13). Although there is no significant difference between **CM**$_{\text{T-B-P5}}$ and **CM**$_{\text{T-B-P8}}$ in Table 2, using BM25 the latter still brings small but significant improvement in NDCG@1, as shown in Rows 12 and 13 in Table 1.

M-CM$_{\text{T-B-P8}}$ (Row 14) uses the same model as **CM**$_{\text{T-B-P8}}$, except that the former can generate multi-term concepts. As described earlier, BM25 and LM-JM treat both queries and documents as bag of words. Thus, although multi-term concepts are added to an original query, they are broken into terms in retrieval. As a result, we did not observe any significant difference by using multi-term concepts (Row 14 vs. Row 13), except for NDCG@3 in Table 1. To take advantage of QE using multi-term concepts, it is desirable to treat both queries and documents as bag of concepts rather than bag of words. This motivates us to use MRF as document ranking model for the comparison experiments, which we will describe in detail in Section 4.2.

In conclusion, the results up to now show that the best QE system is the one using a concept-based translation model, which, trained on query-title pairs via EM, provides a principled mechanism to capture local and global term dependencies and thus effectively bridge the lexical gap between queries and Web documents.

4.1.3 Impact of Weighting Expansion Terms

Two typical questions that need to be answered when developing a QE system are (1) how to determine the best number of expansion terms and (2) how to weight these expansion terms, with respect to the original query terms. The experiments presented in this section will show that there is one answer to both questions.

Recall that the statistical translation models select expansion terms w of a query Q by ranking all candidates via $P(w|Q)$. We thus use this probability to weight each selected expansion term, as Equation (12). Figure 1 plots the value of NDCG@1 achieved by the system using LM-JM and a QE method based on word model, as a function of the number of expansion terms. The solid line is the result using the term weighting function of Equation (12) and the dot line is the result of the un-weighted system where the weights of all expansion terms are set to 1. The results show that the QE system with term weighting not only achieves better NDCG scores but is much more robust. That is, its performance is not as sensitive to the number of expansion terms as the un-weighted system does. Term weighting is particularly useful when many expansion terms are added. This is in agreement with our intuition because term weights, computed by Equation (12), tend to penalize the lower-ranked, noisy expansion terms, avoiding possible query drift.

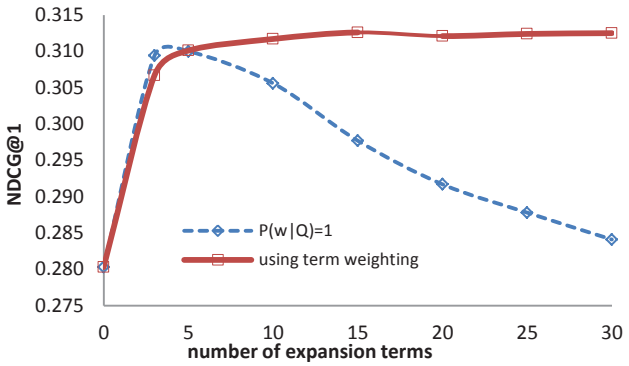

Figure 1: NDCG@1 using LM-JM and a word model based QE, as a function of the number of expansion terms.

We repeated the experiment using phrase models and concept models, and observed very similar results.

4.2 Expansion with Multi-Term Concepts

This section discusses QE results using concept-based models where a query is expanded by both single-term concepts and multi-term concepts (i.e., bigram concepts and proximity concepts). We begin our discussion with a description of the document ranking model that is based on the Markov Random Field (MRF) framework under which multi-term concepts can be used for document retrieval.

4.2.1 Systems

The MRF approach to IR [29] models the joint distribution of $P_\Lambda(Q,D)$ over a set of query term random variables $Q = q_1 \dots q_{|Q|}$ and a document random variable D. It is constructed from a graph G consisting of a document node and nodes for each query term. Nodes in the graph represent random variables and edges define the dependence semantics between the variables. An MRF satisfies the Markov property, which states that a node is independent of all of its non-neighboring nodes given observed values of its neighbors, defined by the clique configurations of G. The joint distribution over the random variables in G is defined as

$$P_\Lambda(Q,D) = \frac{1}{Z_\Lambda} \prod_{c \in C(G)} \varphi(c;\Lambda) \tag{13}$$

where $C(G)$ is the set of cliques in G, and each $\varphi(c;\Lambda)$ is a non-negative potential function defined over a clique configuration c that measures the *compatibility* of the configuration, Λ is a set of parameters that are used within the potential function, and Z_Λ normalizes the distribution. For document ranking, we can drop Z_Λ and simply rank each document D by its unnormalized joint probability with Q under the MRF. It is common to define MRF potential functions of the exponential form as $\varphi(c;\Lambda) = \exp(\lambda_c f(c))$, where $f(c)$ is a real-valued feature function over clique values and λ_c is the weight of the feature function. Then, we can compute the posterior $P_\Lambda(D|Q)$ as

$$\begin{aligned} P_\Lambda(D|Q) = \frac{P_\Lambda(Q,D)}{P_\Lambda(Q)} &\overset{rank}{\Longrightarrow} \sum_{c \in C(G)} \log \varphi(c;\Lambda) \\ &= \sum_{c \in C(G)} \lambda_c f(c), \end{aligned} \tag{14}$$

which is essentially a weighted linear combination of a set of feature functions, identical to the linear discriminative model for IR [17]. In our experiments, we used three feature functions, each for one concept type defined in Section 3.3: individual terms, contiguous bigram phrases, and proximity. They are defined as

$$f_T(q,D) = \log\left[(1-\alpha_T)\frac{tf(q,D)}{|D|} + \alpha_T \frac{cf(q,C)}{|C|}\right], \tag{15}$$

$$\begin{aligned} f_B\big((q_i,q_{i+1}),D\big) = \log\Bigg[&(1-\alpha_B)\frac{tf((q_i,q_{i+1}),D)}{|D|} \\ &+ \alpha_B \frac{cf((q_i,q_{i+1}),C)}{|C|}\Bigg], \text{and} \end{aligned} \tag{16}$$

$$\begin{aligned} f_{P8}\big((q_i,q_{i+1}),D\big) = \log\Bigg[&(1 \\ &- \alpha_{P8})\frac{tf_{\#uw8}\big((q_i,q_{i+1}),D\big)}{|D|} \\ &+ \alpha_{P8}\frac{cf_{\#uw8}\big((q_i,q_{i+1}),D\big)}{|C|}\Bigg]. \end{aligned} \tag{17}$$

In Equations (15) to (16), $|D|$ and $|C|$ indicate respective token counts of the document and the entire collection, α's are interpolation weights whose values are empirically tuned via cross-validation, and $tf(.)$ are document frequencies for different types of concepts: $tf(q,D)$ is the number of times that q matches in D, $tf((q_i,q_{i+1}),D)$ is the number of times that the bigram phrase (q_i,q_{i+1}) matches in D, and $tf_{\#uw8}\big((q_i,q_{i+1}),D\big)$ is the number of times that both terms q_i and q_{i+1} occur within a window of 8 positions in D. The collection frequencies $cf(.)$ are defined analogously. These feature functions have been successfully used by other researchers [4, 27, 30]. α's are smoothing parameters. Following [17], the weights Λ are optimized for NDCG using the Powell Search algorithm [33].

Similar to the QE system that generates single-term expansions as described in Section 4.1.1, the QE system based on concept model, representing both queries and documents as concept vectors, takes the following steps to expand a query:

1. Extract all query terms q (eliminating stopwords) from Q.
2. Find all documents that have clicks on a query that contains one or more of these query terms.
3. For each title concept in these documents $\mathbf{e}^D$ calculate its evidence of being selected as an expansion concept according to the whole query via the concept model $P(\mathbf{e}^D|Q)$ defined in Equation (11).
4. Select $n = |Q| \times 10$ title concepts with the highest probabilities and formulate the new query Q' by adding these concepts into Q.
5. Use Q' to retrieve documents using the MRF model.

We also use the concept-based translation model to assign the weights of the expansion concepts as

$$Weight(\mathbf{e}^D;Q) = \max_{q:C(q,\mathbf{e}^D)>0} \frac{P(\mathbf{e}^D|Q)}{P(q|Q)} \tag{18}$$

where $P(q|Q)$ is the translation probability from Q to one of its original query term q assigned by the concept model of Equation (11), $C(q,\mathbf{e}^D)$ is the number of query-title pairs in training data where q occurs in the query part and the concept $\mathbf{e}^D$ occurs in the title part. We define $Weight(q;Q) = 1$ for all the original query terms q.

#	QE Models	NDCG@1	NDCG@3	NDCG@10
1	**MRF (NoQE)**	0.2802	0.3434	0.4201
2	**LCE (PRF)**	0.2848	0.3509	0.4280
3	**1 + M-CM$_{T\text{-}B\text{-}P8}$**	0.3293	0.3869	0.4548
4	**MRF (Term Feature)**	0.2803	0.3430	0.4199
5	**MRF (Bigram Feature)**	0.2478	0.3124	0.3950
6	**MRF (Proximity Feature)**	0.2536	0.3178	0.3992
7	**4 + M-CM$_{T\text{-}B\text{-}P8}$**	0.3270	0.3844	0.4534
8	**5 + M-CM$_{T\text{-}B\text{-}P8}$**	0.3144	0.3730	0.4434
9	**6 + M-CM$_{T\text{-}B\text{-}P8}$**	0.3216	0.3770	0.4471

Table 3: Document ranking results using MRF with different QE methods using multi-term concepts.

4.2.2 Results

Table 3 shows Web document ranking results using MRF and different QE methods, evaluated on the test set described earlier. **MRF** (Row 1) is the baseline that uses raw input queries without QE. **LCE** (Row 2) is latent concept expansion, the state-of-the-art PRF method developed for the MRF framework [30]. It is easy to see that just as the MRF can be viewed as a generalization of unigram language modeling (Row 1 in Table 2), so can LCE be viewed as a generalization of RM (Row 2 in Table 2). The generalization is due to the explicit modeling of term dependencies. Unfortunately, in our experiments the generalization does not lead to any significant improvement in retrieval results (Rows 1 and 2 in Table 2 vs. Rows 1 and 2 in Table 3). On the other hand, the QE method that uses the concept-based translation model trained on clickthrough data leads to significant improvement (Row 3 vs. Rows 1 and 2). A comparison of this result with **M-CM$_{T\text{-}B\text{-}P8}$** in Table 2 confirms that it is advantageous to bind the word-term expansion terms together during search rather than breaking them into bags of words. To better understand our results, we perform the following additional analysis.

Since the MRF model of Equations (14) to (17) is essentially a linear combination of three feature functions, we can easily investigate the effectiveness of individual features and their combination for document ranking. Rows 4 to 6 in Table 3 are the results of three MRF models, each of which uses an individual feature. Row 4 uses the term concept and is identical to LM-JM (Row 1 of Table 2). Results in Rows 5 and 6 are significantly worse than that in Row 4, showing that the language gap between documents and queries is substantially bigger when only multi-term phrases or word pairs are used as indexing units because these multi-term units encode language difference not only in word distribution but also in language structure. The result is consistent with that reported in [21]. Due to the bigger language discrepancy using multi-term features, the MRF model combining all three feature functions is not significantly better than the term-based model (Row 4 vs. Row 1). Our result is different from the previous results reported in [27, 29]. The possible reason is that these previous studies test MRF on TREC collections, where the language difference between queries and documents is much smaller than that on our datasets.

In Rows 7 to 9, we try to bridge the language gap using QE based on concept model. Notice that for different MRF models, we only expanded the original queries with the concepts whose type is consistent with that of the features defined in the corresponding MRF model. That is, the expansion concepts are only terms in Row 7, bigram phrases in Row 8, and proximity concepts in Row 9. Results show that the QE method using the concept-based translation model significantly improves the document ranking results for all three feature functions. In particular, the improvement on the multi-term feature functions (Row 8 vs. Row 5 and Row 9 vs. Row 6) is much bigger than that on the single-term feature function (Row 7 vs. Row 4). The results demonstrate that the proposed QE method using the concept-based translation model trained on clickthrough data is an effective means to fully utilize term dependence to improve Web document retrieval results.

5. CONCLUSIONS

This paper combines two techniques to improve the log-based QE method based on term correlations proposed by Cui et al. [9, 10] for Web search. First, we replace the term correlation model estimated purely on raw term frequencies with statistical translation models. We select expansion terms for a query according to how likely it is that the expansion terms occur in the titles of documents that are relevant to the query. Assuming that a query is parallel to the titles of documents clicked for that query, the translation models are trained on query-title pairs, extracted from clickthrough data, using the EM algorithm. We showed in our experiments that a term-based statistical translation model, trained using the EM algorithm, outperforms significantly the correlation model. Second, we extend the term-based model that is context-independent to context-dependent models that incorporate term dependence information useful for generating more precious expansions. We confirmed empirically that a phrase-based model, which captures adjacent term dependencies, significantly improves the performance of word-based QE. We demonstrated that a concept-based model, which incorporates term, bigram and proximity features, is able to fusion local and global context in such an effective way that achieves the best QE performance in our experiments. We also showed that the new, improved log-based QE system outperforms significantly other state-of-the-art QE systems, including the one based on SMT [35, 36] and ones using latent semantic models [14].

In future work, we intend to explore strategies of combining log-based QE methods and latent semantic models for Web search. For example, we might learn topics on multi-term concepts rather than on single terms. We will also explore more sophisticated methods of identifying phrases and concepts for building the phrase model and the concept model, respectively. Another interesting area is to apply the similar models on the document side. That is, instead of expanding queries, we expand a document by adding terms or concepts that are likely to occur in the relevant search queries. Since both log-based QE and document expansion can be performed offline, each of them, or some combination of the two, might provide a promising approach to cope with term mismatch for commercial search engines.

REFERENCES

[1] Agichtein, E., Brill, E., and Dumais, S. 2006. Improving web search ranking by incorporating user behavior information. In *SIGIR*, pp. 19-26.

[2] Baeze-Yates, R., and Ribeiro-Neto, B. 2011. *Modern Information Retrieval.* Addison-Wesley.

[3] Bai, J., Song, D., Bruza, P., Nie, J-Y., and Cao, G. 2005. Query expansion using term relationships in language models for information retrieval. In *CIKM*, pp. 688-695.

[4] Bendersky, M., Metzler, D., and Croft, B. 2010. Learning concept importance using a weighted dependence model. In *WSDM*, pp. 31-40.

[5] Berger, A., and Lafferty, J. 1999. Information retrieval as statistical translation. In *SIGIR*, pp. 222-229.

[6] Blei, D. M., Ng, A. Y., and Jordan, M. J. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3: 993-1022.

[7] Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., and Mercer, R. L. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19(2): 263-311.

[8] Cao, G., Nie, J-Y., Gao, J., and Robertson, S. 2008. Selecting good expansion terms for pseudo-relevance feedback. In *SIGIR*, pp. 289-305.

[9] Cui, H., Wen, J-R., Nie, J-Y. and Ma, W-Y. 2002. Probabilistic query expansion using query logs. In *WWW*, pp. 325-332.

[10] Cui, H., Wen, J-R., Nie, J-Y. and Ma, W-Y. 2003. Query expansion by mining user log. *IEEE Trans on Knowledge and Data Engineering.* Vol. 15, No. 4. pp. 1-11.

[11] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T., and Harshman, R. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6): 391-407

[12] Dempster, A., Laird, N., and Rubin, D. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39: 1-38.

[13] Ganchev, K., Graca, J., Gillenwater, J., and Taskar, B. 2010. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research,* 11 (2010): 2001-2049.

[14] Gao, J., Toutanova, K., Yih., W-T. 2011. Clickthrough-based latent semantic models for web search. In *SIGIR*, pp. 675-684.

[15] Gao, J., He, X., and Nie, J-Y. 2010. Clickthrough-based translation models for web search: from word models to phrase models. In *CIKM,* pp. 1139-1148.

[16] Gao, J., Yuan, W., Li, X., Deng, K., and Nie, J-Y. 2009. Smoothing clickthrough data for web search ranking. In *SIGIR*, pp. 355-362.

[17] Gao, J., Qi, H., Xia, X., and Nie, J-Y. 2005. Linear discriminant model for information retrieval. In *SIGIR*, pp. 290-297.

[18] Hasan, S., Ganitkevitch, J., Ney, H., and Andres-Fnerre, J. 2008. Triplet lexicon models for statistical machine translation. In *EMNLP*, pp. 372-381.

[19] Hofmann, T. 1999. Probabilistic latent semantic indexing. In *SIGIR*, pp. 50-57.

[20] Hovy, E., Gerber, L., Hermjakob, U., Junk, M., and Lin, C-Y. 2000. Question answering in webclopedia. In *TREC 9*.

[21] Huang, J., Gao, J., Miao, J., Li, X., Wang, K., and Behr, F. 2010. Exploring web scale language models for search query processing. In *WWW,* pp. 451-460.

[22] Jarvelin, K. and Kekalainen, J. 2000. IR evaluation methods for retrieving highly relevant documents. In *SIGIR*, pp. 41-48

[23] Jin, R., Hauptmann, A. G., and Zhai, C. 2002. Title language model for information retrieval. In *SIGIR*, pp. 42-48.

[24] Jing, Y., and Croft., B. 1994. An association thesaurus for information retrieval. In *RIAO*, pp. 146-160.

[25] Koehn, P., Och, F., and Marcu, D. 2003. Statistical phrase-based translation. In *HLT/NAACL*, pp. 127-133.

[26] Lavrenko, V., and Croft, B. 2001. Relevance-based language models. In *SIGIR*, pp. 120-128.

[27] Lease, M. 2009. An improved markov random field model for supporting verbose queries. In *SIGIR*, pp. 476-483

[28] Li, Y., Hsu, P., Zhai, C., and Wang, K. 2011. Unsupervised query segmentation using clickthrough for information retrieval. In *SIGIR*, pp. 285-294.

[29] Metzler, D., and Croft, B. 2005. A markov random field model for term dependencies. In *SIGIR*, pp. 472-479.

[30] Metzler, D., and Croft, B. 2007. Latent concept expansion using markov random fields. In *SIGIR*, pp. 311-318.

[31] Och, F. 2002. *Statistical machine translation: from single-word models to alignment templates.* PhD thesis, RWTH Aachen.

[32] Prager, J., Chu-Carroll, J., and Czuba, K. 2001. Use of Wordnet hypernyms for answering what is questions. In *TREC 10*.

[33] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. 1992. *Numerical Recipes in C*. Cambridge Univ. Press.

[34] Rocchio, J. 1971. Relevance feedback in information retrieval. In *The SMART retrieval system: experiments in automatic document processing*, pp. 313-323, Prentice-Hall Inc.

[35] Riezler, S., Liu, Y. and Vasserman, A. 2008. Translating queries into snippets for improving query expansion. In *COLING 2008*. 737-744.

[36] Riezler, S., and Liu, Y. 2010. Query rewriting using monolingual statistical machine translation. *Computational Linguistics*, 36(3): 569-582.

[37] Shi, L., and Nie, J-Y. 2010. Modeling variable dependencies between characters in Chinese information retrieval. In *AAIRS*, pp. 539-551.

[38] Tan, B. and Peng, F. 2008. Unsupervised query segmentation using generative language models and wikipedia. In *WWW,* pp. 347-356.

[39] Wei, X., and Croft, W. B. 2006. LDA-based document models for ad-hoc retrieval. In *SIGIR*, pp. 178-185.

[40] Wen, J., Nie, J-Y., and Zhang, H. 2002. Query clustering using user logs. *ACM TOIS*, 20(1): 59-81.

[41] Xu, J., and Croft, B. 1996. Query expansion using local and global document analysis. In *SIGIR*.

[42] Xue, X., Jeon, J., Croft, W. B. 2008. Retrieval models for Question and answer archives. In *SIGIR*, pp. 475-482.

[43] Zhai, C., and Lafferty, J. 2001a. Model-based feedback in the kl-divergence retrieval model. In *CIKM*, pp. 403-410.

[44] Zhai, C., and Lafferty, J. 2001b. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR*, pp. 334-342.