

Chapter 5

Query-Dependent Feature Weighting

1 Overview

Most traditional information retrieval models, such as language modeling and BM25, utilize very simple user query models. These models tend to treat query terms as independent and of uniform importance. Simple heuristics, such as inverse document frequency (*idf*), are integral parts of these models and can be thought of as a simple query term weighting model, but they are very rigid and are based on a single data source. Furthermore, it is not clear if *idf* is an appropriate measure of importance for phrases and other generic concepts (Pickens and Croft 1999). Recent research has shown that modeling query term dependencies and using non-uniform query term weighting (beyond *idf*) can significantly improve retrieval effectiveness, especially on very large collections and for long, complex queries (Bendersky and Croft 2008; Lease 2009; Metzler and Croft 2005).

This chapter extends the basic MRF model by automatically learning query-dependent concept weights. The extension is a generic framework for learning the importance of query term concepts in a way that directly optimizes an underlying retrieval metric. It is important to note that this is quite different from query segmentation approaches (Bergsma and Wang 2007; Guo et al. 2008; Tan and Peng 2008). Optimizing segmentation accuracy is not guaranteed to optimize retrieval effectiveness. By implementing concept weighting directly into the underlying retrieval model we avoid the issue of metric divergence (Morgan et al. 2004). As we will show, this strategy yields strong retrieval effectiveness gains.

As an illustration of such metric divergence, Table 5.1 shows an actual example of unigram and bigram concept importances learned within the model for the query “civil war battle reenactments”. If, instead, the weighting was done based on the output of a query segmenter, then it is likely that the phrase “civil war” and perhaps “battle reenactments” would be given large weights. However, the model assigns high weights to the unigram “reenactments” and the bigram “war battle”, which happen to be the most *discriminative* (between relevant and non-relevant documents) concepts, not the *most likely* concepts in terms of query segmentation.

The remainder of this chapter, which is based on Bendersky et al. (2010), describes one possible approach for incorporating query-dependent weighting into the

Table 5.1 Query-dependent concept weights generated for query “civil war battle reenactments”

Concept	Weight
civil	0.0619
war	0.1947
battle	0.0913
reenactments	0.3487
civil war	0.1959
war battle	0.2458
battle reenactments	0.0540

basic MRF framework. The chapter begins by laying out the theoretical foundations and concludes with a detailed empirical evaluation that demonstrates the practical utility of the approach.

2 Related Work

Modeling atomic query concepts through term dependencies, or proximities, proved to have a significant positive impact on retrieval effectiveness on both TREC and Web corpora (Bai et al. 2008; Bendersky et al. 2009; Cummins and O’Riordan 2009; Metzler and Croft 2005; Mishne and de Rijke 2005; Tao 2007). Most of this work, however, was restricted to modeling term dependencies, rather than weighting them. In other words, all concept matches in the query had the same impact on the document score. While this assumption is reasonable for short keyword queries, it is much less reasonable for longer, more complex queries.

Recent work, focused on verbose queries, started to explore the direction of assigning varying *document independent* weights to query concepts. Kumaran and Carvalho (2009) address this by automatically removing extraneous terms that may have a negative effect on the overall retrieval performance of a query. Bendersky and Croft (2008) use a supervised discovery method for “key concepts” in verbose queries, and use a ranking approach that integrates the weighted key concepts with the original query. They find that weighted key concepts approach outperforms the standard bag-of-words model, however its performance is on par with the sequential dependence model that does not use any concept weighting (Metzler and Croft 2005). Most recently, Lease (2009) extended his previous work on term weighting (Lease et al. 2009) to show that incorporating learned term weights in a sequential dependence model improves the retrieval performance over the unweighted variant for verbose description queries on a number of TREC collections.

An additional information retrieval method that is related to the work described in this chapter is pseudo relevance feedback (PRF), which can be viewed as both query expansion and query term weighting technique (Lavrenko and Croft 2001). Recently, researchers separately focused on both modeling term dependencies (Metzler and Croft 2007) and term weighting (Cao et al. 2008) within the PRF framework.

There are two major differences between the model we describe here and the aforementioned previous work. First, the extension of the basic MRF model described here provides a principled retrieval framework that, unlike previously proposed methods, naturally combines both term and phrase weights. Second, the model parameters are estimated by directly maximizing the underlying retrieval metric. This differentiates the model described here from previous methods for concept weighting that employed indirect parameter estimation, maximizing metrics not directly related to retrieval performance such as classification accuracy (Bendersky and Croft 2008; Cao et al. 2008) or expected query model performance (Lease et al. 2009; Lease 2009). We show that the direct optimization approach allows us to achieve consistent performance gains over a range of query types, while previous work on concept weighting was mainly concentrated on verbose (Bendersky and Croft 2008; Lease 2009) or expanded (Cao et al. 2008) queries.

Direct optimization of an underlying retrieval metric ties the model described in this chapter to learning-to-rank approaches for information retrieval (LR4IR) (see Liu 2009 for a recent survey). The formulation of metric optimization is similar to some previous work, and thus allows us to build upon the existing direct optimization methods, such as those covered in Chap. 6. The primary benefit of the method lies in the fact that we are not limited to a linear combination of pairwise query-document features, as is usually the case in LR4IR (Liu 2009). Instead, we can also use *individual concept features* to effectively learn a concept weighting model in a similar, yet much more flexible, way than that proposed by Gey (1994). As we will show, this approach allows us to improve upon retrieval models that use only query-document dependent features.

3 Weighted Dependence Model

One of the primary limitations of the sequential dependence variant of the MRF model is the fact that all matches of the same type (e.g., term, ordered window, or unordered window) are treated as being equally important. This is the result of the massive parameter tying that is done with the sequential dependence model. Instead, it would be desirable to weight, *a priori*, different terms (or bigrams) within the query differently based on *query-level evidence*. For example, in a verbose query, there will likely be a few concepts (terms or phrases) within the query that will carry the most weight. While the sequential dependence model would treat all of the concepts as equally important, we would like to be able to weight the concepts appropriately, with regard to each other.

There are several ways to model this in the MRF model, but perhaps the most straightforward is to allow the parameters λ to depend on the concept that they are being applied to, rather than some global weight. This can be achieved by defining the potentials within the model as follows:

$$\psi(q_i, D; \Lambda) = \exp[\lambda(q_i) f_T(q_i, D)], \quad (5.1)$$

$$\psi(q_i, q_{i+1}, D; \Lambda) = \exp[\lambda(q_i, q_{i+1})f_O(q_i, q_{i+1}, D) + \lambda(q_i, q_{i+1})f_U(q_i, q_{i+1}, D)], \quad (5.2)$$

where $\lambda(q_i)$ is a parameter that depends on term q_i and $\lambda(q_i, q_{i+1})$ is a parameter that depends on the bigram q_i, q_{i+1} . In this setting, each term and bigram has a separate weight associated with it that is *independent of the document*. This parameter should be some measure of the general importance of the concept with respect to the rest of the query.

Although this formulation of the model is more general, it results in an infeasibly large number of parameters, since each λ now depends on the identity of one (or two) query terms. This was not a problem in the original formulation of the sequential dependence model, because it was assumed that all of the λ parameters, for a given match type, were tied to the same value, resulting in just three parameters. The solution described here is in the middle ground between these two extremes. We assume that the parameters λ take on a parameterized form. For simplicity, we assume the following weighted linear form:

$$\lambda(q_i) = \sum_{j=1}^{k_u} w_j^u g_j^u(q_i), \quad (5.3)$$

$$\lambda(q_i, q_{i+1}) = \sum_{j=1}^{k_b} w_j^b g_j^b(q_i, q_{i+1}), \quad (5.4)$$

where $g^u(q_i)$ and $g^b(q_i, q_{i+1})$ are features defined over unigrams and bigrams, respectively. Similarly, w^u and w^b are free parameters that must be estimated. If there are k_u unigram features and k_b bigram features, then we have a total of $k_u + k_b$ total parameters to estimate, compared to three in the sequential dependence model. The features $g^u(q_i)$ and $g^b(q_i, q_{i+1})$ are document independent and should be useful for determining the relative importance of the concept within the context of the query.

When the parameters λ have this parametric form, the final MRF ranking function can be shown to have the following form:

$$\begin{aligned} P(D|Q) &\stackrel{\text{rank}}{=} \sum_{i=1}^{k_u} w_i^u \sum_{q \in Q} g_i^u(q) f_T(q, D) \\ &\quad + \sum_{i=1}^{k_b} w_i^b \sum_{q_j, q_{j+1} \in Q} g_i^b(q_j, q_{j+1}) f_O(q_j, q_{j+1}, D) \\ &\quad + \sum_{i=1}^{k_b} w_i^b \sum_{q_j, q_{j+1} \in Q} g_i^b(q_j, q_{j+1}) f_U(q_j, q_{j+1}, D) \end{aligned} \quad (5.5)$$

which we call the *weighted sequential dependence model*. It is important to note that this model can easily be extended to handle other dependence assumptions,

Table 5.2 Statistics used to estimate term importance for a concept e . Concept e is either a query term q_i or a sequential query term pair $q_i q_{i+1}$

Data Source	Feature	Description
Collection	cf_e	Collection frequency for concept e
	df_e	Document frequency for concept e
G-Grams	$gf(e)$	n -gram count of concept e
MSN Query Log	$qe_cnt(e)$	Number of exact matches of e in the query log
	$qp_cnt(e)$	Number of times e occurs within the query log
Wikipedia Titles	$we_cnt(e)$	Does a concept e appear as a Wikipedia title?
	$wp_cnt(e)$	Number of times e occurs in a Wikipedia title.

including the so-called full dependence assumption (Metzler and Croft 2005) and other models that focus on key dependencies in the queries (Bendersky et al. 2009).

4 Concept Importance Features

In this section, we describe the features used for determining the importance of a term or a bigram in a weighted sequential dependence model. Recall that parameters $\lambda(q_i)$ and $\lambda(q_i, q_{i+1})$, which determine the concept weights, are represented as a weighted linear combination of features $g^u(q_i)$ and $g^b(q_i, q_{i+1})$. These features are defined over concepts (either terms or bigrams) and are independent of a specific document. This fact allows us to combine the statistics of the underlying document corpus with the statistics of various external data sources to achieve a potentially more accurate weighting. Accordingly, we divide the features used for concept importance weighting into two main types, based on the type of information they are using.

The first type, the *endogenous*, or collection-dependent, features are akin to standard weights used in information retrieval. They are based on collection frequency counts and document frequency counts calculated over a particular document corpus on which the retrieval is performed.

The second type, the *exogenous*, or collection-independent, features are calculated over an array of external data sources. The use of such sources was found to be beneficial for information retrieval models in previous work (Bai et al. 2008; Bendersky and Croft 2008; Lease et al. 2009). Some of these data sources provide better coverage of terms, and can be used for smoothing sparse concept frequencies calculated over smaller document collections. Others provide more focused sources of information for determining concept importance. We consider three external data sources: (i) a large collection of Web n -grams, (ii) a sample of a query log, and (iii) Wikipedia. Although there are numerous additional data sources that could be potentially used, we intentionally limit our attention to these three sources as they

Table 5.3 Summary of language modeling-based unigram and concept weighting functions. Here, $tf_{e,D}$ is the number of times concept e matches in document D , $cf_{e,D}$ is the number of times concept e matches in the entire collection, $|D|$ is the length of document D , and $|C|$ is the total length of the collection. Finally, μ is a weighting function hyperparameter that is set to 2500

Weighting

$$f_T(q_i, D) = \log \left[\frac{tf_{q_i,D} + \mu \frac{cf_{q_i}}{|C|}}{|D| + \mu} \right]$$

$$f_O(q_i, q_{i+1}, D) = \log \left[\frac{tf_{\#1(q_i, q_{i+1}), D} + \mu \frac{cf_{\#1(q_i, q_{i+1})}}{|C|}}{|D| + \mu} \right]$$

$$f_U(q_i, q_{i+1}, D) = \log \left[\frac{tf_{\#uw8(q_i, q_{i+1}), D} + \mu \frac{cf_{\#uw8(q_i, q_{i+1})}}{|C|}}{|D| + \mu} \right]$$

are available for research purposes, and can be easily used to reproduce the reported results.

The first source, *Google n-grams corpus*¹, contains the frequency counts of English n -grams generated from approximately 1 trillion word tokens of text from publicly accessible Web pages. We expect these counts to provide a more accurate frequency estimator, especially for smaller corpora, where some concept frequencies may be underestimated due to the collection size.

In addition, we use a large sample of a query log consisting of approximately 15 million queries². We use this data source to estimate how often a concept occurs in user queries. Intuitively, we assume a positive correlation between an importance of a concept for retrieval and the frequency with which it occurs in queries formulated by search engine users.

Finally, the third external data source is a snapshot of Wikipedia article titles³. Due to the large volume and the high diversity of topics covered by Wikipedia (~ 3 million articles in English alone), we assume that important concepts will often appear as (a part of) article titles in Wikipedia.

Table 5.2 details the statistics used for determining concept weights. As described above, these statistics are based either on the collection or on one of the external data sources. These statistics are used to compute term and bigram features ($g^u(q_i)$ and $g^b(q_i, q_{i+1})$, respectively) in the weighted sequential dependence model (see Eq. 5.5).

For computing the term features, we calculate the statistics presented in the Table 5.2 for all query terms q_i . This provides us with 7 features $g^u(q_i)$ for determining term importance weights.

To compute the bigram features, we calculate the statistics presented in the Table 5.2 for all sequential query term pairs q_i, q_{i+1} . For computing collection statistics, we use both the “exact phrase” matches and “unordered window” matches,

¹ Available from the Linguistic Data Consortium catalog.

² Available as a part of Microsoft 2006 RFP dataset.

³ Available at: <http://download.wikimedia.org/enwiki/>.

as described in Table 5.3. In addition, as bigram features, we compute a ratio $\frac{s(q_i q_{i+1})}{s(q_i)s(q_{i+1})}$ for every statistic s in the Table 5.2. Overall, the combination of the above statistics, provides us with 18 features $g^b(q_i, q_{i+1})$ for determining bigram importance weights.

For each of the data sources, we use a number of standard functions to calculate the features. For endogenous features, we use collection frequency (cf) and document frequency (df). We calculate these functions for both unigram, exact bigram matches ($\#ow1(b_1, b_2)$) and unordered bigram matches within a window of fixed size ($\#uw8(b_1, b_2)$). For exogenous features, we use exact n -gram matches ($e_cnt(\cdot)$) and partial n -gram matches ($p_cnt(\cdot)$) (note that by definition $p_cnt(\cdot) \geq e_cnt(\cdot)$, and for the Wikipedia titles $e_cnt(\cdot)$ is a binary feature).

5 Evaluation

This section presents the experimental results of the method just described. We start by detailing the experimental set-up in Sect. 5.1. Next, in Sect. 5.2, we perform a comprehensive evaluation of the method using several publicly available corpora used at the Text REtrieval Conference (TREC), including newswire and Web collections. Finally, to illustrate the benefits of the approach for Web search, in Sect. 5.3 we test the performance of the method using a proprietary Web corpus and a large sample of user queries.

5.1 Experimental Setup

The retrieval experiments are set up as follows. For all TREC collections, we obtain an initial list of top-1000 results retrieved by an unweighted sequential dependence model. This initial ranking provides a very competitive baseline, as the sequential dependence model was consistently shown to outperform the standard bag-of-word models (Lease 2009; Metzler and Croft 2005). We append all the non-retrieved relevant documents to the top-1000 list, and use this set of results for training and evaluating all the compared retrieval models.

For the proprietary Web corpus, we only index Web pages that have relevance judgments for the query samples. Training and evaluation of the retrieval models is done using this set of judged Web pages, which is a common evaluation practice for this type of test collection. There are, on average, 27 judgments per query.

We compare the performance of the weighted sequential dependence model (WSD) to two baseline retrieval models. The first is the query-likelihood model (Ponte and Croft 1998) (QL), a standard bag-of-words retrieval model implemented by the Indri search engine. The second is the unweighted sequential dependence model (SD). All the initial retrieval parameters are set to default Indri values, which reflect the best-practice settings. All the training/evaluation is done using five fold

cross-validation. The statistical significance of the differences in the performance is determined using a two-sided Wilcoxon sign test, with $\alpha < 0.05$.

We measure the performance using standard retrieval metrics for TREC and Web corpora. For TREC corpora, which uses binary relevance judgments, we use precision at top 10 documents retrieved ($P@10$) and mean average precision (MAP) at rank 1000. See Appendix B for a more detailed description of these measures. When estimating the parameters for the WSD model, we directly optimize MAP (see the next chapter for further details).

For the Web corpus, which uses graded relevance judgments, we use the discounted cumulative gain measure (DCG) (Järvelin and Kekäläinen 2002) at ranks 5 and at the total depth of the ranked list. Relevance is judged as either Perfect, Excellent, Good, Fair, or Bad. The corresponding DCG gains for these grades are 10, 7, 3, 0.5, and 0, respectively. In the direct optimization of the weighted dependence model, we use normalized DCG as the target metric.

5.2 TREC Evaluation

In this section, we describe the retrieval results obtained by the model on three standard TREC collections. A summary of the corpora used for these experiments is shown in Appendix A. We note that collections vary both by type (ROBUST04 is a newswire collection, while W10g and GOV2 are Web collections), number of documents and number of available topics, thus providing a diverse experimental set-up for assessing the robustness of the weighted dependence model.

In the evaluation we use both the *title* and the *description* portions of TREC topics as queries. *Title* queries are generally short, and can be viewed as a keyword queries on the topic. *Description* queries are generally more verbose and syntactically richer natural language expressions of the topic. For instance queries *pet therapy* and *How are pets or animals used in therapy for humans and what are the benefits?* are examples of title and description queries on the same topic, respectively.

5.2.1 Retrieval Results

Table 5.4 shows the summary of the retrieval results for the three TREC collections on both *title* and *description* queries. It is evident that both sequential dependence models (SD and WSD) significantly outperform the query-likelihood model QL in almost all the cases on all the metrics. This verifies the positive impact of term dependencies on the retrieval performance.

From the two sequential dependence models, weighted sequential dependence model (WSD) significantly outperforms the unweighted one (SD) on all collections in terms of MAP (which is used as the metric for direct optimization). The gains in MAP range between 1.6% and 19.5%, and are statistically significant for all collections and both query types.

Table 5.4 Comparison of retrieval results for *title* (top table) and *description* (bottom table) TREC queries with query-likelihood (QL), sequential dependence model (SD) and weighted sequential dependence model (WSD). Numbers in parentheses indicate % improvement in MAP over QL/SD (if available)

<i>title</i>	ROBUST04		W10g		GOV2	
	P@10	MAP	P@10	MAP	P@10	MAP
QL	0.4225	0.2493	0.2560	0.1904	0.5342	0.3019
SD	0.4410*	0.2661*	0.2890*	0.2063*	0.5785*	0.3247*
WSD	0.4462*	0.2721* _†	0.2890*	0.2220* _†	0.5779*	0.3338* _†

<i>desc</i>	ROBUST04		W10g		GOV2	
	P@10	MAP	P@10	MAP	P@10	MAP
QL	0.4269	0.2507	0.3270	0.1971	0.5168	0.2606
SD	0.4177	0.2558	0.3610*	0.2032	0.5356	0.2694*
WSD	0.4269	0.2718* _†	0.3710*	0.2523* _†	0.5181	0.2738* _†

* Statistically significant difference with QL

† Statistically significant difference with SD

It is interesting to note that even on P@10, which was not directly optimized for, WSD is more effective than SD in all but two comparisons (P@10 for GOV2). The gains observed are as high as 2.7% for P@10. We expect that even higher gains for P@10 can be attained by WSD by directly training the model for these measures of interest rather than MAP.

5.2.2 Feature Analysis

In this section we perform a detailed feature analysis, in order to identify the key elements in the success of the weighted sequential model, as compared to its unweighted counterpart.

Unigrams and Bigrams Table 5.5 compares the impact on the retrieval effectiveness of the importance weights assigned by WSD to either unigrams or bigrams in the sequential dependence model. Recall that the weighted sequential dependence model WSD is derived from its unweighted counterpart by replacing the static sequential dependence parameters λ_T , λ , and λ_U with concept dependent parameters $\lambda(q_i)$ and $\lambda(q_i, q_{i+1})$, as shown in Eq. 5.5.

The WSD-UNI model, shown in Table 5.5, is obtained by replacing λ_T with the term dependent $\lambda(q_i)$, while fixing the values of λ_O and λ_U to those of the unweighted sequential dependence model. Alternatively, WSD-BI model is obtained by replacing λ_O and λ_U with the term dependent $\lambda(q_i, q_{i+1})$, while fixing the value of λ_T .

Table 5.5 Comparison of retrieval results for *title* (left) and *description* (right) TREC queries with either only unigram features (WSD-UNI), only bigram features (WSD-BI) or both

<i>title</i>	ROBUST04	W10g	GOV2	<i>desc</i>	ROBUST04	W10g	GOV2
WSD	0.2721	0.2220	0.3338	WSD	0.2718	0.2523	0.2738
WSD-UNI	0.2685 _†	0.2188	0.3343	WSD-UNI	0.2717	0.2486	0.2677 _†
WSD-BI	0.2675 _†	0.2065 _†	0.3258	WSD-BI	0.2602 _†	0.2043 _†	0.2700

† Statistically significant difference with WSD

Table 5.5 compares the performance of both WSD-UNI and WSD-BI models to the performance of the fully weighted sequential dependence model (WSD). We note that while, in general, both WSD-UNI and WSD-BI outperform SD, in most cases WSD-UNI outperforms WSD-BI as well. This indicates that a unigram weighting has more impact on the retrieval performance than the bigram weighting. This result is in line with previous results reported by Lease for TREC collections (Lease 2009), which showed that by solely weighting unigrams, one can significantly outperform the unweighted sequential model baseline.

Another important finding shown in Table 5.5, is that WSD, which combines both unigram and bigram weights, outperforms WSD-UNI in 5 out of 6 comparisons, and always outperforms WSD-BI. In addition, WSD attains statistically significant differences in comparison with WSD-UNI for *description* queries on a large Web collection GOV2. This fact underscores the importance of weighting for all the concepts in the sequential dependence model.

Endogeneous and Exogenous Features Recall from Sect. 4 that WSD uses two types of features for estimating concept importance: endogeneous (collection-dependent) and exogeneous (collection-independent). While applying collection-dependent features for term weighting has been extensively studied in traditional information retrieval (Salton and Buckley 1988), the research on combining them with external sources of information is more recent (Bendersky and Croft 2008; Lease et al. 2009). Therefore, it is interesting to examine the contribution of each of these feature types to the overall model performance.

Table 5.6 compares the performance of the weighted sequential dependence model when either only endogeneous (WSD-ENDO) or only exogeneous (WSD-EXO) features are used to the performance of the fully weighted sequential dependence model (WSD). It is evident from Table 5.6 that using either the endogeneous or the exogeneous features results in comparable performance, and both of them outperform the unweighted dependence model. This indicates that both of these features are useful for learning the optimal weights for WSD. In both cases, however, their combination results in gains in MAP in 5 out of 6 cases. In addition, we found that both WSD-ENDO and WSD-EXO display statistically significant differences with WSD on a large Web collection GOV2 for *description* queries.

Table 5.6 Comparison of retrieval results for *title* (left) and *description* (right) TREC queries with either only endogenous features (WSD-ENDO), only exogenous features (WSD-EXO) or both

<i>title</i>	ROBUST04	W10g	GOV2	<i>desc</i>	ROBUST04	W10g	GOV2
WSD	0.2721	0.2220	0.3338	WSD	0.2718	0.2523	0.2738
WSD-ENDO	0.2685 _†	0.2176	0.3281	WSD-ENDO	0.2707	0.2328	0.2695 _†
WSD-EXO	0.2701	0.2119 _†	0.3354	WSD-EXO	0.2733	0.2468	0.2733 _†

† Statistically significant difference with WSD

Table 5.7 Comparison of retrieval results over a sample of Web queries with query-likelihood (QL), sequential dependence model (SD) and weighted sequential dependence model (WSD). Numbers in parentheses indicate % improvement in DCG over QL/SD (if available)

	Len-2		Len-3		Len-4+	
	DCG@5	DCG	DCG@5	DCG	DCG@5	DCG
QL	2.231	10.750	2.290	8.204	1.691	5.844
SD	2.733	11.539	2.971	9.139	2.383	6.681
WSD	2.754	11.585	2.929	9.087	2.443	6.741

All the differences are statistically significant

5.3 Large-Scale Web Evaluation

Previous research has shown that modeling sequential term dependencies has a significant positive impact on retrieval performance in the Web search setting (Bai et al. 2008; Metzler and Croft 2005; Mishne and de Rijke 2005). Given the retrieval performance gains obtained from using the weighted variant of the sequential dependence model demonstrated on TREC collections in the previous section, the following set of experiments explores whether these gains can be directly transferred into a Web search setting. To this end, in this section we test the ranking with a weighted sequential dependence model on a proprietary Web corpus provided by a large commercial search engine.

To differentiate between the effect of concept weighting on queries of varying length, as was done in the case of TREC corpora, we divide the queries into three groups based on their length. Length is defined as a number of word tokens separated by space in the query. The first group of queries (*Len-2*) includes very short queries of length two. The second group (*Len-3*) includes queries of length three. The third group (*Len-4+*) consists of more verbose queries of length varying between four and twelve. While the queries in the first two groups mostly have a navigational intent, the queries in the third group tend to be more complex informational queries. For each group, we randomly sample 1,000 Web search queries for which relevance judgments are available. We then train and evaluate (using five fold cross-validation) a separate sequential dependence model and weighted sequential dependence model for each group.

Table 5.8 Concept weights generated for query “information about peer gynt suite”

Concept	Weight
information	0.0210
about	0.0193
peer	0.1591
gynt	0.4114
suite	0.1555
information about	−0.0399
about peer	0.0122
peer gynt	0.0891
gynt suite	0.0399

5.3.1 Retrieval Results

Table 5.7 shows the summary of the retrieval results on the three query groups. To demonstrate the impact on the relevance at the top ranks of the retrieved list we report the DCG@5. To demonstrate the overall ranking quality, we report the results for DCG at unlimited depth (denoted DCG).

Table 5.7 demonstrates two important findings. First, including term dependence information is highly beneficial for queries of all lengths. SD attains up to 12.5% improvement over QL, which is a bag-of-words model. This result is highly significant, given the large size of the query set. Second, concept weighting results in significant improvements for longer (*Len-4+*) queries, and its performance is comparable for shorter queries to the performance of the unweighted dependence model (slight improvement on *Len-2* and slight decrease in performance on *Len-3*). For group *Len-4+*, WSD attains improvement of close to 2.5% for DCG@5, a highly significant improvement, especially when taking into account the importance of relevance at top ranks for the Web search task.

5.3.2 Feature Analysis

Similarly to the feature analysis performed in Sect. 5.2.2 for TREC corpora, in this section we analyze the importance of different weights and features in the weighted sequential model for the Web corpus.

Unigrams and Bigrams Table 5.9 compares the impact on the retrieval effectiveness of the importance weights assigned by WSD to either unigrams or bigrams in the sequential dependence model. Notice that, contradictory to what was observed in Table 5.5 for the TREC data, the bigram weights have more impact on the retrieval effectiveness than the unigram weights. For short queries in groups *Len-2* and *Len-3*, using bigram weights alone and omitting the unigram weights results in a slightly higher DCG at all measured ranks than using the fully weighted dependence model.

Table 5.9 Comparison of retrieval results for a sample of Web queries with either only unigram features (WSD-UNI), only bigram features (WSD-BI) or both

	Len-2		Len-3		Len-4+	
	DCG@5	DCG	DCG@5	DCG	DCG@5	DCG
WSD	2.754	11.585	2.929	9.087	2.443	6.741
WSD-UNI	2.743	11.556	2.963	9.132	2.379	6.677
WSD-BI	2.758	11.602	2.967	9.132	2.409	6.711

All the differences are statistically significant

Table 5.10 Comparison of retrieval results for a sample of Web queries with either only endogenous features (WSD-ENDO), only exogenous features (WSD-EXO) or both

	Len-2		Len-3		Len-4+	
	DCG@5	DCG	DCG@5	DCG	DCG@5	DCG
WSD	2.754	11.585	2.929	9.087	2.443	6.741
WSD-ENDO	2.687	11.487	2.924	9.085	2.455	6.760
WSD-EXO	2.749	11.575	2.919	9.079	2.439	6.732

All the differences are statistically significant

A likely explanation for this effect is the dominance of navigational intent for short queries in Web search. TREC topics, including the short *title* queries, mostly have an informational intent and often consist of several separate concepts of unequal importance (e.g., “abandoned mine reclamation”). Short two-three word Web queries, on the other hand, often consist of a single navigational bigram (“yahoo mail”), or a bigram followed by an auxiliary term (“yahoo mail login”).

Compared to the first two groups, using both unigram and bigram weights in queries in group *Len-4+* results in a better performance than using either of them alone, which is in line with the results for the TREC collections. We hypothesize that this stems from the fact that a higher percentage of these queries have an informational intent, and they contain both unigram and bigram concepts of varying importance (“best metal songs of the 1980s”).

Overall, as evident from Table 5.9, the impact of concept weights is influenced both by the query type and by the collection. While the weighted sequential model can naturally incorporate weighted and unweighted concepts, the optimal weighting policy has to be determined using training on the available data.

Endogenous and Exogenous Features Table 5.10 compares the performance of the weighted sequential dependence model when either only endogenous (WSD-ENDO) or only exogeneous (WSD-EXO) features are used to the performance of the fully weighted sequential dependence model (WSD). It is evident from Table 5.10 that using either the endogenous or the exogeneous features results in most cases

in comparable performance. Similarly to WSD, both of them outperform the un-weighted dependence model on queries in group *Len-4+*.

For shorter queries in the first two groups combining the two types of features results in a better performance than using either one in isolation. For queries in a group *Len-4+* using endogenous features alone results in a slightly better performance than the WSD, however the difference is relatively minor (0.3% improvement of the DCG metric). In addition, the impact of exogeneous features on the overall retrieval performance of the Web queries might be potentially boosted by including additional external sources, instead of just three, as is currently done. For instance, a larger and a more recent sample of user queries than the one used in this study could be employed.

As a general “rule of thumb” strategy, a combination of both endogenous and exogenous features appears to be the preferred option both for the TREC and for the Web corpora.

Some illustrative examples of learned concept weights are shown in Table 5.8.



<http://www.springer.com/978-3-642-22897-1>

A Feature-Centric View of Information Retrieval

Metzler, D.

2011, XII, 168 p., Hardcover

ISBN: 978-3-642-22897-1