Word or Phrase?
Learning Which Unit to Stress for Information Retrieval*

Young-In Song† and **Jung-Tae Lee**‡ and **Hae-Chang Rim**‡
†Microsoft Research Asia, Beijing, China
‡Dept. of Computer & Radio Communications Engineering, Korea University, Seoul, Korea
yosong@microsoft.com†, {jtlee,rim}@nlp.korea.ac.kr‡

Abstract

The use of phrases in retrieval models has been proven to be helpful in the literature, but no particular research addresses the problem of discriminating phrases that are likely to degrade the retrieval performance from the ones that do not. In this paper, we present a retrieval framework that utilizes both words and phrases flexibly, followed by a general learning-to-rank method for learning the potential contribution of a phrase in retrieval. We also present useful features that reflect the compositionality and discriminative power of a phrase and its constituent words for optimizing the weights of phrase use in phrase-based retrieval models. Experimental results on the TREC collections show that our proposed method is effective.

1 Introduction

Various researches have improved the quality of information retrieval by relaxing the traditional 'bag-of-words' assumption with the use of phrases. (Miller et al., 1999; Song and Croft, 1999) explore the use n-grams in retrieval models. (Fagan, 1987; Gao et al., 2004; Metzler and Croft, 2005; Tao and Zhai, 2007) use statistically-captured term dependencies within a query. (Strzalkowski et al., 1994; Kraaij and Pohlmann, 1998; Arampatzis et al., 2000) study the utility of various kinds of syntactic phrases.

Although use of phrases clearly helps, there still exists a fundamental but unsolved question: Do all phrases contribute an *equal* amount of increase in the performance of information retrieval models? Let us consider a search query '*World Bank Criticism*', which has the following phrases: '*world bank*' and '*bank criticism*'. Intuitively, the former should be given more importance than its constituents '*world*' and '*bank*', since the meaning of the original phrase cannot be predicted from the meaning of either constituent. In contrast, a relatively less attention could be paid to the latter '*bank criticism*', because there may be alternate expressions, of which the meaning is still preserved, that could possibly occur in relevant documents. However, virtually all the researches ignore the relation between a phrase and its constituent words when combining both words and phrases in a retrieval model.

Our approach to phrase-based retrieval is motivated from the following linguistic intuitions: a) phrases have relatively different degrees of significance, and b) the influence of a phrase should be differentiated based on the phrase's constituents in retrieval models. In this paper, we start out by presenting a simple language modeling-based retrieval model that utilizes both words and phrases in ranking with use of parameters that differentiate the relative contributions of phrases and words. Moreover, we propose a general learning-to-rank based framework to optimize the parameters of phrases against their constituent words for retrieval models that utilize both words and phrases. In order to estimate such parameters, we adapt the use of a cost function together with a gradient descent method that has been proven to be effective for optimizing information retrieval models with multiple parameters (Taylor et al., 2006; Metzler, 2007). We also propose a number of potentially useful features that reflect not only the characteristics of a phrase but also the information of its constituent words for minimizing the cost function. Our experimental results demonstrate that 1) differentiating the weights of each phrase over words yields statistically significant improvement in retrieval performance, 2) the gradient descent-based parameter optimization is reasonably appropriate

*This work was done while Young-In Song was with the Dept. of Computer & Radio Communications Engineering, Korea University.

Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, pages 1048–1056,
Suntec, Singapore, 2-7 August 2009.

to our task, and 3) the proposed features can distinguish good phrases that make contributions to the retrieval performance.

The rest of this paper is organized as follows. The next section discusses previous work. Section 3 presents our learning-based retrieval framework and features. Section 4 reports the evaluations of our techniques. Section 5 finally concludes the paper and discusses future work.

2 Previous Work

To date, there have been numerous researches to utilize phrases in retrieval models. One of the most earliest work on phrase-based retrieval was done by (Fagan, 1987). In (Fagan, 1987), the effectiveness of proximity-based phrases (*i.e.* words occurring within a certain distance) in retrieval was investigated with varying criteria to extract phrases from text. Subsequently, various types of phrases, such as sequential n-grams (Mitra et al., 1997), head-modifier pairs extracted from syntactic structures (Lewis and Croft, 1990; Zhai, 1997; Dillon and Gray, 1983; Strzalkowski et al., 1994), proximity-based phrases (Turpin and Moffat, 1999), were examined with conventional retrieval models (*e.g.* vector space model). The benefit of using phrases for improving the retrieval performance over simple 'bag-of-words' models was far less than expected; the overall performance improvement was only marginal and sometimes even inconsistent, specifically when a reasonably good weighting scheme was used (Mitra et al., 1997). Many researchers argued that this was due to the use of improper retrieval models in the experiments. In many cases, the early researches on phrase-based retrieval have only focused on extracting phrases, not concerning about how to devise a retrieval model that effectively considers both words and phrases in ranking. For example, the direct use of traditional vector space model combining a phrase weight and a word weight virtually yields the result assuming independence between a phrase and its constituent words (Srikanth and Srihari, 2003).

In order to complement the weakness, a number of research efforts were devoted to the modeling of dependencies between words directly within retrieval models instead of using phrases over the years (van Rijsbergen, 1977; Wong et al., 1985; Croft et al., 1991; Losee, 1994). Most studies were conducted on the probabilistic retrieval framework, such as the BIM model, and aimed on producing a better retrieval model by relaxing the word independence assumption based on the co-occurrence information of words in text. Although those approaches theoretically explain the relation between words and phrases in the retrieval context, they also showed little or no improvements in retrieval effectiveness, mainly because of their statistical nature. While a phrase-based approach selectively incorporated potentially-useful relation between words, the probabilistic approaches force to estimate parameters for all possible combinations of words in text. This not only brings parameter estimation problems but causes a retrieval system to fail by considering semantically-meaningless dependency of words in matching.

Recently, a number of retrieval approaches have been attempted to utilize a phrase in retrieval models. These approaches have focused to model statistical or syntactic phrasal relations under the language modeling method for information retrieval. (Srikanth and Srihari, 2003; Maisonnasse et al., 2005) examined the effectiveness of syntactic relations in a query by using language modeling framework. (Song and Croft, 1999; Miller et al., 1999; Gao et al., 2004; Metzler and Croft, 2005) investigated the effectiveness of language modeling approach in modeling statistical phrases such as n-grams or proximity-based phrases. Some of them showed promising results in their experiments by taking advantages of phrases soundly in a retrieval model.

Although such approaches have made clear distinctions by integrating phrases and their constituents effectively in retrieval models, they did not concern the different contributions of phrases over their constituents in retrieval performances. Usually a phrase score (or probability) is simply combined with scores of its constituent words by using a uniform interpolation parameter, which implies that a uniform contribution of phrases over constituent words is assumed. Our study is clearly distinguished from previous phrase-based approaches; we differentiate the influence of each phrase according to its constituent words, instead of allowing equal influence for all phrases.

3 Proposed Method

In this section, we present a phrase-based retrieval framework that utilizes both words and phrases effectively in ranking.

3.1 Basic Phrase-based Retrieval Model

We start out by presenting a simple phrase-based language modeling retrieval model that assumes uniform contribution of words and phrases. Formally, the model ranks a document D according to the probability of D generating phrases in a given query Q, assuming that the phrases occur independently:

$$s(Q;D) = P(Q|D) \approx \prod_{i=1}^{|Q|} P(q_i|q_{h_i}, D) \quad (1)$$

where q_i is the ith query word, q_{h_i} is the head word of q_i, and $|Q|$ is the query size. To simplify the mathematical derivations, we modify Eq. 1 using logarithm as follows:

$$s(Q;D) \propto \sum_{i=1}^{|Q|} \log[P(q_i|q_{h_i}, D)] \quad (2)$$

In practice, the phrase probability is mixed with the word probability (*i.e.* deleted interpolation) as:

$$P(q_i|q_{h_i},D) \approx \lambda P(q_i|q_{h_i},D) + (1-\lambda)P(q_i|D) \quad (3)$$

where λ is a parameter that controls the impact of the phrase probability against the word probability in the retrieval model.

3.2 Adding Multiple Parameters

Given a phrase-based retrieval model that utilizes both words and phrases, one would definitely raise a fundamental question on how much weight should be given to the phrase information compared to the word information. In this paper, we propose to differentiate the value of λ in Eq. 3 according to the importance of each phrase by adding multiple free parameters to the retrieval model. Specifically, we replace λ with well-known logistic function, which allows both numerical and categorical variables as input, whereas the output is bounded to values between 0 and 1.

Formally, the input of a logistic function is a set of evidences (*i.e.* feature vector) X generated from a given phrase and its constituents, whereas the output is the probability predicted by fitting X to a logistic curve. Therefore, λ is replaced as follows:

$$\lambda(X) = \frac{1}{1 + e^{-f(X)}} \cdot \alpha \quad (4)$$

where α is a scaling factor to confine the output to values between 0 and α.

$$f(X) = \beta_0 + \sum_{i=1}^{|X|} \beta_i x_i \quad (5)$$

where x_i is the ith feature, β_i is the coefficient parameter of x_i, and β_0 is the 'intercept', which is the value of $f(X)$ when all feature values are zero.

3.3 RankNet-based Parameter Optimization

The β parameters in Eq. 5 are the ones we wish to learn for resulting retrieval performance via parameter optimization methods. In many cases, parameters in a retrieval model are empirically determined through a series of experiments or automatically tuned via machine learning to maximize a retrieval metric of choice (*e.g.* mean average precision). The most simple but guaranteed way would be to directly perform brute force search for the global optimum over the entire parameter space. However, not only the computational cost of this so-called direct search would become undoubtfully expensive as the number of parameters increase, but most retrieval metrics are non-smooth with respect to model parameters (Metzler, 2007). For these reasons, we propose to adapt a learning-to-rank framework that optimizes multiple parameters of phrase-based retrieval models effectively with less computation cost and without any specific retrieval metric.

Specifically, we use a gradient descent method with the RankNet cost function (Burges et al., 2005) to perform effective parameter optimizations, as in (Taylor et al., 2006; Metzler, 2007). The basic idea is to find a local minimum of a cost function defined over pairwise document preference. Assume that, given a query Q, there is a set of document pairs $\mathcal{R}_Q$ based on relevance judgements, such that $(D_1, D_2) \in \mathcal{R}_Q$ implies document D_1 should be ranked higher than D_2. Given a defined set of pairwise preferences $\mathcal{R}$, the RankNet cost function is computed as:

$$C(\mathcal{Q}, \mathcal{R}) = \sum_{\forall Q \in \mathcal{Q}} \sum_{\forall (D_1, D_2) \in \mathcal{R}_Q} \log(1 + e^Y) \quad (6)$$

where $\mathcal{Q}$ is the set of queries, and $Y = s(Q;D_2) - s(Q;D_1)$ using the current parameter setting.

In order to minimize the cost function, we compute gradients of Eq. 6 with respect to each parameter β_i by applying the chain rule:

$$\frac{\delta C}{\delta \beta_i} = \sum_{\forall Q \in \mathcal{Q}} \sum_{\forall (D_1, D_2) \in \mathcal{R}_Q} \frac{\delta C}{\delta Y} \frac{\delta Y}{\delta \beta_i} \quad (7)$$

where $\frac{\delta C}{\delta Y}$ and $\frac{\delta Y}{\delta \beta_i}$ are computed as:

$$\frac{\delta C}{\delta Y} = \frac{\exp[s(Q;D_2) - s(Q;D_1)]}{1 + \exp[s(Q;D_2) - s(Q;D_1)]} \quad (8)$$

$$\frac{\delta Y}{\delta \beta_i} = \frac{\delta s(Q; D_2)}{\delta \beta_i} - \frac{\delta s(Q; D_1)}{\delta \beta_i} \quad (9)$$

With the retrieval model in Eq. 2 and $\lambda(X)$, $f(X)$ in Eq. 4 and 5, the partial derivate of $s(Q; D)$ with respect to β_i is computed as follows:

$$\frac{\delta s(Q;D)}{\delta \beta_i} = \sum_{i=1}^{|Q|} \frac{x_i \lambda(X)(1 - \frac{\lambda(X)}{\alpha}) \cdot (P(q_i|q_{h_i}, D) - P(q_i|D))}{\lambda(X) P(q_i|q_{h_i}, D) + (1 - \lambda(X)) P(q_i|D)} \quad (10)$$

3.4 Features

We experimented with various features that are potentially useful for not only discriminating a phrase itself but characterizing its constituents. In this section, we report only the ones that have made positive contributions to the overall retrieval performance. The two main criteria considered in the selection of the features are the followings: *compositionality* and *discriminative power*.

Compositionality Features

Features on phrase compositionality are designed to measure how likely a phrase can be represented as its constituent words without forming a phrase; if a phrase in a query has very high compositionality, there is a high probability that its relevant documents do not contain the phrase. In this case, emphasizing the phrase unit could be very risky in retrieval. In the opposite case that a phrase is uncompositional, it is obvious that occurrence of a phrase in a document can be a stronger evidence of relevance than its constituent words.

Compositionality of a phrase can be roughly measured by using corpus statistics or its linguistic characteristics; we have observed that, in many times, an extremely-uncompositional phrase appears as a noun phrase, and the distance between its constituent words is generally fixed within a short distance. In addition, it has a tendency to be used repeatedly in a document because its semantics cannot be represented with individual constituent words. Based on these intuitions, we devise the following features:

Ratio of multiple occurrences (RMO): This is a real-valued feature that measures the ratio of the phrase repeatedly used in a document. The value of this feature is calculated as follows:

$$x = \frac{\sum_{\forall D; count(w_i \rightarrow w_{h_i}, D) > 1} count(w_i \rightarrow w_{h_i}, D)}{count(w_i \rightarrow w_{h_i}, C) + \gamma} \quad (11)$$

where $w_i \rightarrow w_{h_i}$ is a phrase in a given query, $count(x, y)$ is the count of x in y, and γ is a small-valued constant to prevent unreliable estimation by very rarely-occurred phrases.

Ratio of single-occurrences (RSO): This is a binary feature that indicates whether or not a phrase occurs once in most documents containing it. This can be regarded as a supplementary feature of RMO.

Preferred phrasal type (PPT): This feature indicates the phrasal type that the phrase prefers in a collection. We consider only two cases (whether the phrase prefers verb phrase or adjective-noun phrase types) as features in the experiments[1].

Preferred distance (PD): This is a binary feature indicating whether or not the phrase prefers long distance (> 1) between constituents in the document collection.

Uncertainty of preferred distance (UPD): We also use the entropy (H) of the modification distance (d) of the given phrase in the collection to measure the compositionality; if the distance is not fixed and is highly uncertain, the phrase may be very compositional. The entropy is computed as:

$$x = H(p(d = x|w_i \rightarrow w_{h_i})) \quad (12)$$

where $d \in 1, 2, 3, long$ and all probabilities are estimated with discount smoothing. We simply use two binary features regarding the uncertainty of distance; one indicates whether the uncertainty of a phrase is very high (> 0.85), and the other indicates whether the uncertainty is very low (< 0.05)[2].

Uncertainty of preferred phrasal type (UPPT): As similar to the uncertainty of preferred distance, the uncertainty of the preferred phrasal type of the phrase can be also used as a feature. We consider this factor as a form of a binary feature indicating whether the uncertainty is very high or not.

Discriminative Power Features

In some cases, the occurrence of a phrase can be a valuable evidence even if the phrase is very likely to be compositional. For example, it is well known that the use of a phrase can be effective in retrieval when its constituent words appear very frequently in the collection, because each word would have a very low discriminative power for relevance. On the contrary, if a constituent word occurs very

[1] For other phrasal types, significant differences were not observed in the experiments.

[2] Although it may be more natural to use a real-valued feature, we use these binary features because of the two practical reasons; firstly, it could be very difficult to find an adequate transformation function with real values, and secondly, the two intervals at tails were observed to be more important than the rest.

rarely in the collection, it could not be effective to use the phrase even if the phrase is highly uncompositional. Similarly, if the probability that a phrase occurs in a document where its constituent words co-occur is very high, we might not need to place more emphasis on the phrase than on words, because co-occurrence information naturally incorporated in retrieval models may have enough power to distinguish relevant documents. Based on these intuitions, we define the following features:

Document frequency of constituents (DF): We use the document frequency of a constituent as two binary features: one indicating whether the word has very high document frequency ($>$10% of documents in a collection) and the other one indicating whether it has very low document frequency ($<$0.2% of documents, which is approximately 1,000 in our experiments).

Probability of constituents as phrase (CPP): This feature is computed as a relative frequency of documents containing a phrase over documents where two constituent words appear together.

One interesting fact that we observe is that document frequency of the modifier is generally a stronger evidence on the utility of a phrase in retrieval than of the headword. In the case of the headword, we could not find an evidence that it has to be considered in phrase weighting. It seems to be a natural conclusion, because the importance of the modifier word in retrieval is subordinate to the relation to its headword, but the headword is not in many phrases. For example, in the case of the query *'tropical storms'*, retrieving a document only containing *tropical* can be meaningless, but a document about *storm* can be meaningful. Based on this observation, we only incorporate document frequency features of syntactic modifiers in the experiments.

4 Experiments

In this section, we report the retrieval performances of the proposed method with appropriate baselines over a range of training sets.

4.1 Experimental Setup

Retrieval models: We have set two retrieval models, namely the *word model* and the (phrase-based) *one-parameter model*, as baselines. The ranking function of the word model is equivalent to Eq. 2, with λ in Eq. 3 being set to zero (*i.e.* the phrase probability makes no effect on the ranking). The ranking function of the one-parameter model is also equivalent to Eq. 2, with λ in Eq. 3 used "as is" (*i.e.* as a constant parameter value optimized using gradient descent method, without being replaced to a logistic function). Both baseline models cannot differentiate the importance of phrases in a query. To make a distinction from the baseline models, we will name our proposed method as a *multi-parameter model*.

In our experiments, all the probabilities in all retrieval models are smoothed with the collection statistics by using dirichlet priors (Zhai and Lafferty, 2001).

Corpus (Training/Test): We have conducted large-scale experiments on three sets of TREC's Ad Hoc Test Collections, namely TREC-6, TREC-7, and TREC-8. Three query sets, TREC-6 topics 301-350, TREC-7 topics 351-400, and TREC-8 topics 401-450, along with their relevance judgments have been used. We only used the title field as query.

When performing experiments on each query set with the one-parameter and the multi-parameter models, the other two query sets have been used for learning the optimal parameters. For each query in the training set, we have generated document pairs for training by the following strategy: first, we have gathered top m ranked documents from retrieval results by using the word model and the one-parameter model (by manually setting λ in Eq. 3 to the fixed constants, 0 and 0.1 respectively). Then, we have sampled at most r relevant documents and n non-relevant documents from each one and generated document pairs from them. In our experiments, m, r, and n is set to 100, 10, and 40, respectively.

Phrase extraction and indexing: We evaluate our proposed method on two different types of phrases: syntactic head-modifier pairs (syntactic phrases) and simple bigram phrases (statistical phrases). To index the syntactic phrases, we use the method proposed in (Strzalkowski et al., 1994) with Connexor FDG parser[3], the syntactic parser based on the functional dependency grammar (Tapanainen and Jarvinen, 1997). All necessary information for feature values were indexed together for both syntactic and statistical phrases. To maintain indexes in a manageable size, phrases

[3]Connexor FDG parser is a commercial parser; the demo is available at: `http://www.connexor.com/demo`

		Test set ← *Training set*					
		6 ← 7+8		7 ← 6+8		8 ← 6+7	
Model	*Metric* \ *Query*	*all*	*partial*	*all*	*partial*	*all*	*partial*
Word (Baseline 1)	MAP	0.2135	0.1433	0.1883	0.1876	0.2380	0.2576
	R-Prec	0.2575	0.1894	0.2351	0.2319	0.2828	0.2990
	P@10	0.3660	0.3333	0.4100	0.4324	0.4520	0.4517
One-parameter (Baseline 2)	MAP	0.2254	*0.1633*†	0.1988	0.2031	0.2352	0.2528
	R-Prec	0.2738	0.2165	0.2503	0.2543	0.2833	0.2998
	P@10	0.3820	0.3600	**0.4540**	**0.4971**	0.4580	0.4621
Multi-parameter (Proposed)	MAP	***0.2293***‡	***0.1697***‡	***0.2038***†	***0.2105***†	**0.2452**	**0.2701**
	R-Prec	**0.2773**	**0.2225**	**0.2534**	**0.2589**	**0.2891**	**0.3099**
	P@10	**0.4020**	**0.3933**	**0.4540**	**0.4971**	**0.4700**	**0.4828**

Table 1: Retrieval performance of different models on syntactic phrases. ***Italicized*** MAP values with symbols † and ‡ indicate statistically significant improvements over the word model according to Student's t-test at $p < 0.05$ level and $p < 0.01$ level, respectively. **Bold** figures indicate the best performed case for each metric.

that occurred less than 10 times in the document collections were not indexed.

4.2 Experimental Results

Table 1 shows the experimental results of the three retrieval models on the syntactic phrase (head-modifier pair). In the table, *partial* denotes the performance evaluated on queries containing more than one phrase that appeared in the document collection[4]; this shows the actual performance difference between models. Note that the ranking results of all retrieval models would be the same as the result of the word model if a query does not contain any phrases in the document collection, because $P(q_i|q_{h_i}, D)$ would be calculated as zero eventually. As evaluation measures, we used the mean average precision (MAP), R-precision (R-Prec), and precisions at top 10 ranks (P@10).

As shown in Table 1, when a syntactic phrase is used for retrieval, one-parameter model trained by gradient-descent method generally performs better than the word model, but the benefits are inconsistent; it achieves approximately 15% and 8% improvements on the *partial* query set of TREC-6 and 7 over the word model, but it fails to show any improvement on TREC-8 queries. This may be a natural result since the one-parameter model is very sensitive to the averaged contribution of phrases used for training. Compared to the queries in TREC-6 and 7, the TREC-8 queries contain more phrases that are not effective for retrieval (*i.e.* ones that hurt the retrieval performance when used). This indicates that without distinguishing effective phrases from ineffective phrases for retrieval, the model trained from one training set for phrase would not work consistently on other unseen query sets.

Note that the proposed model outperforms all the baselines over all query sets; this shows that differentiating relative contributions of phrases can improve the retrieval performance of the one-parameter model considerably and consistently. As shown in the table, the multi-parameter model improves by approximately 18% and 12% on the TREC-6 and 7 *partial* query sets, and it also significantly outperforms both the word model and the one-parameter model on the TREC-8 query set. Specifically, the improvement on the TREC-8 query set shows one advantage of using our proposed method; by separating potentially-ineffective phrases and effective phrases based on the features, it not only improves the retrieval performance for each query but makes parameter learning less sensitive to the training set.

Figure 1 shows some examples demonstrating the different behaviors of the one-parameter model and the multi-parameters model. On the figure, the un-dotted lines indicate the variation of average precision scores when λ value in Eq. 3 is manually set. As λ gets closer to 0, the ranking formula becomes equivalent to the word model.

As shown in the figure, the optimal point of λ is quiet different from query to query. For example, in cases of the query *'ferry sinking'* and *industrial*

[4]The number of queries containing a phrase in TREC-6, 7, and 8 query set is 31, 34, and 29, respectively.

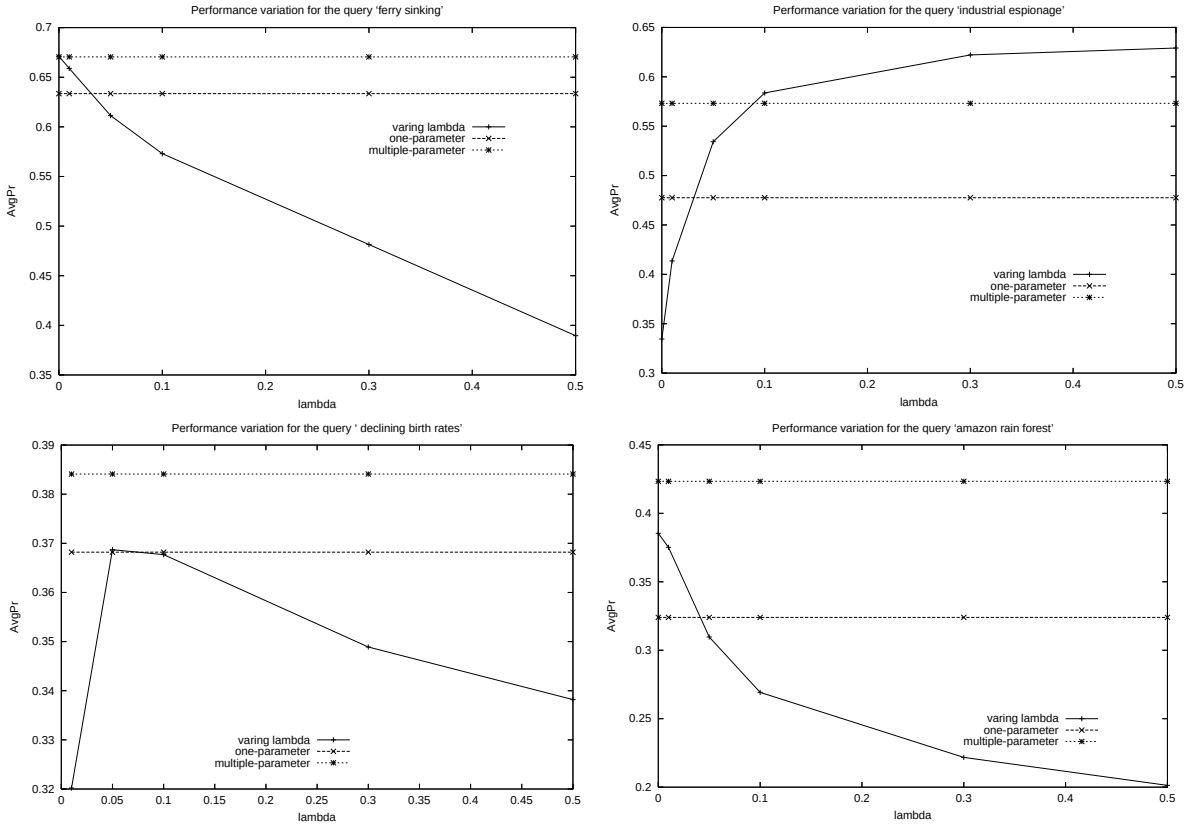

Figure 1: Performance variations for the queries *'ferry sinking', 'industrial espionage', 'declining birth rate'* and *'Amazon rain forest'* according to λ in Eq. 3.

espionage' on the upper side, the optimal point is the value close to 0 and 1 respectively. This means that the occurrences of the phrase *'ferry sinking'* in a document is better to be less-weighted in retrieval while *'industrial espionage'* should be treated as a much more important evidence than its constituent words. Obviously, such differences are not good for one-parameter model assuming relative contributions of phrases uniformly. For both opposite cases, the multi-parameter model significantly outperforms one-parameter model.

The two examples at the bottom of Figure 1 show the difficulty of optimizing phrase-based retrieval using one uniform parameter. For example, the query *'declining birth rate'* contains two different phrases, *'declining rate'* and *'birth rate'*, which have potentially-different effectiveness in retrieval; the phrase *'declining rate'* would not be helpful for retrieval because it is highly compositional, but the phrase *'birth rate'* could be a very strong evidence for relevance since it is conventionally used as a phrase. In this case, we can get only small benefit from the one-parameter model even if we find optimal λ from gradient descent, because it will be just a compromised value between two different, optimized λs. For such query, the multi-parameter model could be more effective than the one-parameter model by enabling to set different λs on phrases according to their predicted contributions. Note that the multi-parameter model significantly outperforms the one-parameter model and all manually-set λs for the queries *'declining birth rate'* and *'Amazon rain forest'*, which also has one effective phrase, *'rain forest'*, and one non-effective phrase, *'Amazon forest'*.

Since our method is not limited to a particular type of phrases, we have also conducted experiments on statistical phrases (bigrams) with a reduced set of features directed applicable; RMO, RSO, PD[5], DF, and CPP; the features requiring linguistic preprocessing (*e.g.* PPT) are not used, because it is unrealistic to use them under bigram-based retrieval setting. Moreover, the feature UPD is not used in the experiments because the uncer-

[5]In most cases, the distance between words in a bigram is 1, but sometimes, it could be more than 1 because of the effect of stopword removal.

		Test ← *Training*		
Model	*Metric*	6 ← 7+8	7 ← 6+8	8 ← 6+7
Word (Baseline 1)	MAP	0.2135	0.1883	0.2380
	R-Prec	0.2575	0.2351	0.2828
	P@10	0.3660	0.4100	0.4520
One-parameter (Baseline 2)	MAP	**0.2229**	0.1979	*0.2492*†
	R-Prec	**0.2716**	0.2456	**0.2959**
	P@10	0.3720	0.4500	**0.4620**
Multi-parameter (Proposed)	MAP	0.2224	***0.2025***†	***0.2499***†
	R-Prec	0.2707	**0.2457**	0.2952
	P@10	**0.3780**	**0.4520**	0.4600

Table 2: Retrieval performance of different models, using statistical phrases.

tainty of preferred distance does not vary much for bigram phrases. The results are shown in Table 2.

The results of experiments using statistical phrases show that multi-parameter model yields additional performance improvement against baselines in many cases, but the benefit is insignificant and inconsistent. As shown in Table 2, according to the MAP score, the multi-parameter model outperforms the one-parameter model on the TREC-7 and 8 query sets, but it performs slightly worse on the TREC-6 query set.

We suspect that this is because of the lack of features to distinguish an effective statistical phrases from ineffective statistical phrase. In our observation, the bigram phrases also show a very similar behavior in retrieval; some of them are very effective while others can deteriorate the performance of retrieval models. However, in case of using statistical phrases, the λ computed by our multi-parameter model would be often similar to the one computed by the one-parameter model, when there is no sufficient evidence to differentiate a phrase. Moreover, the insufficient amount of features may have caused the multi-parameter model to overfit to the training set easily.

The small size of training corpus could be an another reason. The number of queries we used for training is less than 80 when removing a query not containing a phrase, which is definitely not a sufficient amount to learn optimal parameters. However, if we recall that the multi-parameter model worked reasonably in the experiments using syntactic phrases with the same training sets, the lack of features would be a more important reason.

Although we have not mainly focused on features in this paper, it would be strongly necessary to find other useful features, not only for statistical phrases, but also for syntactic phrases. For example, statistics from query logs and the probability of snippet containing a same phrase in a query is clicked by user could be considered as useful features. Also, the size of the training data (queries) and the document collection may not be sufficient enough to conclude the effectiveness of our proposed method; our method should be examined in a larger collection with more queries. Those will be one of our future works.

5 Conclusion

In this paper, we present a novel method to differentiate impacts of phrases in retrieval according to their relative contribution over the constituent words. The contributions of this paper can be summarized in three-fold: a) we proposed a general framework to learn the potential contribution of phrases in retrieval by "*parameterizing*" the factor interpolating the phrase weight and the word weight on features and optimizing the parameters using RankNet-based gradient descent algorithm, b) we devised a set of potentially useful features to distinguish effective and non-effective phrases, and c) we showed that the proposed method can be effective in terms of retrieval by conducting a series of experiments on the TREC test collections.

As mentioned earlier, the finding of additional features, specifically for statistical phrases, would be necessary. Moreover, for a thorough analysis on the effect of our framework, additional experiments on larger and more realistic collections (*e.g.* the Web environment) would be required. These will be our future work.

References

Avi Arampatzis, Theo P. van der Weide, Cornelis H. A. Koster, and P. van Bommel. 2000. Linguistically-motivated information retrieval. In *Encyclopedia of Library and Information Science*.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of ICML '05*, pages 89–96.

W. Bruce Croft, Howard R. Turtle, and David D. Lewis. 1991. The use of phrases and structured queries in information retrieval. In *Proceedings of SIGIR '91*, pages 32–45.

Martin Dillon and Ann S. Gray. 1983. Fasit: A fully automatic syntactically based indexing system. *Journal of the American Society for Information Science*, 34(2):99–108.

Joel L. Fagan. 1987. Automatic phrase indexing for document retrieval. In *Proceedings of SIGIR '87*, pages 91–101.

Jianfeng Gao, Jian-Yun Nie, Guangyuan Wu, and Guihong Cao. 2004. Dependence language model for information retrieval. In *Proceedings of SIGIR '04*, pages 170–177.

Wessel Kraaij and Renée Pohlmann. 1998. Comparing the effect of syntactic vs. statistical phrase indexing strategies for dutch. In *Proceedings of ECDL '98*, pages 605–617.

David D. Lewis and W. Bruce Croft. 1990. Term clustering of syntactic phrases. In *Proceedings of SIGIR '90*, pages 385–404.

Robert M. Losee, Jr. 1994. Term dependence: truncating the bahadur lazarsfeld expansion. *Information Processing and Management*, 30(2):293–303.

Loic Maisonnasse, Gilles Serasset, and Jean-Pierre Chevallet. 2005. Using syntactic dependency and language model x-iota ir system for clips mono and bilingual experiments in clef 2005. In *Working Notes for the CLEF 2005 Workshop*.

Donald Metzler and W. Bruce Croft. 2005. A markov random field model for term dependencies. In *Proceedings of SIGIR '05*, pages 472–479.

Donald Metzler. 2007. Using gradient descent to optimize language modeling smoothing parameters. In *Proceedings of SIGIR '07*, pages 687–688.

David R. H. Miller, Tim Leek, and Richard M. Schwartz. 1999. A hidden markov model information retrieval system. In *Proceedings of SIGIR '99*, pages 214–221.

Mandar Mitra, Chris Buckley, Amit Singhal, and Claire Cardie. 1997. An analysis of statistical and syntactic phrases. In *Proceedings of RIAO '97*, pages 200–214.

Fei Song and W. Bruce Croft. 1999. A general language model for information retrieval. In *Proceedings of CIKM '99*, pages 316–321.

Munirathnam Srikanth and Rohini Srihari. 2003. Exploiting syntactic structure of queries in a language modeling approach to ir. In *Proceedings of CIKM '03*, pages 476–483.

Tomek Strzalkowski, Jose Perez-Carballo, and Mihnea Marinescu. 1994. Natural language information retrieval: Trec-3 report. In *Proceedings of TREC-3*, pages 39–54.

Tao Tao and ChengXiang Zhai. 2007. An exploration of proximity measures in information retrieval. In *Proceedings of SIGIR '07*, pages 295–302.

Pasi Tapanainen and Timo Jarvinen. 1997. A non-projective dependency parser. In *Proceedings of ANLP '97*, pages 64–71.

Michael Taylor, Hugo Zaragoza, Nick Craswell, Stephen Robertson, and Chris Burges. 2006. Optimisation methods for ranking functions with multiple parameters. In *Proceedings of CIKM '06*, pages 585–593.

Andrew Turpin and Alistair Moffat. 1999. Statistical phrases for vector-space information retrieval. In *Proceedings of SIGIR '99*, pages 309–310.

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

S. K. M. Wong, Wojciech Ziarko, and Patrick C. N. Wong. 1985. Generalized vector spaces model in information retrieval. In *Proceedings of SIGIR '85*, pages 18–25.

Chengxiang Zhai and John Lafferty. 2001. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of SIGIR '01*, pages 334–342.

Chengxiang Zhai. 1997. Fast statistical parsing of noun phrases for document indexing. In *Proceedings of ANLP '97*, pages 312–319.