Clickthrough-Based Translation Models for Web Search: from Word Models to Phrase Models

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

Xiaodong He
Microsoft Research
One Microsoft Way
Redmond, WA 98052 USA
xiaohe@microsoft.com

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

ABSTRACT

Web search is challenging partly due to the fact that search queries and Web documents use different language styles and vocabularies. This paper provides a quantitative analysis of the language discrepancy issue, and explores the use of clickthrough data to bridge documents and queries. We assume that a query is parallel to the titles of documents clicked on for that query. Two translation models are trained and integrated into retrieval models: A word-based translation model that learns the translation probability between single words, and a phrase-based translation model that learns the translation probability between multi-term phrases. Experiments are carried out on a real world data set. The results show that the retrieval systems that use the translation models outperform significantly the systems that do not. The paper also demonstrates that standard statistical machine translation techniques such as word alignment, bilingual phrase extraction, and phrase-based decoding, can be adapted for building a better Web document retrieval system.

Categories and Subject Descriptors

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

General Terms

Algorithms, Experimentation

Keywords

Clickthrough Data, Translation Model, Language Model, PLSA, Linear Ranking Model, Web Search

1. INTRODUCTION

This paper is intended to address two fundamental issues in information retrieval (IR) by exploiting clickthrough data: *synonymy* and *polysemy*. Synonyms are different terms with identical or similar meanings, while polysemy means a term with multiple meanings.

These issues are particularly crucial for Web search. Synonyms lead to the so-called *lexical gap* problem in document retrieval: A query often contains terms that are different from, but related to, the terms in the relevant documents. The lexical gap is substantially bigger in Web search largely due to the fact that search queries and Web documents are composed by a large variety of people and in very different language styles [e.g., 18]. Polysemy, on the other hand, increases the ambiguity of a query, and often causes a search engine to retrieve many documents that do not match the user's intent. This problem is also amplified by the high diversity of Web documents and Web users. For example, depending on different users, the query term "titanic" may refer to the rock band from Norway, the 1997 Oscar-winning film, or the ocean liner infamous for sinking on her maiden voyage in 1912. Unfortunately, most popular IR methods developed in the research community, in spite of their state-of-the-art performance on benchmark datasets (e.g., the TREC collections), are based on bag-of-words and *exact* term matching schemes, and cannot deal with these issues effectively [10, 22, 37]. Therefore, the development of a retrieval system that goes beyond exact term matching and bag-of-words has been a long standing research topic, as we will review later.

The problem of synonyms has been addressed previously by creating relationships between terms in queries and in documents. Clickthrough data have been exploited for this purpose [3, 34]. However, relationships are created only between single words without taking into account the context, giving rise to an increasing problem of *noisy proliferation*, i.e., connecting a word to a large number of unrelated or weakly related words. In addition, *ad hoc* similarity measures are often used.

In this paper we propose a more principled method by extending the statistical translation based approach to IR, proposed by Berger and Lafferty [7]. We estimate the relevance of a document given a query according to how likely the query is *translated* from the *title text* of the document[1]. We explore the use of two translation models for IR. Both models are trained on a query-title aligned corpus, derived from one-year clickthrough data collected by a commercial Web search engine. The first model, called word-based translation model, learns the translation probability of a query term given a word in the title of a document. This model, however, does not address the problem of noisy proliferation.

The second model, called phrase translation model, learns the translation probability of a multi-term phrase in a query given a phrase in the title of a document. This model explicitly addresses the problem of noisy proliferation of translation relationships between single words. In theory, the phrase model, subsuming the word model as a special case, is more powerful because words in

CIKM'10, October 26-30, 2010, Toronto, Ontario, Canada.

[1] Notice that we use document titles rather than entire documents because titles are more similar to queries than body texts. We will give the empirical justification in Sections 3 and 4. For the same reason, in most of the retrieval experiments in this study, we use only the title texts of web documents for retrieval.

the relationships are considered with some context words. More precise translations can be determined for phrases than for words. This model is more capable of dealing with both the synonymy and the polysemy issues in a unified manner. It is thus reasonable to expect that using such phrase translation probabilities as ranking features is likely to improve the retrieval results, as we will show in our experiments.

Although several approaches have been proposed to determine relationships between the terms in queries and the terms in documents, most of them rely on a static measure of term similarity (e.g. cosine similarity) according to their co-occurrences across queries and documents. In statistical machine translation (SMT), it has been found that an EM process used to construct the translation model iteratively can significantly improve the quality of the model [9, 27]: A translation model obtained at a later iteration is usually better than the one at an earlier iteration, including the initial translation model corresponding to a static measure. An important reason for this is that some frequent words in one language can happen to co-occur often with many words in another language; yet the former are not necessarily good translation candidates for the latter. The iterative training process helps strengthen the true translation relations and weaken spurious ones. The situation we have is very similar: on the one hand, we have queries written by the users in some sub-language, and on the other hand, we have documents (or titles) written by the authors in another sub-language. Our goal is to detect possible relations between terms in the two sub-languages. This problem can be cast as a translation problem. The fact that the quality of translation models can be improved using the iterative training process strongly suggests that we could also obtain higher-quality term relationships between the two sub-languages with the same process. This is the very motivation to use principled translation models rather than static, *ad hoc*, similarity measures.

Our evaluation on a real world dataset shows that the retrieval systems that use the translation models outperform significantly the systems that do not use them. It is interesting to notice that our best retrieval system, which uses a linear ranking model to incorporate both the word-based and phrase-based translation models, shares a lot of similarities to the state-of-the-art SMT systems described in [23, 27, 28]. Thus, our work also demonstrates that standard SMT techniques such as word alignment, bilingual phrase extraction, and phrase-based decoding, can be adapted for building a better Web document retrieval system.

To the best of our knowledge, this is the first extensive and empirical study of learning word-based and phrase-based translation models using clickthrough data for Web search. Although clickthough data has been proved very effective for Web search [e.g., 2, 16, 33], click information is not available for many URLs, especially new and less popular URLs. Thus, another research goal of this study is to investigate how to learn title-query translation models from a small set of popular URLs that have rich click information, and apply the models to improve the retrieval of those URLs without click information.

In the reminder of the paper, Section 2 reviews previous research on dealing with the issues of synonymy and polysemy. Section 3 presents a large scale analysis of language differences between search queries and Web documents, which will motivate our research. Section 4 describes the data sets and evaluation methodology used in this study. Sections 5 and 6 describe in detail the word-based and phrase-based translation models, respectively. The experimental results are also presented wherever appropriate. Section 7 presents the conclusions.

2. RELATED WORK

Many strategies have been proposed to bridge the lexical gap between queries and documents at the lexical level or at the semantic level. One of the simplest and most effective strategies is automatic query expansion, where a query is refined by adding terms selected from (pseudo) relevant documents. A variety of heuristic and statistical techniques are used to select and (re-)weight the expansion terms [30, 35, 11, 5]. However, directly applying query expansion to a commercial Web search engine is challenging because the relevant documents of a query are not always available and generating pseudo relevant documents requires multi-stage retrieval, which is prohibitively expensive.

The latent variable models, such as LSA [12], PLSA [17], and LDA [8], take a different strategy. Different terms that occur in a similar context are grouped into the same latent semantic cluster. Thus, a query and a document, represented as vectors in the latent semantic space, can still have a high similarity even if they do not share any term. In this paper we will apply PLSA to word translation, and compare it with the other proposed translation models in the retrieval experiments.

Unlike latent variable models, the statistical translation based approach [7] does not map different terms into latent semantic clusters but learns translation relationships directly between a term in a document and a term in a query. A major challenge is the estimation of the translation models. The ideal training data would be a large amount of query-document pairs, in each of which the document is (judged as) relevant to the query. Due to the lack of such training data, [7] resorts to some synthetic query-document pairs, and [21] simply uses the title-document pairs as substitution for training. In this study we mine implicit relevance judgments from one-year clickthrough data, and generate a large amount of *real* query-title pairs for translation model training.

Clickthrough data have been used to determine relationships between terms in queries and in documents [3, 34]. However, relationships are only created between single words by using an *ad hoc* similarity measure. Translation models offer a way to exploit such relationships in a more principled manner, as we explained earlier.

Context information is crucial for detecting a particular sense of a polysemous query term. But most traditional retrieval models assume the occurrences of terms to be completely independent. Thus, research in this area has been focusing on capturing term dependencies. Early work tries to relax the independence assumption by including phrases, in addition to single terms, as indexing units [10, 32]. Phrases are defined by collocations (adjacency or proximity) and selected on the statistical ground, possibly with some syntactic knowledge. Unfortunately, the experiments did not provide a clear indication whether the retrieval effectiveness can be improved in this way.

Recently, within the framework of language models for IR, various approaches that go beyond unigrams have been proposed to capture some term dependencies, notably the bigram and trigram models [31], the dependence model [14], and the Markov Random Field model [25]. These models have shown benefit of capturing dependencies. However, they focus on the utilization of phrases as indexing units, rather than the relationships between phrases. [4] tried to determine such relationships using more complex term co-occurrences within documents. Our study tries to extract such relationships according to clickthrough data. Such relationships are expected to be more effective in bridging the gap between the query and document sub-languages. To our knowledge, this is the first such attempt using clickthrough data.

Dataset	Body	Anchor	Title	Query
#unigram	1.2B	60.3M	150M	251.5M
#bigram	11.7B	464.1M	1.1B	1.3B
#trigram	60.0B	1.4B	3.1B	3.1B
#4-gram	148.5B	2.3B	5.1B	4.6B
Total	1.3T	11.0B	257.2B	28.1B
Size on disk[#]	12.8T	183G	395G	393G

[#] N-gram entries as well as other statistics and model parameters are stored.

Table 1: Statistics of the Web *n*-gram language model collection (count cutoff = 0 for all models). These models will be released to the research community at [1].

In Section 6, we propose a new phrase-based query translation model that determines a probability distribution over "translations" of multi-word phrases from title to query. Our phrases are different from those defined in the previous work. Assuming that queries and documents are composed using two different "languages", our phrases can be viewed as *bilingual phrases* (or *bi-phrases* in short), which are consecutive multi-term sequences that can be translated from one language to another as units. As we will show later, the use of the bi-phrases not only bridges the lexical gap between queries and documents, but also reduces significantly the ambiguities in Web document retrieval.

3. COLLECTIONS OF SEARCH QUERIES AND WEB DOCUMENTS

Language differences between search queries and Web documents have often been assumed in previous studies without a quantitative evaluation [e.g., 2, 16, 33]. Following and extending the study in [18], we performed a large scale analysis of Web and query collections for the sake of quantifying the language discrepancy between search queries and Web documents.

Table 1 summarizes the Web *n*-gram model collection used in the analysis. The collection is built from the English Web documents, in the scale of trillions of tokens, served by a popular commercial Web search engine. The collection consists of several *n*-gram data sets built from different Web sources, including the different text fields from the Web documents such as body text, anchor texts, and titles, as well as search queries sampled from one-year worth of search query logs.

We then developed a set of language models, each on one *n*-gram dataset from a different data source. They are the standard word-based backoff *n*-gram models, where the *n*-gram probabilities are estimated using *maximum likelihood estimation* (MLE) with smoothing [26].

One way to quantify the language difference is to estimate how *certain* a language model trained on one data in one language (e.g., titles) predicts the data in another language (e.g., queries). We use perplexity to measure the certainty of the prediction. Lower perplexities mean higher certainties, and consequently, a higher similarity between the two languages.

Table 2 summarizes the perplexity results of language models trained on different data sources tested on a random sample of 733,147 queries from the search engine's May 2009 query log. The results suggest several conclusions. First, a higher order language model in general reduces perplexity, especially when moving beyond unigram models. This verifies the importance of capturing term dependencies. Second, as expected, the query n-gram language models are most predictive for the test queries, though they are from independent query log snapshots. Third, it is interesting to notice that although the body language models are trained on much larger amounts of data than the title and anchor models, the former lead to much higher perplexity values, indicating that both title and anchor texts are quantitatively much more similar to queries than body texts. We also notice that in the case of lower order (1-2) models, the title models have lower perplexities than the anchor models, but a higher order anchor model reduces the perplexity more. This suggests that title's vocabulary is more similar to that of queries than anchor texts whereas the ordering in the *n*-gram word structure captured by the anchor language models is more similar to the test queries than that by the title language models.

Order	Body	Anchor	Title	Query
Unigram	13242	4164	3633	1754
Bigram	5567	966	1420	289
Trigram	5381	740	1299	180
4-gram	5785	731	1382	168

Table 2: Perplexity results on test queries, using *n*-gram models with different orders, derived from different data sources.

In what follows, we will show the degree to which the language differences (measured in terms of perplexity) affect the performance of Web document retrieval.

4. DATA SETS AND EVALUATION METHODOLOGY

A Web document is composed of several *fields* of information. The field may be written either by the author of the Web page, such as body texts and titles, or by other authors, such as anchor texts and query clicks. The former sources are called *content fields* and the latter sources *popularity fields* [33].

The construction of content fields is straightforward. The construction of popularity fields is trickier because they have to be aggregated over information about the page from other authors or users. Popularity fields are highly repetitive for popular pages, and are empty or very short for new or less popular (or so-called *tail*) pages. In our study, the anchor text field is composed of the text of all incoming links to the page. The query click field is built from query session data, similar to [16]. The query click data consists of query sessions extracted from one year query log files of a commercial search engine. A query session consists of a user-issued query and a rank of documents, each of which may or may not be clicked by the user. The query click field of a document d is represented by a set of query-score pairs (q, $Score(d, q)$), where q is a unique query string and $Score(d, q)$ is a score assigned to that query. $Score(d, q)$ could be the number of times the document was clicked on for that query, but it is important to also consider the number of times the page has been shown to the user and the position in the ranked list at which the page was shown. Figure 1 shows a fragment of the query click field for the document *http://webmessenger.msn.com*, where $Score(d, q)$ is computed using the heuristic scoring function in [16].

The multi-field description of a document allows us to generate query-document pairs for translation model training. As shown in Figure 1, we can form a set of query-title pairs by aligning the title of the document (e.g., the title of the document *http://webmessenger.msn.com* is "*msn web messenger"*) to each unique query string in the query click field of the same document. In this study, we use titles, instead of anchor and body texts, to

msn web	0.6675749
Webmensseger	0.6621253
msn online	0.6403270
windows web messanger	0.6321526
talking to friends on msn	0.6130790
school msn	0.5994550
msn anywhere	0.5667575
web message msn com	0.5476839
msn messager	0.5313351
hotmail web chat	0.5231608
messenger web version	0.5013624
instant messager msn	0.4550409
browser based messenger	0.3814714
im messenger sign in	0.2997275
…	…

Figure 1: A fragment of the query click field for the page *http://webmessenger.msn.com* [16].

form training data for two reasons. First, titles are more similar to queries both in length and in vocabulary (Table 2), thus making word alignment and translation model training more effective. Second, as will be shown later (Table 3), for the pages with an empty query click field, the title field gives a very good single-field retrieval result on our test set, although it is much shorter than the anchor and body fields, and thus it can serve as a reasonable baseline in our experiments. Nevertheless, our method is not limited to the use of titles. It can be applied to other content fields later.

We evaluate the retrieval methods on a large scale real world data set, called the evaluation data set henceforth, containing 12,071 English queries sampled from one-year query log files of a commercial search engine. On average, each query is associated with 185 Web documents (URLs). Each query-document pair has a relevance label. The label is human generated and is on a 5-level relevance scale, 0 to 4, with 4 meaning document d is the most relevant to query q and 0 meaning d is not relevant to q. All the retrieval models used in this study (i.e., BM25, language models and linear ranking models) contain free parameters that must be estimated empirically by trial and error. Therefore, we used 2-fold cross validation: A set of results on one half of the data is obtained using the parameter settings optimized on the other half, and the global retrieval results are combined from those of the two sets.

The performance of all the retrieval models is measured by mean *Normalized Discounted Cumulative Gain* (NDCG) [19]. We report NDCG scores at truncation levels 1, 3, and 10. We also perform a significance test, i.e., a t-test with a significance level of 0.05. A significant difference should be read as significant at the 95% level.

Table 3 reports the results of a set of BM25 models, each using a single content or popularity field. This is aimed at evaluating the impact of each single field on the retrieval effectiveness. The retrieval results are more or less consistent with the perplexity results in Table 2. The field that is more similar to search queries gives a better NDCG score. Most notable is that the body field, though much longer than the title and anchor fields, gives the worst retrieval results due to the substantial language discrepancy from queries. The anchor field is slightly better than the title field because the anchor field is on average much longer, though in Table 2 the anchor unigram model shows higher a perplexity value than the title unigram model. Therefore it would be interesting

Field	NDCG@1	NDCG@3	NDCG@10
Body	0.2798	0.3121	0.3858
Title	0.3181	0.3413	0.4045
Anchor	0.3245	0.3506	0.4117
Query click	N/A	N/A	N/A

Table 3: Ranking results of three BM25 models, each using a different single field to represent Web documents. The click field of a document in the evaluation data set is not valid.

to learn translation models from click-anchor pairs in addition to click-title pairs. We leave it to future work.

Some previous studies [e.g., 16, 33] show that the query click field, when it is valid, is the most effective for Web search. However, click information is unavailable for many URLs, especially new URLs and tail URLs, leaving their click fields invalid (i.e., the field is either empty or unreliable because of sparseness). In this study, we assume that each document contained in the evaluation data set is either a new URL or a tail URL, thus has no click information (i.e., its click field is invalid). Our research goal is to investigate how to learn title-query translation models from the popular URLs that have rich click information, and apply the models to improve the retrieval of those tail or new URLs. Thus, in our experiments, we use BM25 with the title field as baseline.

From one-year query session data, we were able to generate very large amounts of query-title pairs. For training translation models in this study, we used a randomly sampled subset of 82,834,648 pairs whose documents are popular and have rich click information. We then test the trained models in retrieving documents that have no click information. The empirical results will verify the effectiveness of our methods.

5. THE WORD-BASED TRANSLATION MODEL

Let $Q = q_1 \ldots q_J$ be a query and $D = w_1 \ldots w_I$ be the title of a document. The word-based translation model [7] assumes that both Q and D are bag of words, and that the translation probability of Q given D is computed as

$$P(Q|D) = \prod_{q \in Q} \sum_{w \in D} P(q|w)P(w|D). \quad (1)$$

Here $P(w|D)$ is the unigram probability of word w in D, and $P(q|w)$ is the probability of translating w into a query term q.

It is easy to verify that if we only allow a word to be translated into itself, Equation (1) is reduced to the simple exact term matching model. In general, the model allows us to translate w to other semantically related query terms by giving those other terms a nonzero probability.

5.1 Learning Translation Probabilities

This section describes two methods of estimating the word translation probability $P(q|w)$ in Equation (1) using the training data, i.e., the query-title pairs, denoted by $\{(Q_i, D_i), i = 1 \ldots N\}$, derived from the clickthrough data, as described in Section 4.

The first method follows the standard procedure of training statistical word alignment models proposed in [9]. Formally, we optimize the model parameters θ by maximizing the probability of generating queries from titles over the training data:

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

q	$P(q\|w)$	Q	$P(q\|w)$
titanic	0.56218	Vista	0.80575
ship	0.01383	Windows	0.05344
movie	0.01222	Download	0.00728
pictures	0.01211	ultimate	0.00571
sink	0.00697	xp	0.00355
facts	0.00689	microsoft	0.00342
photos	0.00533	bit	0.00286
rose	0.00447	compatible	0.00270
people	0.00441	premium	0.00244
survivors	0.00369	free	0.00211
w = titanic		w = vista	

q	$P(q\|w)$	q	$P(q\|w)$
everest	0.52826	pontiff	0.17288
mt	0.02672	pope	0.09831
mount	0.02117	playground	0.03729
deaths	0.00958	wally	0.03053
person	0.00598	bartlett	0.03051
summit	0.00503	current	0.02712
climbing	0.00454	quantum	0.02373
cost	0.00446	wayne	0.02372
visit	0.00441	john	0.02034
height	0.00397	stewart	0.02031
w = everest		w = pontiff	

Figure 2: Sample word translation probabilities after EM training on the query-title pairs.

where $P(Q|D,\theta)$ takes the form of IBM Model 1 [7] as

$$P(Q|D,\theta) = \frac{\varepsilon}{(I+1)^J}\prod_{q\in Q}\sum_{w\in D} P(q|w,\theta). \quad (3)$$

where ε is a constant, J is the length of Q, and I is the length of title D. To find the optimal word translation probabilities of Model 1, we used the EM algorithm [13], running for only 3 iterations over the training data as a means to avoid overfitting. The details of the training process can be found in [9]. A sample of the resulting translation probabilities is shown in Figure 2, where a title word is shown together with the ten most probable query terms that it will translate according to the model.

The second method uses a heuristic model, inspired by [27]. This model is considerably simpler and easier to estimate. It does not require learning word alignments, but approximates $P(q|w,\theta)$ by a variant of the Dice coefficient:

$$P(q|w,\theta) \propto \frac{C(q,w)}{C(w)}, \quad (4)$$

where $C(q,w)$ is the number of query-title pairs (Q,D) in the training data, where q occurs in the query part and w occurs in the title part, and $C(w)$ is the number of query-title pairs where w occurs in the title part.

5.2 Ranking Documents

The word-based translation model of Equation (1) needs to be *smoothed* before it can be applied to document ranking. We follow [7] to define a smoothed model as

$$P_s(Q|D) = \prod_{q\in Q} P_s(q|D). \quad (5)$$

Here, $P_s(q|D)$ is a linear interpolation of a background unigram model and a word-based translation model:

$$P_s(q|D) = \alpha P(q|C) + (1-\alpha)\sum_{w\in D} P(q|w)P(w|D). \quad (6)$$

where $\alpha \in [0,1]$ is the interpolation weight empirically tuned, $P(q|w)$ is the word-based translation model estimated using either of the two methods described in Section 5.1, and $P(q|C)$ and $P(w|D)$ are the unsmoothed background and document models, respectively, estimated using maximum likelihood estimation as

$$P(q|C) = \frac{C(q;C)}{|C|}, \text{and} \quad (7)$$

$$P(w|D) = \frac{C(w;D)}{|D|}, \quad (8)$$

where $C(q;C)$ and $C(w;D)$ are the counts of q in the collection and in the document, respectively; and $|C|$ and $|D|$ are the sizes of the collection and the document, respectively.

However, the model of Equations (5) and (6) still does not perform well in our retrieval experiments due to the low self-translation problem. This problem has also been studied in [36, 20, 24, 21]. Since the target and the source languages are the same, every word has some probability to translate into itself, i.e., $P(q=w|w) > 0$. On the one hand, low self-translation probabilities reduce retrieval performance by giving low weights to the matching terms. On the other hand, very high self-probabilities do not exploit the merits of the translation models.

Different approaches have been proposed to address the self-translation problem [36, 20, 24, 21]. These approaches assume that the self-translation probabilities estimated directly from data, e.g., using the methods described in Section 5.1, are not optimal for retrieval, and have demonstrated that significant improvements can be achieved by adjusting the probabilities. We compared these approaches in our experiments. The best performer is the one proposed by Xue et al. [36], where Equation (6) is revised as Equation (9) so as to explicitly adjust the self-translation probability by linearly mixing the translation based estimation and maximum likelihood estimation

$$P_s(q|D) = \alpha P(q|C) + (1-\alpha)P_{mx}(q|D), \text{ where} \quad (9)$$

$$P_{mx}(q|D) = \beta P(q|D) + (1-\beta)\sum_{w\in D} P(q|w)P(w|D).$$

Here, $\beta \in [0,1]$ is the tuning parameter, indicating how much the self-translation probability is adjusted. Notice that letting $\beta = 1$ in Equation (9) reduces the model to a unigram language model with Jelinek-Mercer smoothing [37]. $P(q|D)$ in Equation (9) is the unsmoothed document model, estimated by Equation (8). So we have $P(q|D) = 0, \text{for } q \notin D$.

5.3 Results

Table 4 shows the main document ranking results using word-based translation models, tested on the human-labeled evaluation dataset via 2-fold cross validation, as described in Section 4. Row 1 is the baseline model. Rows 2 to 5 are different versions of the word translation based retrieval model, parameterized by Equations (5) to (9). All these models achieve significantly better results than the baseline in Row 1. By setting $\beta = 1$ in Equation (9), the model in Row 2 is equivalent to a unigram language model with Jelinek-Mercer smoothing. Row 3 is the model where the word translation probabilities are assigned by Model 1 trained by the EM algorithm. Row 4 is similar to Row 3 except that the self-

#	Models	NDCG@1	NDCG@3	NDCG@10
1	**BM25**	0.3181	0.3413	0.4045
2	**WTM_M1 (β=1)**	0.3202	0.3445	0.4076
3	**WTM_M1**	0.3310	0.3566	0.4232
4	**WTM_M1 (β=0)**	0.3210	0.3512	0.4211
5	**WTM_H**	0.3296	0.3554	0.4215

Table 4: Ranking results on the evaluation data set, where only the title field of each document is used.

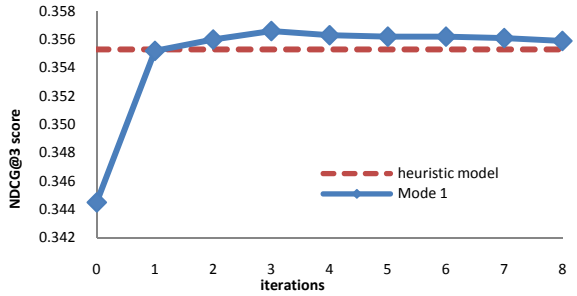

Figure 3: Variations in (top) NDCG@3 score as a function of the number of the EM iterations for word translation model training. Document ranking is performed by the word translation based retrieval model, parameterized by Equations (5) to (9).

translation probability is not adjusted, i.e., $\beta = 0$ in Equation (9). Row 5 is the model where the word translation probabilities are estimated by the heuristic model of Equation (4).

The results show that (1) as observed by other researchers, the simple unigram language model performs similarly to the classical probabilistic retrieval model BM25 (Row 1 vs. Row 2); (2) using word translation model trained on query-title pairs leads to statistically significant improvement (Row 3 vs. Row 2); (3) it is beneficial to boost the self-translation probabilities (Row 3 vs. Row 4 is statistically significant in NDCG@1 and NDCG@3); and (4) Model 1 outperforms the heuristic model with a small but statistically significant margin (Row 3 vs. Row 5). Analyzing the variation of the document retrieval performance as a function of the EM iterations in Model 1 training is instructive. As shown in Figure 3, after the first iteration, Model 1 achieves a slightly worse retrieval result than the heuristic model, but the second iteration of Model 1 gives a significantly better result.

6. THE PHRASE-BASED TRANSLATION MODEL

The phrase-based translation model is a generative model that translates a document title D into a query Q. Rather than translating single words in isolation, as in the word-based translation model, the phrase model translates sequences of words (i.e., phrases) in D into sequences of words in Q, thus incorporating contextual information. For example, we might learn that the phrase "stuffy nose" can be translated from "cold" with relatively high probability, even though neither of the individual word pairs (i.e., "stuffy"/"cold" and "nose"/"cold") might have a high word translation probability. We assume the following generative story: first the title D is broken into K non-empty word sequences $\mathbf{w}_1,...,\mathbf{w}_K$, then each is translated to a new non-empty word sequence $\mathbf{q}_1,...,\mathbf{q}_K$, finally these phrases are permuted and concatenated to form the query Q. Here $\mathbf{w}$ and $\mathbf{q}$ denote consecutive sequences of words.

D:	… cold home remedies …	*title*
S:	["cold", "home remedies"]	*segmentation*
T:	["stuffy nose", "home remedy"]	*translation*
M:	(1 → 2, 2→ 1)	*permutation*
Q:	"home remedy stuffy nose"	*query*

Figure 4: Example demonstrating the generative procedure behind the phrase-based translation model.

To formulate this generative process, let S denote the segmentation of D into K phrases $\mathbf{w}_{1,...,}\mathbf{w}_K$, and let T denote the K translation phrases $\mathbf{q}_1,\ldots,\mathbf{q}_K$ – we refer to these $(\mathbf{c}_i, \mathbf{q}_i)$ pairs as *bi-phrases*. Finally, let M denote a permutation of K elements representing the final reordering step. Figure 2 demonstrates the generative procedure.

Next let us place a probability distribution over rewrite pairs. Let $B(D, Q)$ denote the set of S, T, M triples that translate D into Q. If we assume a uniform probability over segmentations, then the phrase-based translation probability can be defined as:

$$P(Q|D) \propto \sum_{\substack{(S,T,M)\in \\ B(D,Q)}} P(T|D,S) \cdot P(M|D,S,T) \tag{10}$$

Then, we use the maximum approximation to the sum:

$$P(Q|D) \approx \max_{\substack{(S,T,M)\in \\ B(D,Q)}} P(T|D,S) \cdot P(M|D,S,T) \tag{11}$$

Although we have defined a generative model for translating titles to queries, our goal is not to generate new queries, but rather to provide scores over existing Q and D pairs that will be used to rank documents. However, the model cannot be used directly for document ranking because D and Q are often of very different lengths, leaving many words in D unaligned to any query term. This is the key difference between our task and the general natural language translation. As pointed out by Berger and Lafferty [7], document-query translation requires a *distillation* of the document, while translation of natural language tolerates little being thrown away.

Thus we restrict our attention to those *key title words* that form the distillation of the document, and assume that a query is translated only from the key title words. In this work, the key title words are identified via word alignment. Let $A = a_1...a_J$ be the "hidden" word alignment, which describes a mapping from a query term position j to a title word position a_j. We assume that the positions of the key title words are determined by the Viterbi alignment A^*, which can be obtained using Model 1 (or the heuristic model) as follows:

$$A^* = \arg\max_A P(Q, A|D) \tag{12}$$

$$= \arg\max_A \left\{ P(J|I) \prod_{j=1}^{J} P(q_j|w_{aj}) \right\} \tag{13}$$

$$= \left[\arg\max_{aj} P(q_j|w_{aj}) \right]_{j=1}^{J} \tag{14}$$

Given A^*, when scoring a given Q/D pair, we restrict our attention to those S, T, M triples that are consistent with A^*, which we denote as $B(C, Q, A^*)$. Here, consistency requires that if two words are aligned in A^*, then they must appear in the same bi-

	A	B	C	D	E	F
a	#					
d				#		
c			#			
f						#

a	A
adc	ABCD
d	D
dc	CD
dcf	CDEF
c	C
f	F

Figure 5: Toy example of (left) a word alignment between two strings "adcf" and "ABCDEF"; and (right) the bilingual phrases containing up to five words that are consistent with the word alignment

phrase ($\mathbf{w}_i$, $\mathbf{q}_i$). Once the word alignment is fixed, the final permutation is uniquely determined, so we can safely discard that factor. Thus we rewrite Equation (11) as

$$P(Q|D) \approx \max_{\substack{(S,T,M)\in \\ \mathcal{B}(D,Q,A^*)}} P(T|D,S) \quad (15)$$

For the sole remaining factor $P(T|D, S)$, we make the assumption that a segmented query $T = \mathbf{q}_1 \ldots \mathbf{q}_K$ is generated from left to right by translating each phrase $\mathbf{w}_1 \ldots \mathbf{w}_K$ independently:

$$P(T|D,S) = \prod_{k=1}^{K} P(\mathbf{q}_k|\mathbf{w}_k), \quad (16)$$

where $P(\mathbf{q}_k|\mathbf{w}_k)$ is a phrase translation probability, the estimation of which will be described in Section 6.1.

The phrase-based query translation probability $P(Q|D)$, defined by Equations (10) to (16), can be efficiently computed by using a dynamic programming approach, similar to the monotone decoding algorithm described in [22]. Let the quantity α_j be the total probability of a sequence of query phrases covering the first j query terms. $P(Q|D)$ can be calculated using the following recursion:

1. Initialization: $\alpha_0 = 1$ (17)
2. Induction: $\alpha_j = \sum_{j'<j, \mathbf{q}=q_{j'+1}\ldots q_j} \{\alpha_{j'} P(\mathbf{q}|\mathbf{w}_\mathbf{q})\}$ (18)
3. Total: $P(Q|D) = \alpha_J$ (19)

6.1 Learning Translation Probabilities

This section describes the way $P(\mathbf{q}|\mathbf{w}_\mathbf{q})$ is estimated. We follow a method commonly used in SMT [23, 27] to extract bilingual phrases and estimate their translation probabilities.

First, we learn two word translation models using the EM training of Model 1 on query-title pairs in two directions: One is from query to title and the other from title to query. We then perform Viterbi word alignment in each direction according to Equations (12) to (14). The two alignments are combined as follows: we start from the intersection of the two alignments, and gradually include more alignment links according to a set of heuristic rules described in [27]. Finally, the bilingual phrases that are consistent with the word alignment are extracted using the heuristics proposed in [27]. The maximum phrase length is five in our experiments. The toy example shown in Figure 5 illustrates the bilingual phrases we can generate by this process.

Given the collected bilingual phrases, the phrase translation probability is estimated using relative counts:

$$P(\mathbf{q}|\mathbf{w}) = \frac{N(\mathbf{w},\mathbf{q})}{N(\mathbf{w})} \quad (20)$$

q	$P(\mathbf{q}\|\mathbf{w})$	**q**	$P(\mathbf{q}\|\mathbf{w})$
titanic	0.43195	sierra vista	0.61717
rms titanic	0.03793	sv	0.02260
titanic sank	0.02114	vista	0.01678
titanic sinking	0.01695	sierra	0.01581
titanic survivors	0.01537	az	0.00417
titanic ship	0.01112	bella vista	0.00320
titanic sunk	0.00960	arizona	0.00223
titanic pictures	0.00593	dominoes sierra vista	0.00221
titanic exhibit	0.00540	dominos sierra vista	0.00221
ship titanic	0.00383	meadows	0.00029
w = rms titanic		**w** = sierra vista	

Figure 6: Sample phrase translation probabilities learned from the word-aligned query-title pairs.

where $N(\mathbf{w}, \mathbf{q})$ is the number of times that **w** is aligned to **q** in training data. The estimation of Equation (20) suffers the data sparseness problem. Therefore, we also estimate the so-called *lexical weight* [23] as a smoothed version of the phrase translation probability. Let $P(q|w)$ be the word translation probability described in Section 5.1, and A the word alignment between the query term position $i = 1 \ldots |\mathbf{q}|$ and the title word position $j = 1 \ldots |\mathbf{w}|$, then the lexical weight, denoted by $P_w(\mathbf{q}|\mathbf{w}, A)$, is computed as

$$P_w(\mathbf{q}|\mathbf{w},A) = \prod_{i=1}^{|\mathbf{q}|} \frac{1}{|\{j|(j,i)\in A\}|} \sum_{\forall(i,j)\in A} P(q_i|w_j) \quad (21)$$

A sample of the resulting phrase translation probabilities is shown in Figure 6, where a title phrase is shown together with the ten most probable query phrases that it will translate into according to the phrase model. Comparing to the word translation sample in Figure 2, phrases lead to a set of less ambiguous, more precise translations. For example, the term "vista", used alone, most likely refers to the Microsoft operating system, while in the query "sierra vista" it has a very different meaning.

6.2 Ranking Documents

Similar to the case of the word translation model, directly using the phrase-based query translation model, computed in Equations (17) to (19), to rank documents does not perform well. Unlike the word-based translation model, the phrase translation model cannot be interpolated with a unigram language model. We therefore resort to the linear ranking model framework for IR in which different models are incorporated as features [15].

The linear ranking model assumes a set of M features, f_m for $m = 1 \ldots M$. Each feature is an arbitrary function that maps (Q,D) to a real value, $f(Q, D) \in \mathbb{R}$. The model has M parameters, λ_m for $m = 1 \ldots M$, each for one feature function. The relevance score of a document D of a query Q is calculated as

$$Score(Q,D) = \sum_{m=1}^{M} \lambda_m f_m(Q,D) \quad (22)$$

Because NDCG is used to measure the quality of the retrieval system in this study, we optimize λ's for NDCG directly using the Powell Search algorithm [29] via cross-validation.

The features used in the linear ranking model are as follows.

#	Models	NDCG@1	NDCG@3	NDCG@10
1	**BM25**	0.3181	0.3413	0.4045
2	**WTM_M1**	0.3310	0.3566	0.4232
3	**PTM (*l*=5)**	0.3355	0.3605	0.4254
4	**PTM (*l*=3)**	0.3349	0.3602	0.4253
5	**PTM (*l*=2)**	0.3347	0.3603	0.4252

Table 5: Ranking results on the evaluation data set, where only the title field of each document is used. **PTM** is the linear ranking model of Equation (22), where all the features, including the two phrase translation model features f_{PT} and f_{LW} (with different maximum phrase length, specified by l), are incorporated.

Phrase lengths	NDCG@1	NDCG@3	NDCG@10
1	0.2966	0.3213	0.3861
2	0.2981	0.3248	0.3906
3	0.2996	0.3260	0.3917
4	0.3018	0.3278	0.3926
5	0.3028	0.3287	0.3932

Table 6: Ranking results on the evaluation data set, where only the title field of each document is used, using the linear ranking model of Equation (22) to which only two phrase translation model features f_{PT} and f_{LW} (with different phrase lengths) are incorporated.

- **Phrase translation feature:** $f_{PT}(Q,D,A) = \log P(Q|D)$, where $P(Q|D)$ is computed by Equations (17) to (19), and the phrase translation probability $P(\mathbf{q}|\mathbf{w_q})$ is estimated using Equation (20).
- **Lexical weight feature:** $f_{LW}(Q,D,A) = \log P(Q|D)$, where $P(Q|D)$ is computed by Equations (17) to (19), and the phrase translation probability is the computed as lexical weight according to Equation (21).
- **Phrase alignment feature:** $f_{PA}(Q,D,B) = \sum_{k=2}^{K} |a_k - b_{k-1} - 1|$, where B is a set of K bilingual phrases, a_k is the start position of the title phrase that was translated into the kth query phrase, and b_{k-1} is the end position of the title phrase that was translated into the (k-1)th query phrase. The feature, inspired by the distortion model in SMT [23], models the degree to which the query phrases are reordered. For all possible B, we only compute the feature value according to the Viterbi $B, B^* = \text{argmax}_B P(Q,B|D)$. We find B^* using the Viterbi algorithm, which is almost identical to the dynamic programming recursion of Equations (17) to (19), except that the *sum* operator in Equation (18) is replaced with the *max* operator.
- **Unaligned word penalty feature** $f_{UWP}(Q,D,A)$ is defined as the ratio between the number of unaligned query terms and the total number of query terms.
- **Language model feature:** $f_{LM}(Q,D) = \log P_s(Q|D)$, where $P_s(Q|D)$ is the unigram model with Jelinek-Mercer smoothing, i.e., defined by Equations (5) to (9), with $\beta = 1$.
- **Word translation feature:** $f_{WT}(Q,D) = \log P(Q|D)$, where $P(Q|D)$ is the word translation model defined by Equation (1), where the word translation probability is estimated with the EM training of Model 1.

6.3 Results and Discussions

Table 5 shows the main results of different phrase translation based retrieval models. Row 1 and Row 2 are models described in Table 4, and are listed here for comparison. Rows 3 to 5 are the linear ranking models using all the features described in Section 6.2, with different maximum phrase lengths, used in the two phrase translation features, f_{PT} and f_{LW}. The results show that (1) the phrase-based translation model leads to significant improvement (Row 3 vs. Row 2); and (2) using longer phrases in the phrase-based translation models does not seem to produce significantly better ranking results (Row 3 vs. Rows 4 and 5 is not statistically significant).

Phrase length	Query phrases	Title phrases
1	2,522,394	4,075,367
2	836,943	332,250
3	539,539	68,613
4	322,294	13,177
5	271,725	3,488

Table 7: Length distributions of title phrases and query phrases

To investigate the impact of the phrase length on ranking in more detail, we trained a series of linear ranking models that only use the two phrase translation features, i.e., f_{PT} and f_{LW}. The results in Table 6 show that longer phrases do yield some visible improvement up to the maximum length of five. This may suggest that some properties captured by longer phrases are also captured by other features. However, it will still be instructive, as future work, to explore the methods of preserving the improvement generated by longer phrases when more features are incorporated.

Table 7 shows the phrase length distributions in queries and titles. The phrases are detected using the Viterbi algorithm with a maximum length of 5. It is interesting to see that while the average length of titles is much larger than that of queries, the phrases detected in queries are longer than the phrases in titles. This implies that many long query phrases are translated from short title phrases. There are two possible interpretations. First, titles are longer than queries because a title is supposed to be a summary of a web document which may cover multiple topics whereas a user query usually focuses on only one particular topic of the document. Second, title language is more formal and concise whereas query language is more causal and wordy. So, for a specific topic, the description in the title (title phrase) is usually more well-formed and concise than that in queries, as illustrated by the examples in Table 8.

Analyzing the example bi-phrases extracted from titles and queries shown in Table 8 also helps us understand how the phrase-based translation model impacts retrieval results. The phrase model improves the effectiveness of retrieval from two aspects. First, it matches multi-word phrases in titles and queries (e.g., #1, #5, #6 and #7 query-title pairs in Table 8), thus reduces the ambiguities by capturing contextual information. Comparing with the previous approaches that are based on phrase retrieval models [10, 30] and higher-order n-gram models [31, 14], the phrase-based translation model provides an alternative, and in many cases more effective approach to dealing with the polysemy issue. Second, the phrase model is able to identify the phrase pairs that consist of different words but are semantically similar (e.g., #2, #3, #4 and #6 query-title pairs). We notice that these pairs cannot be easily captured by a word-based translation model. Thus, the phrase model is more effective than the word model in bridging the lexical gap between queries and documents. In summary, the results justify that the phrase-based translation model provides a unified solution to dealing with both the synonymy and the polysemy issues, as we claim in the introductory section of this paper.

#	Queries	Titles	Bi-phrases
1	canon d40 digital cameras	nikon d 40 digital camera reviews yahoo shopping	[canon d40 / nikon] [digital cameras / digital camera]
2	jerlon hair products	croda usa news and news releases	[jerlon hair products / croda]
3	jerlon hair products	curlaway testimonials	[jerlon hair products / curlaway]
4	recipe zucchini nut bread	cashew curry recipe 101 cookbooks	[recipe / recipe] [zucchini nut bread / cashew]
5	recipe zucchini nut bread	bellypleasers cookbook free recipe zucchini nut bread	[recipe / recipe] [zucchini / zucchini] [nut bread / nut bread]
6	home remedy stuffy nose	the best cold and flu home remedies	[home remedy / home remedies] [stuffy nose / cold]
7	washington tulip festival	tulip festival komo news seattle washington you-news trade	[washington / washington] [tulip festival / tulip festival]
8	cambridge high schools wisconsin	cambridge elementary school cambridge wisconsin wi school overview	[cambridge / cambridge] [high schools / school] [wisconsin / wisconsin]

Table 8: sample query/title pairs and the bi-phrases identified by the phrase-based translation model.

We also analyze the queries where the phrase model has a negative impact. An example is shown in #8 in Table 8. The model maps "*high schools*" in *D* to "*school*" in *Q*, ignoring the fact that the "*school*" in *Q* is actually an "*elementary school*". One possible reason is that the phrase model tries to learn bi-phrases that are most likely to be aligned without taking into account whether these phrases are reasonable in the monolingual context (i.e., in *D* and *Q*). Future improvement can be achieved by using an objective function in learning bi-phrases that takes into account both the likelihood of phrase alignment between *D* and *Q,* and the likelihood of monolingual phrase segmentation in *D* and *Q*.

6.4 Comparison with Latent Variable Models

This section compares the translation models with PLSA [17], one of the most studied latent variable models. Instead of building a full p.d.f. to probabilistically translate words in titles to words in queries, PLSA uses a factored generative model for word translation as

$$P(q|w) = \sum_z P(q|z)P(z|w)$$

where z is a vector of factors that mix to produce an observation [6]. The probabilities $P(q|z)$ and $P(z|w)$ are estimated using the EM algorithm on the query-title pairs derived from the clickthrough data. Empirically, the derived factors, frequently called topics or aspects, form a representation in the latent semantic space. Therefore, PLSA takes a different approach than phrase models to enhance the word-based translation model. Whilst the phrase model reduces the translation ambiguities by capturing some context information, PLSA smoothes translation probabilities among words occurring in similar context by capturing some semantic information.

In our retrieval experiments, we mix the PLSA model with the unigram language model, and use the ranking function as

$$P_s(Q|D) = \prod_{q \in Q} P_s(q|D) \quad (23)$$

$$P_s(q|D) = \alpha P(q|C) + (1-\alpha)P_{mx}(q|D) \quad (24)$$

$$P_{mx}(q|D) = \beta P(q|D) + (1-\beta)\sum_{w \in D} P_{fm}(q|w)P(w|D) \quad (25)$$

$$P_{fm}(q|w) = \sum_{k=1}^{K} P(q|z_k)P(z_k|w) \quad (26)$$

Notice that this ranking function has a similar form to that of the word-based translation model in Equations (5) and (9). *K* in Equation (26) is the number of factors of PLSA. Setting *K*=1 reduces the PLSA to the word-based translation model. In our experiments, we built PLSA models with *K* = 20, 50, 100, 200, 300, 500, and found no significant difference in retrieval results when $K \geq 100$.

As shown in Table 9, similar to the case of word-based translation model, using PLSA alone does not produce good retrieval results (Row 3 vs. Row 4). When mixing with unigram model, PLSA outperforms the word-based translation model by significant margins, but still slightly underperforms the phrase model. Since PLSA and phrase models use different strategies of improving word models, it will be interesting to explore how to combine their strengths. We leave it to future work.

#	Models	NDCG@1	NDCG@3	NDCG@10
1	WTM_M1	0.3310	0.3566	0.4232
2	PTM (*l*=5)	0.3355	0.3605	0.4254
3	PLSA (*K*=100)	0.3329	0.3592	0.4256
4	PLSA (*K*=100, *β*=1)	0.3244	0.3505	0.4145

Table 9: Comparison results of word, phrase translation models and PLSA, tested on the evaluation data set.

7. CONCLUSIONS

It has often been observed that search queries and Web documents are written in very different styles and with different vocabularies. In order to improve search results, it is important to bridge queries terms and document terms. Clickthrough data have been exploited for this purpose in several recent studies. In this paper, we extend the previous studies by developing a more general framework based on translation models and by extending noisy word-based translation to more precise phrase-based translation. This study shows that many techniques developed in SMT can be used for IR.

Instead of using query and document body pairs to train translation models, we use query and document title pairs. This choice is motivated by the smaller language discrepancy that we observed between queries and document titles. Two translation models are trained and integrated into the retrieval process: a word model and a phrase model. Our experimental results show that the translation models bring significant improvements to retrieval effectiveness. In particular, the use of the phrase translation model can bring additional improvements over the word translation model. This suggests the high potential of applying more sophisticated statistical machine translation techniques for improving Web search.

ACKNOWLEDGMENTS

The authors would like to thank Chris Quirk, Xiaolong Li, Kuansan Wang and Guihong Cao for the very helpful discussions and collaboration.

REFERENCES

[1] Microsoft web n-gram services. http://research.microsoft.com/web-ngram

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

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

[4] Bai, J., Nie, J-Y., Cao, G., and Bouchard, H. 2007. Using query contexts in information retrieval. In *SIGIR*, pp. 15-22.

[5] 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.

[6] Berger, A., Caruana, R., Cohn, D., Freitag, D., and Mittal, V. 2000. Bridging the lexical chasm: statistical approaches to answer-finding. In *SIGIR*, pp. 192-199.

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

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

[9] 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.

[10] Buckley, D., Allan, J., and Salton, G. 1995. Automatic retrieval approaches using SMART: TREC-2. *Information Processing and Management*, 31: 315-326.

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

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

[13] 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.

[14] Gao, J., Nie, J-Y., Wu, G., and Cao, G. 2004. Dependence language model for information retrieval. In *SIGIR*, pp. 170-177.

[15] Gao, J., Qin, H., Xia, X. and Nie, J-Y. 2005. Linear discriminative models for information retrieval. In *SIGIR,* pp. 290-297.

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

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

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

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

[20] Jeon, J., Croft, W. B., and Lee, J. H. 2005. Finding similar questions in large question and answer archives. In *CIKM*, pp. 84-90.

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

[22] Jones, K. S., Walker S., and Robertson, S. 1998. *A probabilistic model of information retrieval: development and status*. Technical Report TR-446, Cambridge University Computer Laboratory.

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

[24] Murdock, V., and Croft, W. B. 2005. A statisitcal model for sentence retrieval. In *HLT/EMNLP*, pp. 684-691.

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

[26] Nguyen, P., Gao, J., and Mahajan, M. 2007. *MSRLM: a scalable language modeling toolkit*. Technical report TR-2007-144, Microsoft Research.

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

[28] Och, F., and Ney, H. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4): 417-449.

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

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

[31] Song, F., and Croft, B. 1999. A general language model for information retrieval. In: *CIKM'99*, pp. 316–321.

[32] Sparck Jones, K. 1998. What is the role of NLP in text retrieval? In: *Naturnal language information retrieval* (Ed. T. Strzalkowski), Dordrecht: Kluwer.

[33] Svore, K., and Burges, C. 2009. A machine learning approach for improved BM25 retrieval. In *CIKM*, pp. 1811-1814.

[34] Wen, J. Nie, J.Y. and Zhang, H. 2002. Query Clustering Using User Logs, *ACM TOIS*, 20 (1): 59-81.

[35] Xu, J., and Croft, W. B. 2000. Improving effectiveness of information retrieval with local context analysis. In: *ACM TOIS*, 18(1): 79-112.

[36] Xue, X., Jeon, J., and Croft, B. 2008. Retrieval models for question and answer archives. In SIGIR, pp. 475-482.

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