Learning Lexicon Models from Search Logs for Query Expansion

Jianfeng Gao
Microsoft Research, Redmond
Washington 98052, USA
jfgao@microsoft.com

Shasha Xie
Educational Testing Service, Princeton
New Jersey 08540, USA
sxie@ets.org

Xiaodong He
Microsoft Research, Redmond
Washington 98052, USA
xiaohe@microsoft.com

Alnur Ali
Microsoft Bing, Bellevue
Washington 98004, USA
alnurali@microsoft.com

Abstract

This paper explores log-based query expansion (QE) models for Web search. Three lexicon models are proposed to bridge the lexical gap between Web documents and user queries. These models are trained on pairs of user queries and titles of clicked documents. Evaluations on a real world data set show that the lexicon models, integrated into a ranker-based QE system, not only significantly improve the document retrieval performance but also outperform two state-of-the-art log-based QE methods.

1 Introduction

Term mismatch is a fundamental problem in Web search, where queries and documents are composed using different vocabularies and language styles. Query expansion (QE) is an effective strategy to address the problem. It expands a query issued by a user with additional related terms, called *expansion terms*, so that more relevant documents can be retrieved.

In this paper we explore the use of clickthrough data and translation models for QE. We select expansion terms for a query according to how likely it is that the expansion terms occur in the title of a document that is relevant to the query. Assuming that a query is parallel to the titles of documents clicked for that query (Gao et al. 2010a), three lexicon models are trained on query-title pairs extracted from clickthrough data. The first is a word model that learns the translation probability between single words. The second model uses lexicalized triplets to incorporate word dependencies for translation. The third is a bilingual topic model, which represents a query as a distribution of hidden topics and learns the translation between a query and a title term at the semantic level. We will show that the word model provides a rich set of expansion candidates while the triplet and topic models can effectively select good expansion terms, and that a ranker-based QE system which incorporates all three of these models not only significantly improves Web search result but outperforms other log-based QE methods that are state-of-the-art.

There is growing interest in applying user logs to improve QE. A recent survey is due to Baeze-Yates and Ribeiro-Neto (2011). Below, we briefly discuss two log-based QE methods that are closest to ours and are re-implemented in this study for comparison. Both systems use the same type of log data that we used to train the lexicon models. The term correlation model of Cui et al. (2002; 2003) is to our knowledge the first to explore query-document relations for direct extraction of expansion terms for Web search. The method outperforms traditional QE methods that do not use log data e.g. the local analysis model of Xu and Croft (1996). In addition, as pointed out by Cui et al. (2003) there are three important advantages that make log-based QE a promising technology to improve the performance of commercial search engines. First, unlike traditional QE methods that are based on relevance feedback, log-based QE derives expansion terms from search logs, allowing term correlations to be pre-computed offline. Compared to methods that are based on thesauri either compiled manually (Prager et al. 2001) or derived au-

tomatically from document collections (Jing and Croft 1994), 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. Second, since search logs retrain query-document pairs clicked by millions of users, the term correlations reflect the preference of the majority of users. Third, the term correlations evolve along with the accumulation of user logs, thus can reflect updated user interests at a specific time.

However, as pointed out by Riezler et al. (2008), Cui et al.'s correlation-based method suffers low precision of QE partly because the correlation model does not explicitly capture context information and is susceptible to noise. Riezler et al. developed a QE system by retraining a standard phrase-based statistical machine translation (SMT) system using query-snippet pairs extracted from clickthrough data (Riezler et al. 2008; Riezler and Liu 2010). The SMT-based system can produce cleaner, more relevant expansion terms because rich context information useful for filtering noisy expansions is captured by combining language model and phrase translation model in its decoder. Furthermore, in the SMT system all component models are properly smoothed using sophisticated techniques to avoid sparse data problems while the correlation model relies on pure counts of term frequencies. However, the SMT system is used as a black box in their experiments. So the relative contribution of different SMT components is not verified empirically. In this study we break this black box in order to build a better, simpler QE system. We will show that the proposed lexicon models outperform significantly the term correlation model, and that a simpler QE system that incorporates the lexicon models can beat the sophisticated, black-box SMT system.

2 Lexicon Models

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, triplets, and topics, respectively, and the way these models are learned from query-title pairs extracted from clickthrough data.

2.1 Word Model

The word model takes the form of IBM Model 1 (Brown et al. 1993; Berger and Lafferty 1999). Let $Q = (q_1, \dots, q_J)$ be a query, e be an expansion term candidate, the translation probability from Q to e is defined as

$$P(e|Q)_{wm} = \sum_{j=1}^{J} P(e|q_j)P(q_j|Q) \qquad (1)$$

where $P(q|Q)$ is the unsmoothed unigram probability of word q in query Q. The word translation probabilities $P(e|q)$ are estimated on the query-title pairs derived from the clickthrough data by assuming that the title terms are likely to be the desired expansions of the paired query. Our training method follows the standard procedure of training statistical word alignment models proposed by Brown et al. (1993). Formally, we optimize the model parameters θ by maximizing the probability of generating document titles from queries over the entire training corpus:

$$\theta^* = \text{argmax}_\theta \prod_{i=1}^{H} P(D_i|Q_i, \theta) \qquad (2)$$

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

$$P(D|Q, \theta) = \frac{\varepsilon}{(J+1)^I} \prod_{i=1}^{I} \sum_{j=1}^{J} P(w_i|q_j, \theta) \qquad (3)$$

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

2.2 Triplet Model

The word model is context independent. The triplet model, which is originally proposed for SMT (Hasan et al. 2008), is intended to capture inter-term dependencies for selecting expansion terms. The model is based on lexicalized triplets (e, q, q') which can be understood as two query terms triggering one expansion term. The translation probability of e given Q for the triplet model is parameterized as

$$P(e|Q)_{tm} = \frac{1}{Z}\sum_{j=1}^{J-1}\sum_{k=j+1}^{J} P(e|q_j, q_k) \qquad (4)$$

where Z is a normalization factor based on the corresponding query length, i.e., $Z = \frac{J(J-1)}{2}$, and $P(e_i|q_j, q_k)$ is the probability of translating q_j into e_i given another query word q_k. Since q_k can be any word in Q that is not necessary to be adjacent to q_j, the triple model is able to combine local (i.e. word and phrase level) and global (i.e. query level) contextual information useful for word translation.

Similar to the case of word model, we used the EM algorithm to estimate the translation probabilities $P(e|q, q')$ on the query-title pairs. Since the number of all possible triplets (e, q, q') is large and as a consequence the model training could suffer the data sparseness problem, in our experiments count-based cutoff is applied to prune the model to a manageable size.

2.3 Bilingual Topic Model (BLTM)

The BLTM was originally proposed for Web document ranking by Gao et al. (2011). The idea underlying the model is that a search query and its relevant Web documents share a common distribution of (hidden) topics, but use different (probably overlapping) vocabularies to express these topics. Intuitively, BLTM-based QE works as follows. First, a query is represented as a vector of topics. Then, all the candidate expansion terms, which are selected from document, are ranked by how likely it is that these document terms are selected to best describe those topics. In a sense, BLTM is similar to the word model and the triplet model since they all map a query to a document word. BLTM differs in that the mapping is performed at the topic level (via a language independent semantic representation) rather than at the word level. In our experiments BLTM is found to often select a different set of expansion terms and is complementary to the word model and the triplet model.

Formally, BLTM-based QE assumes the following story of generating e from Q:

1. First, for each topic z, a pair of different word distributions $(\boldsymbol{\phi}_z^Q, \boldsymbol{\phi}_z^D)$ are selected from a Dirichlet prior with concentration parameter β, where $\boldsymbol{\phi}_z^Q$ is a topic-specific query term distribution, and $\boldsymbol{\phi}_z^D$ a topic-specific document term distribution. Assuming there are T topics, we have two sets of distributions $\boldsymbol{\Phi}^Q = (\boldsymbol{\phi}_1^Q, \dots, \boldsymbol{\phi}_T^Q)$ and $\boldsymbol{\Phi}^D = (\boldsymbol{\phi}_1^D, \dots, \boldsymbol{\phi}_T^D)$.
2. Given Q, a topic distribution $\boldsymbol{\theta}^Q$ is drawn from a Dirichlet prior with concentration parameter α.
3. Then a document term (i.e., expansion term candidate) e is generated by first selecting a topic z according to the topic distribution $\boldsymbol{\theta}^Q$, and then drawing a word from $\boldsymbol{\phi}_z^D$.

By summing over all possible topics, we end up with the following model form

$$P_{bltm}(e|Q) = \sum_z P(e|\boldsymbol{\phi}_z^D) P(z|\boldsymbol{\theta}^Q) \qquad (5)$$

The BLTM training follows the method described in Gao et al. (2011). We used the EM algorithm to estimate the parameters $(\boldsymbol{\theta}, \boldsymbol{\Phi}^Q, \boldsymbol{\Phi}^D)$ of BLTM by maximizing the joint log-likelihood of the query-title pairs and the parameters. In training, we also constrain that the paired query and title have similar fractions of tokens assigned to each topic. The constraint is enforced on expectation using posterior regularization (Ganchev et al. 2010).

Original query	jaguar locator
Ranked expansion candidates (altered words are in italic)	jaguar *finder* *car* locator jaguar *location* jaguar *directory* … jaguar *list*
Expanded query (selected expansion terms are in italic)	`OR` (jaguar, *car*) `OR` (locator, *finder*, *location*, *directory*)

Figure 1. An example of an original query, its expansion candidates and the expanded query generated by the ranker-based QE system.

3 A Ranker-Based QE System

This section describes a ranker-based QE system in which the three lexicon models described above are incorporated. The system expands an input query in two distinct stages, candidate generation and ranking, as illustrated by an example in Figure 1.

In candidate generation, an input query Q is first tokenized into a sequence of terms. For each term q that is not a stop word, we consult a word model described in Section 2.1 to identify the best M altered words according to their word translation probabilities from q. Then, we form a list of expansion candidates, each of which contains all the original words in Q except for the word that is substituted by one of its altered words. So, for a query with J terms, there are at most $M \times J$ candidates.

In the second stage, all the expansion candidates are ranked using a ranker that is based on the Markov Random Field (MRF) model in which the three lexicon models are incorporated as features. Expansion terms of a query are taken from those terms in the N-best ($N = 10$ in our experiments) expansion candidates of the query that have not been seen in the original query string.

In the remainder of this section we will describe in turn the MRF-based ranker, the ranking features, and the way the ranker parameters are estimated.

3.1 MRF-Based Ranker

The ranker is based on the MRF model that models the joint distribution of $P_\Lambda(E, Q)$ over a set of expansion term random variables $E = \{e_1, \dots, e_I\}$ and a query random variable Q. It is constructed from a graph G consisting of a query node and nodes for each expansion term. Nodes in the graph represent random variables and edges define the independence semantics between the variables. An MRF satisfies the Markov property (Bishop 2006), 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

$$P_\Lambda(E, Q) = \frac{1}{Z_\Lambda} \prod_{c \in C(G)} \varphi(c; \Lambda) \qquad (6)$$

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 ranking expansion candidates, we can drop the expensive computation of Z_Λ since it is independent of E, and simply rank each expansion candidate E 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 $P_\Lambda(E|Q)$ as

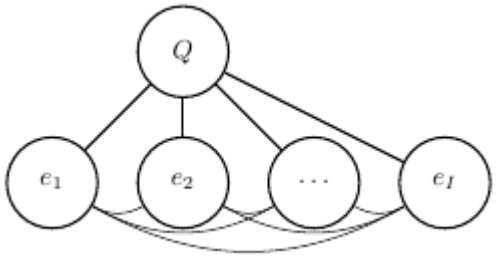

Figure 2: The structure of the Markov random field for representing the term dependency among the query Q and the expansion terms $e_1, \dots, e_I$.

$$P_\Lambda(E|Q) = \frac{P_\Lambda(E, Q)}{P_\Lambda(Q)} \qquad (7)$$

$$\stackrel{rank}{\Longrightarrow} \sum_{c \in C(G)} \log \varphi(c; \Lambda) = \sum_{c \in C(G)} \lambda_c f(c),$$

which is essentially a weighted linear combination of a set of features.

Therefore, to instantiate the MRF model, one needs to define a graph structure and a set of potential functions. In this paper, the graphical model representation we propose for QE is a fully connected graph shown in Figure 2, where all expansion terms and the original query are assumed dependent with each other. In what follows, we will define six types of cliques that we are interested in defining features (i.e., potential functions) over.

3.2 Features

The cliques and features are inspired by the component models used in SMT systems. The cliques defined in G for MRF can be grouped into two categories. The first includes three types of cliques involving both the query node and one or more expansion terms. The potential functions defined over these cliques attempt to abstract the idea behind the query to title translation models. The other three types, belonging to the second category, involve only expansion terms. Their potential func-

tions attempt to abstract the idea behind the target language models.

The first type of cliques involves a single expansion term and the query node. The potentials functions for these cliques are defined as

$$\varphi_T(e_i, Q; \Lambda) = \exp[\lambda_{wm} f_{wm}(e_i, Q) + \lambda_{tm} f_{tm}(e_i, Q) + \lambda_{bltm} f_{bltm}(e_i, Q)] \quad (6)$$

where the three feature functions of the form $f(e, Q)$ are defined as the log probabilities of translating Q to e according to the word, triplet and topic models defined in Equations (1), (4) and (5), respectively.

$$f_{wm}(e_i, Q) = \log P_{wm}(e_i|Q)$$
$$f_{tm}(e_i, Q) = \log P_{tm}(e_i|Q)$$
$$f_{bltm}(e_i, Q) = \log P_{bltm}(e_i|Q)$$

The second type of cliques contains the query node and two expansion terms, e_i and e_{i+1}, which appear in consecutive order in the expansion. The potential functions over these cliques are defined as

$$\varphi_{ob}(e_i, e_{i+1}, Q; \Lambda) = \exp[\lambda_{ob} f_{ob}(e_i, e_{i+1}, Q)] \quad (7)$$

where the feature $f_{ob}(.)$ is defined as the log probability of generating an expansion bigram given Q

$$f_{ob}(e_i, e_{i+1}|Q) = \log P(e_i, e_{i+1}|Q)$$

Unlike the language models used for document ranking (e.g., Zhai and Lafferty 2001), we cannot compute the bigram probability by simply counting the relative frequency of (e_i, e_{i+1}) in Q because the query is usually very short and the bigram is unlikely to occur. Thus, we approximate the bigram probability by assuming that the words in Q are independent with each other. We thus have

$$P(e_i, e_{i+1}|Q) = \frac{P(e_i, e_{i+1}, Q)}{P(Q)} = \frac{P(e_i)P(e_{i+1}|e_i)\prod_{j=1}^{J} P(q_j|e_i, e_{i+1})}{\prod_{j=1}^{J} P(q_j)},$$

where $P(q_j|e_i, e_{i+1})$ is the translation probability computed using a variant of the triplet model described in Section 2.2. The model variation differs from the one of Equation (4) in two respects. First, it models the translation in a different direction i.e., from expansion to query. Second, we add a constraint to the triplets such that (e_i, e_{i+1}) must be an ordered, contiguous bigram. The model variation is also trained using EM on query-title pairs. $P(e_i)$ and $P(e_{i+1}|e_i)$ are assigned respectively by the unigram and bigram language models, estimated from the collection of document titles of the clickthrough data, and $P(q_j)$ is the unigram probability of the query term, estimated from the collection of queries of the clickthrough data.

The third type of cliques contains the query node and two expansion terms, e_i and e_k, which occur unordered within the expansion. The potential functions over these cliques are defined as

$$\varphi_{ub}(e_i, e_k, Q; \Lambda) = \exp[\lambda_{ub} f_{ub}(e_i, e_k, Q)] \quad (8)$$

where the feature $f_{ub}(.)$ is defined as the log probability of generating a pair of expansion terms (e_i, e_k) given Q

$$f_{ub}(e_i, e_k|Q) = \log P(e_i, e_k|Q).$$

Unlike $f_{ob}(e_i, e_{i+1}|Q)$ defined in Equation (7), this class of features captures long-span term dependency in the expansion candidate. Similar to the computation of $P(e_i, e_{i+1}|Q)$ in Equation (7), we approximate $P(e_i, e_k|Q)$ as

$$P(e_i, e_k|Q) = \frac{P(e_i, e_k, Q)}{P(Q)} = \frac{P(e_i)P(e_k|e_i)\prod_{j=1}^{J} P(q_j|e_i, e_k)}{\prod_{j=1}^{J} P(q_j)},$$

where $P(q_j|e_i, e_k)$ is the translation probability computed using the triplet model described in Section 2.2, but in the expansion-to-query direction. $P(e_i)$ is assigned by a unigram language model estimated from the collection of document titles of the clickthrough data. $P(e_k|e_i)$ is assigned by a co-occurrence model, estimated as

$$P(e_k|e_i) = \frac{N(e_i, e_k)}{\sum_e N(e_i, e)}$$

where $N(e_i, e_k)$ is the number of times that the two terms occur in the same title in clickthrough data.

We now turn to the other three types of cliques that do not contain the query node. The fourth type of cliques contains only one expansion term. The potential functions are defined as

$$\varphi_{um}(e_i;\Lambda) = \exp[\lambda_{um} f_{um}(e_i)] \quad (9)$$
$$f_{um}(e_i) = \log P(e_i)$$

where $P(e_i)$ is the unigram probability computed using a unigram language model trained on the collection of document titles.

The fifth type of cliques contains a pair of terms appearing in consecutive order in the expansion. The potential functions are defined as

$$\varphi_{bm}(e_i, e_{i+1};\Lambda) = \exp[\lambda_{bm} f_{bm}(e_i, e_{i+1})] \quad (10)$$
$$f_{bm}(e_i, e_{i+1}) = \log P(e_{i+1}|e_i)$$

where $P(e_{i+1}|e_i)$ is the bigram probability computed using a bigram language model trained on the collection of document titles.

The sixth type of cliques contains a pair of terms appearing unordered within the expansion. The potential functions are defined as

$$\varphi_{cm}(e_i, e_k;\Lambda) = \exp[\lambda_{cm} f_{cm}(e_i, e_k)] \quad (11)$$
$$f_{cm}(e_i, e_k) = \log P(e_k|e_i)$$

where $P(e_k|e_i)$ is the assigned by a co-occurrence model trained on the collection of document titles.

3.3 Parameter Estimation

The MRF model uses 8 classes of features defined on 6 types of cliques, as in Equations (6) to (11). Following previous work (e.g., Metzler and Croft 2005; Bendersky et al. 2010), we assume that all features within the same feature class are weighted by the same tied parameter λ. Thus, the number of free parameters of the MRF model is significantly reduced. This not only makes the model training easier but also improves the robustness of the model. After tying the parameters and using the exponential potential function form, the MRF-based ranker can be parameterized as

$$P_\Lambda(E|Q) \overset{rank}{\Longrightarrow} \lambda_{wm} \sum_{i=1}^{I} f_{wm}(e_i, Q) + \quad (12)$$
$$\lambda_{tm} \sum_{i=1}^{I} f_{tm}(e_i, Q) +$$
$$\lambda_{bltm} \sum_{i=1}^{I} f_{bltm}(e_i, Q) +$$
$$\lambda_{ob} \sum_{i=1}^{I-1} f_{ob}(e_i, e_{i+1}, Q) +$$
$$\lambda_{ub} \sum_{i=1}^{I-1} \sum_{k=i+1}^{I} f_{ob}(e_i, e_k, Q) +$$
$$\lambda_{um} \sum_{i=1}^{I} f_{um}(e_i) +$$
$$\lambda_{bm} \sum_{i=1}^{I-1} f_{bm}(e_i, e_{i+1}) +$$
$$\lambda_{cm} \sum_{i=1}^{I-1} \sum_{k=i+1}^{I} f_{cm}(e_i, e_k)$$

where there are in total 8 λ's to be estimated.

Although the MRF is by nature a generative model, it is not always appropriate to train the parameters using conventional likelihood based approaches due to the metric divergence problem (Morgan et al. 2004): i.e., the maximum likelihood estimate is unlikely to be the one that optimizes the evaluation metric. In this study the effectiveness of a QE method is evaluated by first 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.

For this reason, the parameters of the MRF-based ranker are optimized directly for Web search. In our experiments, the objective in training is *Normalized Discounted Cumulative Gain* (NDCG, Jarvelin and Kekalainen 2000), which is widely used as quality measure for Web search. Formally, we view parameter training as a multi-dimensional optimization problem, with each feature class as one dimension. Since NDCG is not differentiable, we tried in our experiments numerical algorithms that do not require the computation of gradient. Among the best performers was the Powell Search algorithm (Press et al., 1992). It first constructs a set of N virtual directions that are conjugate (i.e., independent with each other), then it uses *line search* N times ($N = 8$ in our case), each on one virtual direction, to find the optimum. Line search is a one-dimensional optimization algorithm. Our implementation follows the one described in Gao et al. (2005), which is used to optimize averaged precision.

4 Experiments

We evaluate the performance of a QE method by first 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., Cui et al. 2003; Riezler et al. 2008), 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 previous state-of-the-art log-based QE methods.

The relevance judgment set consists of 4,000 multi-term English queries. On average, each query is associated with 197 Web documents (URLs). Each query-URL 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 reflect a natural query distribution, we do not try to control the quality of these queries. For example, in our query sets, there are roughly 20% misspelled queries, 20% navigational queries, and 10% transactional queries. 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 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 contains 2,000 queries.

The query-title pairs used for model training are extracted from one year of query log files using a procedure similar to Gao et al. (2009). In our experiments we used a randomly sampled subset of 20,692,219 pairs that do not overlap the queries and documents in the test set.

#	QE methods	NDCG@1	NDCG@3	NDCG@10
1	**NoQE**	34.70	36.50	41.54
2	**TC**	33.78	36.57	42.33$^{\alpha}$
3	**SMT**	34.79$^{\beta}$	36.98$^{\alpha\beta}$	42.84$^{\alpha\beta}$
4	**MRF**	**36.10**$^{\alpha\beta\gamma}$	**38.06**$^{\alpha\beta\gamma}$	**43.71**$^{\alpha\beta\gamma}$
5	**MRF**$_{um+bm+cm}$	33.31	36.12	42.26$^{\alpha}$
6	**MRF**$_{tc}$	34.50$^{\beta}$	36.59	42.33$^{\alpha}$
7	**MRF**$_{wm}$	34.73$^{\beta}$	36.62	42.73$^{\alpha\beta}$
8	**MRF**$_{tm}$	35.13$^{\alpha\beta}$	37.46$^{\alpha\beta\gamma}$	42.82$^{\alpha\beta}$
9	**MRF**$_{bltm}$	34.34$^{\beta}$	36.19	41.98$^{\alpha}$
10	**MRF**$_{wm+tm}$	35.21$^{\alpha\beta\gamma}$	37.46$^{\alpha\beta\gamma}$	42.83$^{\alpha\beta}$
11	**MRF**$_{wm+tm+bltm}$	35.84$^{\alpha\beta\gamma}$	37.70$^{\alpha\beta\gamma}$	43.14$^{\alpha\beta\gamma}$

Table 1: Ranking results using BM25 with different query expansion systems. The superscripts $\alpha, \beta,$ and γ indicate statistically significant improvements ($p < 0.05$) over **NoQE**, **TC**, and **SMT**, respectively. Rows 5 to 11 are different versions of **MRF** in Row 5, They use the same candidate generator but use in the ranker different feature classes, as specified by the subscript. **tc** specifies the feature class defined as the scoring function in Equation (13). Refer to Equation (12) for the names of other feature classes.

Our Web document collection consists of approximately 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 Web search performance is evaluated by mean NDCG. We report NDCG scores at truncation levels of 1, 3, and 10. We also perform a significance test using the paired *t*-test. Differences are considered statistically significant when *p*-value is less than 0.05.

4.1 Comparing Systems

Table 1 shows the main document ranking results using different QE systems, developed and evaluated using the datasets described above.

NoQE (Row 1) is the baseline retrieval system that uses the raw input queries and the BM25 document ranking model. Rows 2 to 4 are different QE systems. Their results are obtained by first expanding a query, then using BM25 to rank the documents with respect to the expanded query.

TC (Row 2) is our implementation of the correlation-based QE system (Cui et al. 2002; 2003). It takes the following steps to expand an input 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 (where the value of n is optimized on training data) and formulate the expanded query by adding these terms into Q.
5. Use the expanded query to rank documents.

The scoring function is based on the term correlation model, and is defined as

$$Score(w|Q) = \ln\left(\prod_{q\in Q} P(w|q) + 1\right) \qquad (13)$$

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

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. Table 1 shows that **TC** leads to significant improvement over **NoQE** in NDCG@10, but not in NDCG@1 and NDCG@3 (Row 2 vs. Row 1). The result is not entirely consistent with what reported in Cui et al. (2003). A possible reason is that Cui et al. performed the evaluation using documents and search logs collected from the Encarta website, which is much cleaner and more homogenous than the data sets we used. The result suggests that although QE improves the recall of relevant documents, it is also likely to introduce noise that hurts the precision of document retrieval.

SMT (Row 3) is a SMT-based QE system. Following Riezler et al. (2008), the system is an implementation of a phrase-based SMT system with a standard set of features for translation model and language model, combined under a log linear model framework (Koehn et al. 2003). 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. We see that **SMT** significantly outperforms **TC** in NDCG at all levels. The result confirms the conclusion of Riezler et al., demonstrating that context information is crucial for improving retrieval precision by filtering noisy expansions.

Both **TC** and **SMT**, considered as state-of-the-art QE methods, have been frequently used for comparison in related studies. Thus, we also used them as baselines in our experiments.

MRF (Row 4) is the ranker-based QE system described in Section 3, which uses a MRF-based ranker to incorporate all 8 classes of features derived from a variety of lexicon translation models and language models as in Equation (12). Results show that the ranker-based QE system significantly outperforms both **NoQE** and the two state-of-the-art QE methods. The fact that **MRF** beats **SMT** with a statistically significant margin although the former is a much simpler system indicates that text translation and QE are different tasks and some SMT components, designed for the task of regular text translation, are not as effective in selecting expansion terms. We will explore this in more detail in the next section.

4.2 Comparing Models

The experiments presented in this section investigate in detail the effectiveness of different models, e.g., the lexicon models and the language models described in Sections 2 and 3, in ranking expansion candidates for QE. The results are summarized in Rows 5 to 11 in Table 1, where a number of different versions of the ranker-based QE system are compared. These versions, labeled as $\mathbf{MRF}_f$, use the same candidate generator, and differ in the feature classes (which are specified by the subscript f) incorporated in the MRF-based ranker. In what follows, we focus our discussion on the results of the three lexicon models.

$\mathbf{MRF}_{wm}$ (Row 7) uses the word translation model described in Section 2.1. Both the word model and term correlation model used in $\mathbf{MRF}_{tm}$ (Row 6) are context independent. They differ mainly in the training methods. For the sake of comparison, in our experiment the word model is

EM-trained with the correlation model as initial point. Rezler et al. (2008) hypothesize that statistical 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. Our result (Row 7 vs. Row 6) verifies the hypothesis. Notice that **MRF**$_{tc}$ outperforms **TC** in NDCG@1 (Row 6 vs. Row 2) mainly because in the former the expansion candidates are generated by a word translation model and are less noisy.

It is encouraging to observe that the rankers using the triplet model features achieve the QE performance either in par with or better than that of **SMT** (Rows 8, 10 and 11 vs. Row 3), although the latter is a much more sophisticated system. The result suggests that not all SMT components are useful for QE. For example, language models are indispensable for translation but are less effective than word models for QE (Row 5 vs. Rows 6 and 7). We also observe that the triplet model not only outperforms significantly the word model due to the use of contextual information (Row 8 vs. Row 7), but also seems to subsume the latter in that combining the features derived from both models in the ranker leads to little improvement over the ranker that uses only the triplet model features (Row 10 vs. Row 8).

The bilingual topic model underperforms the word model and the triplet model (Row 9 vs. Rows 7 and 8). However, we found that the bilingual topic model often selects a different set of expansion terms and is complementary to the other two lexicon models. As a result, unlike the case of combining the word model and triplet model features, incorporating the bilingual topic model features in the ranker leads to some visible improvement in NDCG at all positions (Row 11 vs. Row 10).

To better understand empirically how the MRF-based QE system achieves the improvement, we analyzed the expansions generated by our system in detail and obtained several interesting findings. First, as expected, in comparison with the word model, the triplet translation model is more effective in benefitting long queries, e.g., notably queries containing questions and queries containing song lyrics. Second, unlike the two lexicon models, the bilingual topic model tends to generate expansions that are more likely to relate to an entire query rather than individual query terms. Third, the features involving the order of the expansion terms benefitted queries containing named entities.

5 Related Work

In comparison with log-based methods studied in this paper, the QE methods based on automatic relevance feedback have been studied much more extensively in the information retrieval (IR) community, and have been proved useful for improving IR performance on benchmark datasets such as TREC (e.g., Rocchio 1971; Xu and Croft 1996; Lavrenko 2001; Zhai and Lafferty 2001). 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. Although automatic relevance feedback is not the focus of this study, our method shares a lot of similarities with some of them. For example, similar to the way the parameters of our QE ranker are estimated, Cao et al. (2008) propose a method of selecting expansion terms to directly optimize average precision. The MRF model has been previously used for QE, in the form of relevance feedback and pseudo-relevance feedback (Metzler et al. 2007; Lang et al. 2010). While their MRF models use the features derived from IR systems such as Indri, we use the SMT-inspired features.

Using statistical translation models for IR is not new (e.g., Berger and Lafferty 1999; Jin et al. 2002; Xue et al. 2008). The effectiveness of the statistical translation-based approach to Web search has been demonstrated empirically in recent studies where word-based and phrase-based translation models are trained on large amounts of clickthrough data (e.g., Gao et al. 2010a; 2011). Our work extends these studies and constructs QE-oriented translation models that capture more flexible dependencies.

In addition to QE, search logs have also been used for other Web search tasks, such as document ranking (Joachims 2002; Agichtein et al. 2006), search query processing and spelling correction (Huang et al. 2010; Gao et al. 2010b) image retrieval (Craswell and Szummer 2007), and user query clustering (Baeza-Yates and Tiberi 2007; Wen et al. 2002).

6 Conclusions

In this paper we extend the previous log-based QE methods 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 techniques developed for SMT to QE. Specially, we propose three lexicon models based on terms, lexicalized triplets, and topics, respectively. These models are trained on pairs of user queries and the titles of clicked documents using EM. Second, we present a ranker-based QE system, the heart of which is a MRF-based ranker in which the lexicon models are incorporated as features. We perform experiments on the Web search task using a real world data set. Results show that the proposed system outperforms significantly other state-of-the-art QE systems.

This study is part of a bigger, ongoing project, aiming to develop a real-time QE system for Web search, where simplicity is the key to the success. Thus, what we learned from this study is particularly encouraging. We demonstrate that with large amounts of clickthrough data for model training, simple lexicon models can achieve state-of-the-art QE performance, and that the MRF-based ranker provides a simple and flexible framework to incorporate a variety of features capturing different types of term dependencies in such an effective way that the Web search performance can be directly optimized.

References

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

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

Baeza-Yates, R. and Tiberi, A. 2007. Extracting semantic relations from query logs. In *SIGKDD*, pp. 76-85.

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.

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

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

Bishop, C. M. 2006. *Patten recognition and machine learning.* Springer.

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

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.

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

Craswell, N. and Szummer, M. 2007. Random walk on the click graph. In *SIGIR*. pp. 239-246.

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

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.

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.

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.

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

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

Gao, J., Li, X., Micol, D., Quirk, C., and Sun, X. 2010b. A large scale ranker-based system for query spelling correction. In *COLING*, pp. 358-366.

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.

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

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

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.

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

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

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

Joachims, T. 2002. Optimizing search engines using clickthrough data. In *SIGKDD*, pp. 133-142.

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

Lang, H., Metzler, D., Wang, B., and Li, J-T. 2010. Improving latent concept expansion using markov random fields. In *CIKM*, pp. 249-258.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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.