Modeling Click-through Based Word-pairs for Web Search

Jagadeesh Jagarlamudi*
Department of Computer Science
University of Maryland
College Park, USA
jags@umiacs.umd.edu

Jianfeng Gao
Microsoft Research
One Microsoft Way
Redmond, WA 98052, U.S.A.
jfgao@microsoft.com

ABSTRACT

Statistical translation models and latent semantic analysis (LSA) are two effective approaches to exploiting click-through data for Web search ranking. While the former learns semantic relationships between query terms and document terms directly, the latter maps a document and the queries for which it has been clicked to vectors in a lower-dimensional semantic space. This paper presents two document ranking models that combine the strengths of both the approaches by explicitly modeling word-pairs. The first model, called PairModel, is a monolingual ranking model based on word-pairs derived from click-through data. It maps queries and documents into a concept space spanned by these word-pairs. The second model, called Bilingual Paired Topic Model (BPTM), uses bilingual word translations and can jointly model query-document collections written in multiple languages. This model uses topics to capture term dependencies and maps queries and documents in multiple languages into a lower dimensional semantic sub-space spanned by the topics. These models are evaluated on the Web search task using real world data sets in three different languages. Results show that they consistently outperform various state-of-the-art baseline models, and the best result is obtained by interpolating PairModel and BPTM.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; I.2.6 [**Artificial Intelligence**]: Learning; I.5.4 [**Pattern Recognition**]: Applications—*Text Processing*

General Terms

Learning, Algorithms, Experimentation

*Work done during his internship at Microsoft Research.

SIGIR'13, July 28–August 1, 2013, Dublin, Ireland.

Keywords

Click-through Data, Latent Semantic Analysis, Topic Models, Multilingual IR, Translation Model, Web Search

1. INTRODUCTION

Web search engines till date rely strongly on matching words in a query-document pair. But very often, words tend to be either synonymous or polysemous resulting in a concept being expressed in different ways. It is well known that pure lexical matching is inaccurate and often leads to suboptimal performance [19, 32]. This problem has been addressed in mainly two ways:

- By using pairwise word associations to capture the probability of a document word translating into a query word [5]. The pairwise associations are typically learnt using document's content and its title [26], based on click-through data [3, 25, 14] or using query expansion techniques [35, 27].
- By representing documents and queries in a lower dimensional semantic space, in which words are identified by their semantics and not by their lexical form [13, 18]. Thus a query and document pair can have a non-zero similarity even though they don't share any terms.

The use of click-through data has proved to be effective in both the approaches. Statistical translation based approaches, first, learn pairwise associations between query and document words and then use these translation probabilities to rank the documents (Sec. 2.1) [14, 3]. On the other hand, topic modeling based approaches treat a document and the set of queries for which it has been clicked as an *aligned* query document pair and learn a shared topic distribution [15] (Sec. 2.2). They use the shared topic distribution to rank documents. A serious limitation of these topic modeling based approaches is that they require aligned query document pairs. But, there are lot of user queries without any click information rendering their application limited.[1] Notice that both these kinds of approaches harness different aspects of the click-through data. In specific, translation models use click-through data to mine pairwise word associations while topic modeling based approaches exploit alignment information to learn a shared topic distribution.

[1]In the literature, click-through data usually refers to clicked query document pairs. But in this paper we refer to both clicked and not-clicked query document pairs. We use aligned and unaligned to distinguish between these two sets.

1	The celebration of Thanksgiving will be incomplete without the legendary **Turkey** bird. The customary dinner reminds of the 'Four Wild Turkeys' served $\cdots$
2	**Turkey**, known officially as the Republic of **Turkey** is a Eurasian country located in Western Asia and in East Thrace in Southeastern Europe. $\cdots$

Table 1: Example documents where the term 'turkey' is used in different contexts. Relevant context words are colored in Blue.

So it is natural to investigate if exploiting these two aspects together will yield any further improvements.

With this goal, we aim to design models which can use pairwise word associations to jointly model an aligned and unaligned query-document collections. We achieve this by using word-pairs (*e.g.*, Table 2) to disambiguate the usage of terms. To understand the intuition, consider the occurrence of a term like 'turkey' in a document or a query. This term can mean the "Turkey bird" or the "Turkey country". Our approach tries to identify its most relevant concept based on its context words. For example, in the first document of Table 1, the occurrence of terms like 'thanksgiving' and 'dinner' along with 'turkey' indicates that it is used in the bird concept while the occurrence of terms like 'republic' and 'country', in the second document, indicates that it is used in the country concept. In our models, we try to disambiguate the term with the help of word-pairs. In this particular example, the bird concept is captured by associating the term to the word-pairs "turkey:thanksgiving" and "turkey:dinner" while the country concept is identified by associating it with "turkey:country" and "turkey:republic". By linking two related words, via a word-pair, we gather the evidence from both the words to make better judgements about of each of the words. Thus, we aim to build better document models which in turn lead to better ranking models.

Using this idea, we propose two models which explicitly use word-pairs to model a query-document collection. Our first model, called PairModel, is a monolingual model. It uses monolingual word-pairs derived from click-through data to model term dependencies in a query-document collection. It maps queries and documents into a concept space spanned by these word-pairs (Sec. 3). As shown in Table 2, the word-pairs capture different types of term correlations such as morphological variations, misspellings, term relatedness, *etc.*. By accounting for these different variations, PairModel builds better document models. The use of term co-occurrences to build better query or document models is not new [34], but the way we use it (*i.e.*, by combining word-pairs and topic models) is novel.

Our second model is a bilingual model called Bilingual Paired Topic Model (BPTM). It uses bilingual word-pairs, that are translations of each other say between French and English, to model a *bilingual* query-document collection (Sec. 4). Unlike PairModel, the word-pairs in BPTM are word translations and do not capture other types of semantic term correlations. So, we introduce topics to capture the semantic relatedness. This model maps queries and documents of both languages into a common lower dimensional subspace. Thus BPTM addresses the lexical gap problem and also leverages the abundant training data available in a resource rich *assisting* language (such as English) to improve ranking in a resource poor *search* language. While the use of assisting language to improve search language's ranking is not new [20, 10], the way we use it (*i.e.*, via topic modeling) is novel.

PairModel exploits term dependencies from queries and documents within the search language (French) while BPTM uses information gathered from the assisting (English) language to make better relevance judgements. So, these models are complimentary to each other and can be interpolated. Our experiments (Sec. 5) in French and German languages, using English as the assisting language, show that the combined model outperforms both the individual models and also beats various state-of-the-art baseline systems by a significant margin.[2]

2. RELATED WORK

Many strategies have been proposed to bridge the lexical gap between queries and documents using the click-through data. Click-through data is also shown to be effective in the learning to rank framework [1], but here we discuss the work that is most relevant to our problem and our approach.

2.1 Statistical Translation Based Approaches

In Language Modeling (LM) framework [29], documents are ranked based on the likelihood of generating a query. Statistical translation based approaches address the lexical gap by ranking the documents based on the likelihood of *translating* into a query. Let $\mathbf{q} = \{q_1, \ldots q_{N_q}\}$ be a query and $\mathbf{d} = \{w_1, \ldots w_{N_d}\}$ be a document, then a word based translation model [5] ranks the documents based on:

$$P(\mathbf{q}|\mathbf{d}) = \prod_{q \in \mathbf{q}} P(q|\mathbf{d}) = \prod_{q \in \mathbf{q}} \sum_{w \in \mathbf{d}} P(q|w) P(w|\mathbf{d}) \qquad (1)$$

where $P(w|\mathbf{d})$ is the unigram probability of the word w in $\mathbf{d}$, and $P(q|w)$ is the probability of the document word w translating into the query word q. In these methods, a major challenge is the estimation of the translation probabilities $P(q|w)$. An ideal training data would be a large amount of aligned query-document pairs (in which each of the document is judged as relevant to the query). Due to the lack of such training data, [5] resorts to some synthetic query-document pairs while [26] uses the title-document pairs for estimating the translation probabilities. Click-through data has been explored to determine relationships between terms in queries and documents [3, 25]. However, these relationships and their probabilities are created using ad hoc similarity measures. Gao *et al.* [14] take a word alignment based approach popular in the Statistical Machine Translation community [8]. They treat an aligned query-document pair as a parallel sentence pair in two different languages and use IBM Model 1 to learn the translation probabilities. They show an improved performance by using translation probabilities learned from an year worth of click-through data.

Wei and Croft [34] discuss several co-occurrence based methods to identify term associations and their effectiveness for information retrieval. Moreover, lexical gap problem is also addressed using query expansion with automatic relevance feedback (*e.g.*, pseudo relevance feedback or PRF) [35]. Though these techniques are shown to be effective on TREC benchmark data sets [35, 27, 37, 9], their applicability to a commercial Web search engine is limited because

[2]A shorter version of this document is available at [22].

generating pseudo-relevant documents requires multi-phase retrieval, which is prohibitively expensive.

2.2 LSA Based Approaches

Latent Semantic Analysis (LSA) based approaches assume that both queries and documents lie in a lower dimensional sub-space and try to learn this space [13]. Probabilistic Latent Semantic Analysis (PLSA) [18] assumes that each document is a Multinomial distribution over T topics (called document-topic distribution) where each of the topics is in turn a Multinomial distribution over words (called topic-word distribution). Like in LM framework, the relevance of a document towards a query is assumed to be proportional to the likelihood of it generating the query. Latent Dirichlet Allocation (LDA) generalizes PLSA to unseen documents by employing a conjugate Dirichlet prior on the document-topic distributions [6]. Though, LDA is superior to PLSA in theory, its effectiveness for IR, as demonstrated in [33], has not been compared against PLSA.

Recently, Gao *et al.* [15] proposed a generative model called Bilingual Topic Model (BLTM) for Web wearch. They show that, by including the click-through data, their model achieves better performance compared to the PLSA. They assume that an aligned query and document pair share the document-topic distribution. Given this document-topic distribution, the query and the document are assumed to be independent and are generated separately. During the ranking stage, they keep the topic-word distributions fixed and fold in the unseen documents to learn their document-topic distributions. Finally, for a given query $\mathbf{q}$, they rank the documents as follows:

$$P(\mathbf{q}|\mathbf{d}) = \prod_{q \in \mathbf{q}} P(q|\mathbf{d}) = \prod_{q \in \mathbf{q}} \sum_{z=1}^{T} P(q|z)P(z|\mathbf{d}) \qquad (2)$$

Notice that this approach exploits the alignment information between queries and documents to learn the sub-space and hence it is limited to aligned data sets.

2.3 Dictionary Based Semantic Models

On the other hand, there are approaches that learn semantic sub-space using word-pairs [21, 7, 38]. These approaches usually model unaligned bilingual document collections using bilingual dictionaries, but they can be modified to model unaligned query-document collections using monolingual word-pairs. In [21], each document is assumed to be a Multinomial distribution over T bilingual topics. A bilingual topic is a Multinomial distribution over bilingual concepts (c) and each of these concepts generates a word depending on the document language. In these models, the probability of generating a string $\mathbf{s} = (w_1, \ldots, w_{N_s})$ given a document $\mathbf{d}$ is given by:

$$P(\mathbf{s}|\mathbf{d}) = \prod_{w_i \in \mathbf{s}} P(w_i|\mathbf{d}) = \prod_{w_i \in \mathbf{s}} \sum_{c,z} P(w_i|c)P(c|z)P(z|\mathbf{d}) \qquad (3)$$

where $z = 1 \ldots T$ is the bilingual topic indicator.

In this paper, we propose models that use ideas from the above mentioned approaches. More specifically, we 1) use IBM model 1 to learn word-pairs from the click-through data (Sec. 2.1), 2) exploit the association between an aligned query-document pair (Sec. 2.2) and 3) finally use the word-pairs, derived in step 1, as proxy for the concepts in the dictionary based approaches to model unaligned query-document collections as well. Thus our models combine the strengths of these individual approaches. As shown in Table 2, the word-pairs used for PairModel capture different types of semantic term correlations, so this model does not need topics. In contrast, BPTM uses bilingual word translations and can not account for the different types of variations and so it uses topics to learn the semantic space.

turkey	thanksgiving
turkey	bird
turkey	country
turkey	istanbul
government	govenment
colleges	college
university	colleges
addiction	rehab
camera	canon
camera	D3000

Table 2: Example monolingual word-pairs. The word 'govenment' is misspelled purposefully.

As explained in Sec. 1, our models use word-pairs to disambiguate the concept in which a term is used. The usage of term co-occurrences to build better document models is not new [4]. For example, most of the approaches discussed in Section 2.2 inherently use term co-occurrences in their process. But our approach uses only a subset of selected word-pairs (Sec. 5.2). As argued in [23], considering all possible pairwise associations can be noisy and hence using a pre-selected set of word-pairs is desirable.

3. PAIR MODEL

In this section, we propose our model assuming that we have monolingual pairwise word associations. An example set of word-pairs is shown in Table 2.[3] Our model is independent of the dictionary's source.

The most important aspect of our model is that we treat a word-pair as a hidden variable, *i.e.*, when we see a term we model the concept in which it is used as a hidden variable and try to infer it based on the context. The order of words in a word-pair doesn't carry any significance, *i.e.*, "addiction:rehab" is same as "rehab:addiction". By grouping two related words together we effectively combine the evidence of both words so that each word can help make an appropriate decision about the other word. Moreover, by linking evidence from related words, we move beyond bag-of-word based models and, since these words can lie anywhere in the document, our approach is different from other higher order models which consider n-gram relationships between words [30, 31].

We first describe our model for aligned query-document pairs and later discuss, briefly, on extending it to unaligned queries and documents. We assume that each aligned query-document pair is a distribution over word-pairs or concepts referred to as concept distribution. From now on, we use concept and word-pair interchangeably. Each concept is in turn a Binomial distribution over its two component words. Notice that there are no topics in this model and the only hidden variables are concepts. This is the primary difference

[3]For conciseness, we refer to this as monolingual dictionary.

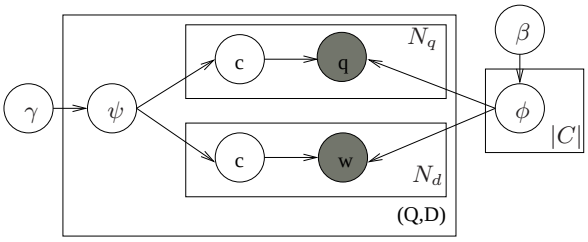


Figure 1: Graphical notation of the PairModel, the concept distribution is shared across an aligned query-document pair.

between our PairModel and other related generative models such as JointLDA [21] and BLTM [15].

Formally, we assume that an aligned query-document pair $(\mathbf{q}, \mathbf{d}) = \big((q_1, \ldots, q_{N_q}), (w_1, \ldots, w_{N_d})\big)$ share the concept distribution $\psi_{(\mathbf{q},\mathbf{d})}$[4] which is drawn from a Dirichlet distribution with symmetric prior γ. For each word in the query-document pair, we first draw a concept (c) from the concept distribution $\psi_{(\mathbf{q},\mathbf{d})}$ and then draw a word from the words associated with the concept. Let $|C|$ be the total number of concepts, then we assume the following generative story for generating an aligned query-document pair. The corresponding graphical notation is shown in Fig. 1.

1. For each concept $c = 1 \ldots |C|$, choose $\phi_c \sim \text{Beta}(\beta, \beta)$
2. For each aligned query-document pair
   (a) Choose $\psi_{(\mathbf{q},\mathbf{d})} \sim \text{Dir}(\gamma)$
   (b) For each document term $i = 1 \ldots N_d$
      - Select a concept $c_i \sim \text{Mult}(\psi_{(\mathbf{q},\mathbf{d})})$
      - Select a word $w_i \sim \text{Bin}(\phi_{c_i})$
   (c) For each query term $i = 1 \ldots N_q$
      - Select a concept $c_i \sim \text{Mult}(\psi_{(\mathbf{q},\mathbf{d})})$
      - Select a word $q_i \sim \text{Bin}(\phi_{c_i})$

Since the size of the monolingual dictionary is limited, there will be some words that do not have any pairwise associations. For example, the word 'world' is not part of the extremely small dictionary provided in Table 2. To handle these Out-of-Dictionary words, we add dummy concepts of the form word:word ("world:world" in this particular case). Notice that this concept can generate only one word 'world' with a probability of 1. The model can be easily extended to unaligned data, in which case the generative story is limited to either a query or a document.

To understand how this model uses context words to disambiguate the concept associated with a term and improve the search ranking, consider the example documents from Table 1 and the term 'turkey'. Based on the input monolingual dictionary, the word 'turkey' has four possible concepts associated with it (as shown in Table 2). When the term 'turkey' is seen in the first document, then all the four concepts are equally likely to have generated this term, so all of them get equal probability for this document. But when we see its context words, say 'thanksgiving' then the probability of the concept "thanksgiving:turkey" increases for this document as it is triggered for both the words 'turkey' and 'thanksgiving'. Likewise, the concept "turkey:bird" also becomes more probable when the word 'bird' is seen. Similarly, in the second document, the words 'country' and 'republic' will cause the concept distribution to peak for the country sense compared to the bird sense. When we link all the terms to their most relevant concepts and gather these statistics over all queries and documents, we obtain statistics that better fit this particular data set and hence they tend to be more accurate than the translation probabilities of the original dictionary (as evidenced in our experiments in Sec. 5.2) resulting in better ranking models.

In our model, word-pairs enable us to leverage the term correlations within a document or a query. At the same time, the fact that an aligned query-document pair share concept distribution means that it also leverages term correlations across a query-document pair.

3.1 Inference

In [2], authors show that MAP inference performs comparably to the best Bayesian inference methods for generative models such as LDA. So, we use Expectation Maximization [12] algorithm to learn the MAP values for (ϕ, ψ). The E-step involves finding the posterior probabilities, *i.e.*, the probability of associating each term with a concept, as follows:

$$P(c|w, (\mathbf{q}, \mathbf{d})) = \frac{P(w|\phi_c)P(c|\psi_{(\mathbf{q},\mathbf{d})})}{\sum_{c'} P(w|\phi_{c'})P(c'|\psi_{(\mathbf{q},\mathbf{d})})} \tag{4}$$

For each concept c, $P(w|\phi_c)$ is non-zero for only the two words that are part of this concept. So, the above posterior probability is non-zero for only those concepts that can generate the given word. Because of this, although there are a total of $O(V^2)$ possible concepts, only a tiny fraction of them will have non-zero probability in a given document. Moreover, this set can be precomputed based on the document words and can be stored in an appropriate data structure for efficient processing. Again, because on an average a word has a small number of associated concepts, most of the terms in the summation term in the denominator of Eq. 4 are zeros and hence it can also be computed efficiently.

The M-step updates the parameters based on the posterior probabilities estimated in the E-step. Let $n(q, \mathbf{q})$ denote the frequency of a query word q in the query $\mathbf{q}$ and, similarly, $n(w, \mathbf{d})$ denote the frequency of a word w in the document $\mathbf{d}$. Moreover, let $N(q, c, \mathbf{q}) = n(q, \mathbf{q})P(c|q, (\mathbf{q}, \mathbf{d}))$ and $N(w, c, \mathbf{d}) = n(w, \mathbf{d})P(c|w, (\mathbf{q}, \mathbf{d}))$ then,

$$P(c|\psi_{(\mathbf{q},\mathbf{d})}) \propto (\gamma - 1) + \sum_{q \in \mathbf{q}} N(q, c, \mathbf{q}) + \sum_{w \in \mathbf{d}} N(w, c, \mathbf{d})$$

$$P(w|\phi_c) \propto (\beta - 1) + \sum_{(\mathbf{q},\mathbf{d})} \Big(N(w, c, \mathbf{q}) + N(w, c, \mathbf{d})\Big)$$

3.2 PairModel Variants

In PairModel, we force an aligned query-document pair to share the concept distribution. Instead we can allow them to have different concept distributions and then take an average of them to denote the concept distribution of the pair. The corresponding graphical notation is shown in Fig. 2. During inference, we learn $\psi_{\mathbf{q}}$ and $\psi_{\mathbf{d}}$ independently and then use $\psi = \frac{\psi_{\mathbf{q}} + \psi_{\mathbf{d}}}{2}$ as the resulting concept distribution of the document. We refer to this model as PairModel(Averaged) in the rest of the paper.

[4]We use ψ instead of the traditional θ to remind the reader that this is not a distribution over topics, instead it is a distribution over the word-pairs.

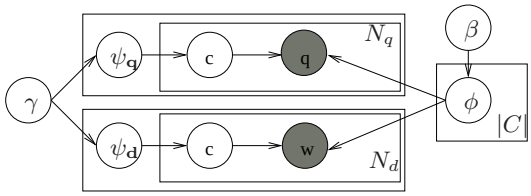


Figure 2: Graphical notation for the PairModel (Averaged), query and document do not share the concept distribution.

In another variant, we run PairModel without any word-pairs. As stated earlier, we add dummy concepts of the form "word:word" for Out-of-Dictionary words. In this variant, we use dummy concept for each word. Thus, the model reduces to a unigram language model whose parameters, shared by queries and documents, are estimated on the concatenation of queries and documents. We refer to this model as PairModel(–Pairs) in the experimental section (Sec. 5.2).

Finally, we can use posterior regularization to constrain the concepts sampled for a query and document to match with each other. But [15], shows that posterior regularization brings only a minor improvement so we skip this variant in our experimental section.

3.3 Document Ranking

We use the LM framework to rank documents. Following [36, 14], we score the relevance of documents given a query using a mixture model

$$P(\mathbf{q}|\mathbf{d}) = \prod_{q\in\mathbf{q}} \Big(\lambda_1 P_{mx}(q|\mathbf{d}) + (1-\lambda_1)P(q|\mathcal{C})\Big) \quad \text{where}$$

$$P_{mx}(q|\mathbf{d}) = \lambda_2 P_{ml}(q|\mathbf{d}) + (1-\lambda_2)P_{sys}(q|\mathbf{d}) \tag{5}$$

where $P_{ml}(q|\mathbf{d})$ is the maximum likelihood estimate of the query word q in the document and $P(q|\mathcal{C})$ is the unigram probability of the query word in the entire collection. And $P_{sys}(q|\mathbf{d})$ is the probability of q in the given document whose value varies among different ranking models. In PairModel, $P_{sys}(p|d)$ is estimated as

$$P_{sys}(q|\mathbf{d}) = \sum_{c|q\in c} P(q|c)P(c|\mathbf{d}) \tag{6}$$

where $c|q \in c$ denotes the set of all concepts that have the query word q as one of them.

4. BILINGUAL PAIRED TOPIC MODEL

Bilingual Paired Topic Model (BPTM) uses training data from an assisting language to improve ranking of documents in a search language. We assume that the assisting language (English) has richer resources, such as larger amounts of click-through data, document and query collections, that are useful for building a better Web search ranker. For concreteness, we consider the task of ranking French documents using English as an assisting language.

In what follows, we will first describe the way aligned query-document collections are modeled by BPTM. Then we extend BPTM to model unaligned query-document collections. We assume that we have aligned query-document collections (which are extracted from click-through logs) in both languages $\big\{(\mathbf{q}_i^e, \mathbf{d}_i^e), (\mathbf{q}_j^f, \mathbf{d}_j^f)\big\}$ for i=$1 \ldots m$ and $j = 1 \ldots n$. The queries and documents across different languages are assumed to be comparable (*e.g.*, from the same time period) but not necessarily translations of each other.

The underlying idea behind BPTM is to jointly model bilingual collections such that useful knowledge can be transferred across languages. The joint modeling is motivated by the previous study [20], which shows that training a ranker on a bilingual data is more effective than than learning a ranker in English and transferring it to French. Note that simply replacing word-pairs of PairModel with bilingual word translations does not lead to a model that is different from a document's unigram model because the model cannot capture any inter-word dependencies in documents of the same language. For example, the probability of a word-pair ("camera:caméra") in a particular French document depends only on the French word 'caméra' and is independent of other French words. Therefore, in addition to the use of bilingual translations, BPTM also uses bilingual topics. These bilingual topics map the queries and documents in different languages into a common lower dimensional semantic space. In that sense, BPTM bears resemblance to JointLDA [21] and BLTM [15] but there are some key differences. First of all, unlike BLTM where a topic is a distribution over words, a topic in BPTM is a distribution over bilingual word translation pairs (or concepts).[5] Second, unlike JointLDA which does not make use of alignment information between query and document, BPTM assumes that a query and its paired document share the same topic distribution.

We now describe BPTM more formally. We assume that an aligned query-document pair share the topic distribution $\theta_{(\mathbf{q},\mathbf{d})}$ which is a multinomial distribution over T bilingual topics and is drawn from a Dirichlet distribution with symmetric prior (α). Each bilingual topic (ψ_k) is a multinomial distribution over concepts. Finally, depending on the language of the query-document pair, these concepts generate words. Given a concept and the language, there is only one option for choosing the word and it is deterministic. To explain words that are not present in the bilingual dictionary, we add dummy translations for both Out-of-Dictionary English and French words. A dummy translation for an English word can only generate a English word and can not generate French words and vice versa. Jagarlamudi and Daumé III [21] showed that these dummy translations lead to deficient probability models and suggested adding a dependency link between the document language and the concept variable. Following their argument, let $\mathbb{I}(c_i, l_d)$ denote a binary indicator variable that denotes whether the concept c_i can generate a word from the language l_d, then the generative process of BPTM is as follows. The corresponding graphical notation is shown in Fig. 3.

1. For each topic $k = 1 \ldots T$, choose $\psi_k \sim \text{Dir}(\gamma)$
2. For each aligned query-document pair
   (a) Choose $l_{(\mathbf{q},\mathbf{d})} \sim \text{Bin}(\frac{1}{2})$.
   (b) Choose $\theta_{(\mathbf{q},\mathbf{d})} \sim \text{Dir}(\alpha)$
   (c) For each document term $i = 1 \ldots N_d$
      - Select a topic $z_i \sim \text{Mult}(\theta_{(\mathbf{q},\mathbf{d})})$
      - Select a concept $c_i \sim \text{Mult}(\psi_{z_i}) \cdot \mathbb{I}(c_i, l_{(\mathbf{q},\mathbf{d})})$

[5]We advise the reader to distinguish that a concept in BPTM refers to a word translation pair while it refers to a semantically related word pair in the PairModel.

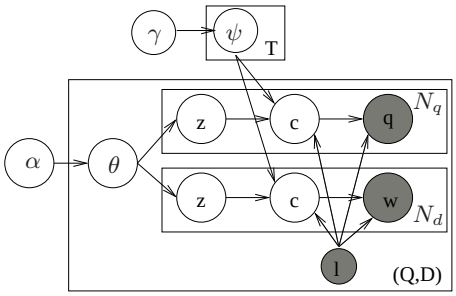


Figure 3: Graphical notation for the Bilingual Paired Topic Model (BPTM).

- Select a word $w_i \sim P(w_i|c_i, l_{(\mathbf{q},\mathbf{d})})$

(d) For each query term $i = 1 \cdots N_q$

- Select a topic $z_i \sim \text{Multi}(\theta_{(\mathbf{q},\mathbf{d})})$
- Select a concept $c_i \sim \text{Mult}(\psi_{z_i}) \cdot \mathbb{I}(c_i, l_{(\mathbf{q},\mathbf{d})})$
- Select a word $q_i \sim P(q_i|c_i, l_{(\mathbf{q},\mathbf{d})})$

To understand the generative process, consider generating an English aligned query-document pair. First, choose a topic mixture say 70% of education and 30% of sports. Then, we generate the query term by term. For the first term, choose a topic, say 'education', and then choose a concept from the education topic, let it be "university:université". We then generate the first English query term 'university' from the concept. If we were to generate the French query then we would have chosen 'université' instead. This process repeats until all words in the query and its aligned document are generated. Like in PairModel, BPTM can be trivially extended to generating queries and documents that are not aligned. In such a case, the generative process is limited to either a query or a document.

In BPTM, we use bilingual word-pairs which are translation equivalents where as in the PairModel we use monolingual word-pairs that are semantically related. Because of this distinction, BPTM doesn't inherit all the advantages of the PairModel (as evidenced in Sec. 5.3). We can use semantically related bilingual word-pairs but it is very difficult to obtain such word-pairs.

4.1 Inference

Again we use EM algorithm to derive the parameter update equations. In the E-step, we estimate the posterior probabilities and then use them to update the MAP estimates in the M-step. The E-step involves estimating:

$$P(c|z,l) = \frac{P(c|\psi_z)\mathbb{I}(c,l)}{\sum_{c'} P(c'|\psi_z)\mathbb{I}(c',l)} \quad (7)$$

$$P(c,z|w,\mathbf{d},l) = \frac{P(z|\theta_{(\mathbf{q},\mathbf{d})})P(c|z,l)}{\sum_{z',c'} P(z'|\theta_{(\mathbf{q},\mathbf{d})})P(c'|z',l)} \quad (8)$$

To compute the posterior probability for a query word, simply replace the pair $w, \mathbf{d}$ with $q, \mathbf{q}$ in Eq. 8.

As before, let $n(q,\mathbf{q})$ denote the frequency of the query word q in the query $\mathbf{q}$ and, similarly, $n(w,\mathbf{d})$ be the frequency of the word w in the document $\mathbf{d}$. Moreover, let $N(c,z,q,\mathbf{q},l) = n(q,\mathbf{q})P(c,z|q,(\mathbf{q},\mathbf{d}),l)$ and $N(c,z,w,\mathbf{d},l) = n(w,\mathbf{d})P(c,z|w,(\mathbf{q},\mathbf{d}),l)$ then the M-step involves:

$$P(z|\theta_{(\mathbf{q},\mathbf{d})}) \propto (\alpha - 1) + \sum_{q,c} N(c,z,q,\mathbf{q},l) + \sum_{w,c} N(c,z,w,\mathbf{d},l)$$

$$P(c|\psi_z) \propto (\gamma - 1) + \sum_{q|q\in c} N(c,z,q,\mathbf{q},l) + \sum_{w|w\in c} N(c,z,w,\mathbf{w},l)$$

where $w \in c$ denotes that the word w is one of the two words represented by c. Given a concept c and the language l, there is only one option for choosing the word and hence $P(w|c,l)$ need not be estimated.

		De	Fr	En	En(all)
Test	# of queries	4K	5K	5.3K	6K
	# docs	44K	54K	49K	190K
Training #(**q**, **d**) pairs		128K	133K	2.1M	32M
Vocab Size		110K	76K	500K	500K
Dictionary Size		100K	82K	489K	489K

Table 3: Training and test data statistics.

4.2 Document Ranking

We first train BPTM model on the bilingual query-document collections. Then, we keep the topic-concept distributions (ψ) fixed and fold in the test documents to get their topic distributions (θ). We use the same mixture model described in Sec. 3.3 to rank the documents. Finally, the $P_{sys}(q|\mathbf{d})$ required to compute Eq. 5 is computed as follows:

$$P_{sys}(q|\mathbf{d}) = \sum_{c,z} P(q|c)P(c|z)P(z|\mathbf{d}) \quad (9)$$

5. EXPERIMENTS

In this section, we evaluate our models against state-of-the-art baseline systems in three languages: English, German and French. We first compare PairModel with different baselines on monolingual Web Search task, in Sec. 5.2, and then move on to evaluating BPTM in Sec. 5.3.

5.1 Experimental Setup

Since our ranking models rely on click-through data, we can not evaluate our models on standard TREC data sets. Therefore, following previous studies of using user log for Web search ranking [1, 14, 15], we use proprietary datasets that have been developed for building a commercial Web search engine and compare our models with state-of-the-art ranking models that are originally developed for TREC data sets [27] as well as approaches that use click-through data [14, 15]. We evaluate our models on data sets collected in three different languages, English, French, and German. The queries in all the three languages are sampled from a year of search engine query logs. 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, around 20% are misspelled queries, and around 20% are navigational queries and 10% are transactional queries, *etc.*. Second, for each query, we collect Web documents to be judged by issuing the query to Bing. Subsequently, query-document pairs are manually judged on a scale of 0 to 4, with 0 being totally irrelevant and 4 being most relevant. For all the three data sets, we filtered the queries that have less than ten relevant

	English			German			French		
	ndcg@1	ndcg@3	ndcg@10	ndcg@1	ndcg@3	ndcg@10	ndcg@1	ndcg@3	ndcg@10
JMLM	28.70	37.54	48.66	34.59	43.79	56.20	38.22	46.36	60.57
RM	31.30	40.21	50.82	36.18	45.48	57.84	40.31	48.60	62.51
WTM	31.79	40.77	51.31	36.54	46.26	*58.63*	40.09	*48.89*	**63.06**
BLTM	34.70	43.16	53.03	37.26	46.50	58.33	40.03	48.36	62.49
PairModel(–Pairs)	*34.74*	*43.46*	*53.17*	35.74	45.05	57.45	38.94	47.54	61.82
PairModel(Averaged)	34.64	43.40	53.10	*38.28*	*47.02*	*58.63*	*40.34*	48.42	62.30
PairModel	**35.02***	**43.46***	**53.26***	**39.55***	**47.64***	**58.98**	**40.94***	**48.95**	*62.58*
Δ over best baseline	+0.32	+0.30	+0.23	+2.29	+1.14	+0.35	+0.63	+0.06	-0.48

Table 4: Comparison of PairModel with state-of-the-art baseline systems. In each column, the best system is bolded, the second best system is italicized and * denotes statistical significance improvement compared to the best baseline measured by t-test at p-value of 0.05.

documents. For most of the experiments, we include documents with at least one clicked query. To evaluate the effectiveness of different systems in leveraging unaligned documents, we prepare a separate English data set, referred to as En(all), by including queries and documents that do not have any user click information. For all the data sets, the click information is extracted from query logs using a process similar to [17]. The training and test set statistics are shown in Table 3. Notice that the number of test queries is much larger than the typical 50 queries used in standard TREC evaluation settings.

We use the following baseline systems. **JMLM**: This is the unigram language model of the document with Jelinek-Mercer smoothing. **RM**: Relevance model [27], is one of the state-of-the-art PRF methods developed for the LM framework. The number of expansion terms and the number of top-ranked documents used for query expansion are optimized via cross validation. **WTM**: The third baseline is a word based translation model, a variant of statistical translation model, as implemented in [14]. We use same dictionaries for WTM and PairModel. **BLTM**: The final baseline is the Bilingual Topic Model [15].

The interpolation parameters in all the systems, λ_1 and λ_2, are estimated using 2-fold cross validation. We split the data into two halves, find the best interpolation parameters on one half using grid search and use them for testing on the other half. We report averaged results on both the splits. Finally, we use Normalized Cumulative Discounted Gain (ndcg) [24] to evaluate the ranking against the human judgements. We report ndcg at ranks 1, 3, and 10.

5.2 Monolingual Web Search Results

In this section we compare PairModel with the baseline systems. PairModel requires a monolingual dictionary of word-pairs. For English, this dictionary is learnt from a year worth of query logs (as in [14]). For German and French languages, we learn the dictionaries from the training data of aligned query-document pairs which is much smaller compared to the data used for English. For all the three languages, we run IBM Model 1 on the aligned query-document pairs and then filter all the word-pairs with conditional translation probability less than a threshold of 0.005. For quicker I/O, we keep only those word-pairs in which both the words are seen in our vocabulary and the resulting dictionary statistics are also shown in Table 3. We use the dictionary to fit PairModel and its variants and estimate the parameter values. After learning the parameter values, the documents are ranked using Eqs. 5 and 6.

Table 4 shows the results of different systems on English, German and French data sets. Across all the languages, JMLM smoothing does poorly because of the lexical gap problem. As expected, RM and WTM outperform JMLM demonstrating the effectiveness of addressing the term mismatch problem using co-occurrence statistics. BLTM performs significantly better than WTM on English but is indistinguishable on German and French. This is largely due to the size of training data we used to learn BLTM, which is considerably larger in English than in other languages as shown in Table 3. The order of the baseline systems is consistent with what has been reported in the previous literature [14, 15].

Recall that PairModel without word-pairs (PairModel(–Pairs) in Sec. 3.2) reduces to a unigram language model whose parameters are estimated from both the document title and the queries for which the document is clicked. As shown in Table 4, the performance of PairModel(–Pairs) is not consistent across different languages. It outperforms WTM and BLTM in English but underperforms in German and French. We speculate that this result is due to the fact that, on average, English documents have rich click data than German and French documents.

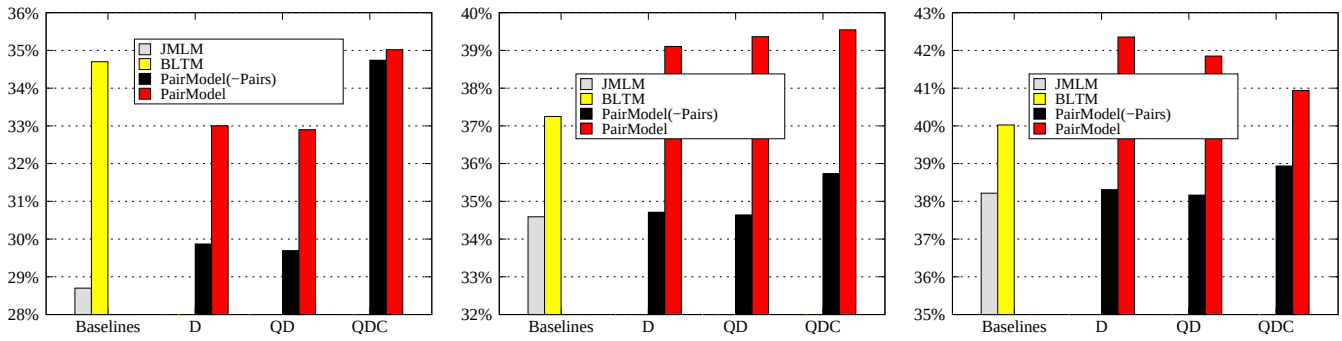
On the other hand, using word-pairs but neglecting the alignment between query-document pairs, as indicated by PairModel(Averaged) row of Table 4, is indistinguishable from PairModel(–Pairs) model on English, but is significantly better in other languages. This shows the effectiveness of using word-pairs especially when the click data is smaller. This model compares favorably to the state-of-the-art baselines (*i.e.*, WTM and BLTM).

Finally, PairModel which uses both the word-pairs and the alignment information between query and document pairs outperforms not only the variants, but also the baseline systems in most cases. The last row of the Table 4 shows the improvement of this model compared to the best of the four baseline systems. It achieves a significant improvement of 2.29 points at ndcg@1 for German. Overall, PairModel performed best and one of its variants performed second best.

Although PairModel and WTM use same set of word-pairs, the significantly better performance of PairModel indicates that our model seems to have the capability to adapt the translation probabilities to the data set. To verify this, we ran a separate experiment where we (re-)estimate the translation probabilities for the original pairs based on the

	English			German			French		
	ndcg@1	ndcg@3	ndcg@10	ndcg@1	ndcg@3	ndcg@10	ndcg@1	ndcg@3	ndcg@10
WTM	31.79	40.77	51.31	36.54	46.26	58.63	40.09	48.89	63.06
WTM (Adapted dict.)	**32.49**	**41.32**	**51.64**	**38.04**	**47.58**	**59.49**	**40.70**	**49.36**	**63.39**
Δ over original	+0.70	+0.55	+0.33	+1.50	+1.32	+0.86	+0.61	+0.47	+0.33

Table 5: Performance of WTM model with adapted translation dictionary.



(a) English data set (b) German data set (c) French data set

Figure 4: Performance (ndcg@1) of PairModel under different resource settings. JMLM uses only titles (D) while BLTM uses titles, queries, and click data (QDC).

posterior probabilities learnt by PairModel. Let c_{12} denote the word-pair $w_1 : w_2$ and let $\{c'|w \in c'\}$ denote the set of concepts that have w as on of their words, then

$$P(w_1|w_2) = \frac{\#(w_1, w_2)}{\#(w_2)} = \frac{\#(w_1 : w_2)}{\sum_{w'} \#(w' : w_2)} = \frac{N(c_{12})}{\sum_{c'|w_2 \in c'} N(c')}$$

where $N(c)$ denote the expected number of times this concept is assigned to any of the words and is given by:

$$N(c) = \sum_{(\mathbf{q},\mathbf{d})} \Big(\sum_{q \in \mathbf{q}\ \&\ q \in c} N(q, c, \mathbf{q}) + \sum_{w \in \mathbf{d}\ \&\ w \in c} N(w, c, \mathbf{d}) \Big)$$

$N(q, c, \mathbf{q})$ and $N(w, c, \mathbf{d})$ are as defined in Sec. 3.1. For every word-pair of the input dictionary, we re-estimate its translation probability and use this re-weighted dictionary as input to WTM. Table 5 shows the results with the re-estimated dictionary. This clearly confirms that our model learns to tune the translation probabilities for this data set.

5.2.1 Variation with the Data

To better understand the model, we test the effectiveness of PairModel with varying resources. Specifically, we trained the model on document titles only (D), document titles + queries but without click information (QD) and document titles + queries + click data (QDC). Fig. 4 shows the results. The x-axis marks the resource setting and the y-axis marks the ndcg@1 scores.

Except in French, performance of PairModel increases as we add queries and their alignment information. Moreover, PairModel achieves significantly higher improvements in resource poor situations, which is justifiable as exploiting pairwise relations is expected to be more useful when there is not enough evidence for a word on its own. Another interesting observation is that, in German and French, even with only documents PairModel beats the state-of-the-art BLTM model. In the QDC setting, the performance of BLTM is closer to that of PairModel in English than in other languages, probably due to the availability of relatively larger amounts of click-through data in English.

	English (All)		
	ndcg@1	ndcg@3	ndcg@10
JMLM	26.30	31.84	43.80
WTM	28.85	34.76	47.42
BLTM	31.24	36.85	**49.36**
PairModel(–Pairs)	30.30	35.61	47.44
PairModel(Averaged)	*32.25*	*37.10*	48.88
PairModel	**32.23**	**37.13**	*48.91*
Δ over best baseline	0.99	0.28	-0.45

Table 6: Performance of all the models after adding unaligned documents and queries (in English).

In the final monolingual experiment, we evaluate the effectiveness of PairModel by adding unaligned queries and documents. Table 6 shows the results on this data set. As expected, this decreased the performance of all the systems, but notably PairModel is still able to better identify the relevant documents at rank 1. But the improvements decrease (compared to BLTM model) at higher ranks.

5.3 Bilingual Web Search Results

In this section, we report our bilingual results on German and French. Our aim is to test the effectiveness of word-pairs, in using assist language training data, to improve the accuracy in a search language and is *not* to build a competitive multilingual IR algorithm. So in this section we mainly resort to comparing with the same baselines and not with the state-of-the-art multilingual IR algorithms like [10, 11, 20].

We use the same data sets and the experimental setup as described in Sec. 5.1 and report results on Web Search task in French and German using English as the assisting

		German			French		
		ndcg@1	ndcg@3	ndcg@10	ndcg@1	ndcg@3	ndcg@10
Monolingual	BLTM	37.26	46.50	58.33	40.03	48.36	62.49
Bilingual	BPTM	37.38	46.58	58.48	40.85*	49.01*	62.73
	Δ over BLTM	+0.12	+0.08	+0.15	+0.82	+0.65	+0.24
	BPTM+PairModel	**39.66***	**47.73***	**59.04***	**42.25***	**50.01***	**63.42***
	Δ over BLTM	+2.40	+1.23	+0.71	+2.22	+1.65	+0.93

Table 7: Bilingual IR results on French and German. For comparison purposes, we have also shown the scores of best monolingual baseline system (BLTM). Again * denotes statistically significant improvement compared to BLTM as measured by t-test at p-value of 0.05.

language. For both these language pairs, we used statistical dictionaries learnt using Giza++ [28] on parallel data used for training a commercial MT system. We remove all translation pairs that have conditional translational probability less than 0.001 and keep only those pairs in which both the words are seen in our vocabularies. After the filtering, we are left with 181K and 269K word-pairs in En–De and En–Fr language pairs.

As explained in Sec. 1, PairModel exploits evidence from the search language while BPTM uses evidence from the assisting language so we also report the results of a combined model, 'BPTM+PairModel'. We use query wise interpolation to combine these models, *i.e.*, we first score the documents independently using the two models (described in sections 3.3 and 4.2) and then interpolate the document scores using a variant of Powell Search algorithm [16]. Table 7 shows the results.

We remind the reader that the major difference between BLTM and BPTM is that BPTM models query-document collections of both search and assisting languages while BLTM models only search language query-document collection. We expect the effectiveness of BPTM to depend on the coverage and quality of the bilingual word translations. Comparing BPTM with BLTM, we observe that BPTM is able to get 0.82 and 0.65 ndcg improvements at ranks 1 and 3 in French[6] but it gives almost the same performance in German. This is justifiable because of the poor quality dictionary for En–De language pair. The rich morphology and the compound word phenomenon of German limits the coverage of the En–De dictionary. The combined model, as denoted by 'BPTM+PairModel', improves over both BPTM and PairModel and the improvements are higher in French (again because of the high quality dictionary). While the reader might think that the combination of BLTM and PairModel can perform as good as BPTM+PairModel, we argue that BPTM is a better choice than BLTM since it uses assisting language query-document collection and hence is more likely to be complementary than BLTM.

6. DISCUSSION

In this paper, we proposed two models that explicitly use word-pairs to model aligned and unaligned query document collections. Our models combine the strengths of the previous click-through based approaches to lexical gap and, hence, achieve statistically significant improvements compared to the state-of-the-art baseline systems. The improvements are especially promising when the click data is small, *i.e.*, for emerging languages such as French and German. We have also observed that, significant improvements can be obtained by using a resource rich assisting language which presumably has more click-through data. To this end, our second model uses bilingual dictionaries to map queries and documents of both the languages into a common lower dimensional sub-space. Hence, it can use training data from a data rich assisting language to improve ranking in a resource poor language. Experimental analysis indicates that the performance of this model depends on the quality and the coverage of bilingual dictionaries. Since PairModel and BPTM models exploit different collections, search and assisting language resources respectively, they are complimentary to each other and hence can be combined. The combined model gave superior results compared to the individual models and also the baseline systems.

We have used word-pairs derived from click-through data for the monolingual model. Though, in principle, our model does not depend on the source of these word-pairs, its effectiveness for the Web search task will probably depend on the source. Moreover, for BPTM, we used bilingual word-pairs that are likely to be translations of each other and, again, such word-pairs may not be ideal for the Web search task. In this paper, our primarily goal is to show the utility of modeling word-pairs and is not to evaluate the suitability of the word-pairs – although the latter task is equally important. In future, we would like to study the effect of different sources on the final performance. Moreover we would like to move from word-pairs to small bilingual clusters, so that the model can effectively combine evidence from query-document pairs with-in and across languages.

[6]The improvements in French are statistically significant as measured by the t-test with p-value of 0.05

7. REFERENCES

[1] E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *SIGIR '06*, pages 19–26, New York, NY, USA, 2006. ACM.

[2] A. Asuncion, M. Welling, P. Smyth, and Y. W. Teh. On smoothing and inference for topic models. UAI '09, pages 27–34, Arlington, Virginia, United States, 2009. AUAI Press.

[3] R. Baeza-Yates and A. Tiberi. Extracting semantic relations from query logs. In *Proceedings of the 13th ACM SIGKDD*, KDD '07, pages 76–85, New York, NY, USA, 2007. ACM.

[4] J. Bai, J.-Y. Nie, G. Cao, and H. Bouchard. Using query contexts in information retrieval. In *Proceedings of the SIGIR '07*, pages 15–22, New York, NY, USA, 2007. ACM.

[5] A. Berger and J. Lafferty. Information retrieval as statistical translation. SIGIR '99, pages 222–229, New York, NY, USA, 1999. ACM.

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Maching Learning Research*, 3:993–1022, 2003.

[7] J. Boyd-Graber and D. M. Blei. Multilingual topic models for unaligned text. In *Uncertainty in Artificial Intelligence*, 2009.

[8] P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer. The mathematics of statistical machine translation: parameter estimation. *Comput. Linguist.*, 19:263–311, June 1993.

[9] G. Cao, J.-Y. Nie, J. Gao, and S. Robertson. Selecting good expansion terms for pseudo-relevance feedback. SIGIR '08, pages 243–250, New York, NY, USA, 2008. ACM.

[10] M. K. Chinnakotla, K. Raman, and P. Bhattacharyya. Multilingual PRF: english lends a helping hand. In *SIGIR '10*, pages 659–666, NY, USA, 2010. ACM.

[11] M. K. Chinnakotla, K. Raman, and P. Bhattacharyya. Multilingual pseudo-relevance feedback: performance study of assisting languages. In *ACL '10:*, pages 1346–1356, Morristown, NJ, USA, 2010.

[12] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Scociety, Series B*, 39(1):1–38, 1977.

[13] S. T. Dumais. Latent semantic analysis. *Annual Review of Information Science and Technology*, 38(1):188–230, 2004.

[14] J. Gao, X. He, and J.-Y. Nie. Clickthrough-based translation models for web search: from word models to phrase models. In *Proceedings of the CIKM '10*, pages 1139–1148, New York, NY, USA, 2010. ACM.

[15] J. Gao, K. Toutanova, and W.-t. Yih. Clickthrough-based latent semantic models for web search. In *Proceedings of the SIGIR '11*, pages 675–684, New York, NY, USA, 2011. ACM.

[16] J. Gao, Q. Wu, C. Burges, K. Svore, Y. Su, N. Khan, S. Shah, and H. Zhou. Model adaptation via model interpolation and boosting for web search ranking. In *EMNLP '09*, pages 505–513, NJ, USA, 2009.

[17] J. Gao, W. Yuan, X. Li, K. Deng, and J.-Y. Nie. Smoothing clickthrough data for web search ranking. SIGIR '09, pages 355–362, New York, NY, USA, 2009. ACM.

[18] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the SIGIR '99*, pages 50–57, New York, NY, USA, 1999. ACM.

[19] J. Huang, J. Gao, J. Miao, X. Li, K. Wang, F. Behr, and C. L. Giles. Exploring web scale language models for search query processing. In *Proceedings of the WWW '10*, pages 451–460, New York, NY, USA, 2010. ACM.

[20] J. Jagarlamudi and P. Bennett. Fractional similarity: Cross-lingual feature selection for search. In *Proceedings of the ECIR '11*. Springer, 2011.

[21] J. Jagarlamudi and H. Daumé III. Extracting multilingual topics from unaligned comparable corpora. In *Proceedings of the ECIR '10*, volume 5993, pages 444–456, Milton Keynes, UK, 2010. Springer.

[22] J. Jagarlamudi and J. Gao. Modeling click-through based word-pairs for web search. In *Proceedings of the WWW '12 Companion*, pages 537–538, New York, NY, USA, 2012. ACM.

[23] J. Jagarlamudi, R. Udupa, H. Daumé III, and A. Bhole. Improving bilingual projections via sparse covariance matrices. In *Proceedings of the EMNLP '11*, pages 930–940, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics.

[24] K. Järvelin and J. Kekäläinen. IR evaluation methods for retrieving highly relevant documents. In *SIGIR '00*, pages 41–48, New York, NY, USA, 2000. ACM.

[25] H. Z. Ji-Rong Wen, Jian-Yun Nie. Query clustering using user logs. *ACM Trans. Inf. Syst.*, 20:59–81, January 2002.

[26] R. Jin, A. G. Hauptmann, and C. X. Zhai. Title language model for information retrieval. SIGIR '02, pages 42–48, New York, NY, USA, 2002. ACM.

[27] V. Lavrenko and W. B. Croft. Relevance based language models. In *Proceedings of the SIGIR '01*, pages 120–127, New York, NY, USA, 2001. ACM.

[28] F. J. Och and H. Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.

[29] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. SIGIR '98, pages 275–281, New York, NY, USA, 1998. ACM.

[30] F. Song and W. B. Croft. A general language model for information retrieval. In *Proceedings of the CIKM '99*, pages 316–321, New York, NY, USA, 1999. ACM.

[31] M. Srikanth and R. Srihari. Biterm language models for document retrieval. In *Proceedings of the SIGIR '02*, pages 425–426, New York, NY, USA, 2002. ACM.

[32] K. Wang, X. Li, and J. Gao. Multi-style language model for web scale information retrieval. In *Proceeding of the SIGIR '10*, pages 467–474, New York, NY, USA, 2010. ACM.

[33] X. Wei and W. B. Croft. Lda-based document models for ad-hoc retrieval. In *SIGIR '06*, pages 178–185, New York, NY, USA, 2006. ACM.

[34] X. Wei and W. B. Croft. Modeling term associations for ad-hoc retrieval performance within language modeling framework. ECIR'07, pages 52–63, Berlin, Heidelberg, 2007. Springer-Verlag.

[35] J. Xu and W. B. Croft. Query expansion using local and global document analysis. In *Research and Development in Information Retrieval*, SIGIR '96, pages 4–11, New York, NY, USA, 1996. ACM.

[36] X. Xue, J. Jeon, and W. B. Croft. Retrieval models for question and answer archives. SIGIR '08, pages 475–482, New York, NY, USA, 2008. ACM.

[37] C. Zhai and J. Lafferty. Model-based feedback in the kl-divergence retrieval model. In *Proceedings of the CIKM '01*, 2001.

[38] D. Zhang, Q. Mei, and C. Zhai. Cross-lingual latent topic extraction. In *Proceedings of the ACL '10*, pages 1128–1137, Stroudsburg, PA, USA, 2010.