Semi-Parametric and Non-parametric Term Weighting for Information Retrieval

Donald Metzler[1] and Hugo Zaragoza[1]

Yahoo! Research
{metzler,hugoz}@yahoo-inc.com

Abstract. Most of the previous research on term weighting for information retrieval has focused on developing specialized parametric term weighting functions. Examples include $TF.IDF$ vector-space formulations, BM25, and language modeling weighting. Each of these term weighting functions takes on a specific parametric form. While these weighting functions have proven to be highly effective, they impose strict constraints on the functional form of the term weights. Such constraints may possibly degrade retrieval effectiveness. In this paper we propose two new classes of term weighting schemes that we call semi-parametric and non-parametric weighting. These weighting schemes make fewer assumptions about the underlying term weights and allow the data to speak for itself. We argue that these robust weighting schemes have the potential to be significantly more effective compared to existing parametric schemes, especially with the growing amount of training data becoming available.

1 Introduction

A great deal of research has been devoted to developing highly effective term weighting schemes for information retrieval. Some examples include $tf.idf$ [1], pivoted length normalization [2], BM25 [3], language modeling [4], divergence from randomness [5], axiomatic weighting [6], genetic programming [7], and impact-based weighting [8]. Despite their differences, all of these approaches share one thing in common – they all assume that the underlying term weighting function takes on a specific functional form. Therefore, most, if not all, of the previously proposed term weighting schemes for information retrieval can be considered *parametric*.

Parametric term weighting functions restrict expressiveness and, possibly, effectiveness because the resulting weights are biased, *a priori*, to conform to the chosen functional form. Indeed, there is evidence that term weighting functions with more degrees of freedom, and therefore fewer functional restrictions, are more effective than weighting functions with fewer degrees of freedom. One classical example is that a well-tuned BM25, which has two parameters ($k1$, b) typically outperforms language modeling with Dirichlet smoothing, which has only just one parameter (μ). Of course, it is difficult to prove that the improved effectiveness is due to the extra degree of freedom, but it is certainly a possibility.

The current state-of-the-art term weighting schemes were developed when collections were small and training data was sparse. However, things are quite different now. Collections are larger than ever and training data is abundant, in the form of human judgments and click logs. We either have reached, or will soon reach, the point where we allow the data to "speak for itself", thereby eliminating the need to resort to parametric term weighting functions. While there has been some recent interest in developing *parameter-free* weighting functions [5], we believe that such models are better suited for "cold start" retrieval systems that have no training data, and that richer models with multiple parameters will be significantly more effective when training data is available.

In this paper, we look beyond traditional parametric term weighting functions, to more expressive weighting functions that have fewer functional constraints. Our primary contribution is two classes of term weighting functions that we call *semi-parametric* and *non-parametric* functions. As we will show, our proposed weighting functions mark a significant departure from previous term weighting research. We hypothesize this new direction could result in significant improvements in retrieval effectiveness and promote renewed interest in term weighting research.

The remainder of this paper is laid out as follows. First, in Section 2 we survey previous term weighting research. In Section 3 we describe our semi-parametric and non-parametric term weighting frameworks. Then, in Section 4 we discuss how the parameters of the proposed models can be estimated. Finally, we conclude the paper in Section 5 and describe possible avenues for future work.

2 Related Work

We now briefly describe three popular existing term weighting schemes. The first two, BM25 and language modeling, are based on probabilistic retrieval models. Such models are inherently parametric, because each assumes terms are distributed according to some parametric statistical model, such as a multinomial or mixture of Poissons. The other term weighting scheme that we discuss, which is based on ordinal term weighting, makes fewer assumptions about the data, although the weights still take on a parametric form. We will show how we can easily combine, and build upon each of these to develop even more robust weighting schemes.

2.1 BM25 Term Weighting

The classical probabilistic retrieval model ranks documents in decreasing order of likelihood of relevance, in accordance with the Probability Ranking Principle [9]. The general form of the model is:

$$S(Q,D) = P(r|q,d) \stackrel{rank}{=} \sum_{t \in Q \cap D} \log \frac{P(t|r)P(\bar{t}|\bar{r})}{P(\bar{t}|r)P(t|\bar{r})}$$

where t and $\bar{t}$ represent the occurrence and non-occurrence of term t, respectively. Furthermore, $P(t|r)$ and $P(t|\overline{r})$ represent the likelihood of event t in the relevant and non-relevant classes of documents, respectively. Previous researchers have made various distributional assumptions for these distributions. Assuming a multivariate-Bernoulli results in the Binary Independence Retrieval model [10], whereas the assumption of a 2-Poisson model, after some simplifying assumptions, results in the BM25 model [3]. The BM25 ranking function has the following form:

$$S(Q,D) = \sum_{t \in Q \cap D} \frac{tf_{t,D}}{k_1(1-b+b\frac{|D|}{|D|_{avg}}) + tf_{t,D}} idf_t$$

where k_1 and b are free parameters that control for term frequency saturation and document length normalization.

2.2 Language Modeling Term Weighting

The language modeling framework for information retrieval is another widely used probabilistic model [4]. It is based on the assumption that documents can be topically represented by a probabilistic model called a *document model*. Document models are commonly modeled as multinomial distributions that are smoothed against the collection model in various ways [11]. Queries are similarly represented as *query models*, which are also typically modeled as multinomial distributions. Query models can be estimated in various ways, including maximum likelihood estimation, local blind feedback [12, 13], or global blind feedback [14].

Documents are ranked according to the similarity between the query and document models. Kullback-Leibler divergence is often used as the (dis)similarity measure. Therefore, documents are ranked according to:

$$\begin{aligned} S(Q,D) &= -KL(P(\cdot|Q), P(\cdot|D)) \\ &\stackrel{rank}{=} \sum_{t \in \mathcal{V}} P(t|Q) \log P(t|D) \end{aligned}$$

where $P(\cdot|Q)$ and $P(\cdot|D)$ are the query and document models, respectively. Although the sum goes over the entire vocabulary $\mathcal{V}$, most query models are sparse, which significantly reduces the computational complexity. Language modeling term weights are parametric, where the parameterization depends on the type of smoothing used.

2.3 Ordinal Term Weighting

Document-centric impacts, originally proposed by Anh and Moffat [15], assign weights to document and query terms based on their relative importance to other terms. Terms are weighted as follows. First, given document, a total ordering of the (unique) terms is imposed. This is typically done by sorting the terms according to their term frequency and breaking ties with inverse document

frequency. Once the terms have been totally ordered, they are partitioned into k bins. Here, it is assumed that all terms within bin k are equally important and that terms in bin $i+1$ are more important than those in bin i. Essentially, the total ordering is transformed into a partial ordering. This binning procedure is typically done by geometrically binning the terms, where a small number of terms are considered "most important" (i.e., assigned to bin k) and a large number are considered "least important" (i.e., assigned to bin 1). A similar, yet slightly different, procedure is done on the query side to map query terms to bins.

Once terms have been assigned to bins, they must be assigned a weight for the purpose of scoring. Anh and Moffat, for simplicity, assign integral weights to the bins, with all of the terms within bin i being assigned weight i. We denote the weight for term w in documents and queries as $w_{bin(t,Q)}$ and $w_{bin(t,D)}$, respectively. Given a query Q and a document D, the score assigned under the Anh and Moffat model is:

$$\begin{aligned} S(Q,D) &= \sum_{w \in Q \cap D} w_{bin(t,Q)} w_{bin(t,D)} \\ &= \sum_{w \in Q \cap D} bin(t,Q) bin(t,D) \end{aligned}$$

where $bin(t,Q)$ and $bin(t,D)$ is the bin that term w is assigned in the query and document, respectively, and the equivalence follows from the fact that integral weights are used (i.e., $w_{bin(t,Q)} = bin(t,Q)$).

Anh and Moffat show that a very small number of bins is sufficient to achieve good retrieval effectiveness, but not as good as BM25 or language modeling. Adding more bins tends not to significantly improve effectiveness. Furthermore, fewer bins results in smaller indexes and considerably faster query execution times. Therefore, 4, 8, or 16 bins are often used in practice. One reason why the method tends to have sub-standard retrieval effectiveness compared to BM25 and language modeling is because of the choice of integral weights, which is an oversimplification. It has been shown that automatically learning the weights can lead to improvements in retrieval effectiveness [16].

3 Term Weighting

The term weighting problem for information retrieval requires a system to assign weights to word/query and word/document pairs. The weight should accurately reflect the importance of the term with respect to the query or document, with higher weights indicating more important terms. Of course, the ultimate goal is to assign term weights in such a way that the underlying retrieval model is highly effective according to some metric.

More formally, given a vocabulary $\mathcal{V}$, a set of documents $\mathcal{D}$, and a set of queries $\mathcal{Q}$ the term weighting problem requires us estimate $W \in \mathbb{R}^{|\mathcal{V}|} \times \mathbb{R}^{|\mathcal{Q}|} \times \mathbb{R}^{|\mathcal{D}|}$, where entry $w_{t,Q,D}$ corresponds to the weight of term t assigned to document D for query Q. We may also wish to condition the term weights on the

user, time, or various other factors. However, for simplicity, we ignore these factors, as they would only complicate things and make our problem even more difficult to solve. Our goal is to find the W that, when used in conjunction with the underlying (yet to be specified) ranking function, optimizes some evaluation metric of interest.

Some care must be taken when solving this problem, because there are a total of $|\mathcal{V}||\mathcal{D}||\mathcal{Q}|$ parameters. Obviously this estimation problem is infeasibly large for any non-trivial search task. This is one reason why parametric term weighting schemes have been so popular and appealing. Such schemes effectively reduce this enormous solution space down to just a handful of parameters.

In this paper, we assume that the underlying ranking function has the following form:

$$S(Q, D) = \sum_{t \in Q} w_{t,Q,D}$$

where Q is a query, D is a document, and $w_{t,Q,D}$ is the weight of t with respect to Q and D. We refer to this as the *joint form*, since the weight $w_{t,Q,D}$ depends jointly on Q and D.

While the joint formulation is the most generic way of representing most ranking functions, a vast majority of the widely used retrieval models, including BM25 and language modeling, can be written in a slightly simpler form, as follows:

$$S(Q, D) = \sum_{t \in Q} w_{t,Q} w_{t,D}$$

We refer to this as the *factored form*, since the weight $w_{t,Q,D}$ can be factored into the product of a weight for t in Q ($w_{t,Q}$) and a weight for t in D ($w_{t,D}$). This factorization reduces the size of the parameter space from $|\mathcal{V}||\mathcal{D}||\mathcal{Q}|$ to $|\mathcal{V}|(|\mathcal{D}| + |\mathcal{Q}|)$, which unfortunately is still infeasibly large.

Solving the term estimation problem, in either the joint or factored form, is simply not possible. Therefore, we must resort to measures that reduce the dimensionality of the problem while still maintaining expressiveness and effectiveness. We will now describe a general framework for reducing the term weighting dimensionality. We will then show how, within this framework, it is possible to develop whole new classes of term weighting schemes that make far fewer assumptions about the data than current approaches.

3.1 Dimensionality Reduction

There are various ways to reduce the dimensionality of the term weighting problem. Previous researchers have used latent semantic analysis [17, 18], topic modeling [19], and various parametric functions (see Section 2) for this purpose. Rather than subscribe to any one of these approaches, we present the dimensionality reduction problem more generally, since we believe that information retrieval specific techniques may be superior to the previously proposed approaches.

Our proposed dimensionality reduction framework is very similar in spirit to the binning strategies used by Anh and Moffat [8]. In fact, their binning strategies can be used directly within our framework. However, as we will show, our framework is more general and results in a more formal estimation procedure.

When scoring a document D with respect to a query Q, we first bin the terms in the query and then bin the terms in the document. This binning can be thought of as a form of dimensionality reduction or massive parameter tying. We assume that the query terms are mapped (deterministically) into k_Q bins and document terms are mapped (deterministically) into k_D bins, where k_Q and k_D are fixed *a priori* and are constant across all queries and documents. Given the binned query terms and binned document terms, the retrieval status value (i.e., score) is computed as follows:

$$S(Q, D) = \sum_{t \in Q} \hat{w}_{bin(t,Q),bin(t,D)}$$

where $bin(t, Q)$ and $bin(t, D)$ are the query and document bins, respectively for term t, and $\hat{w}_{i,j}$ is an entry in $\hat{W} \in \mathbb{R}^{k_Q} \times \mathbb{R}^{k_D}$, which is a lower dimensional approximation of the full weight specification W. This approximation has $k_Q k_D$ parameters which is substantially smaller than both $|\mathcal{V}||\mathcal{D}||\mathcal{Q}|$ and $|\mathcal{V}|(|\mathcal{D}|+|\mathcal{Q}|)$.

It is important to note that, although binning and weighting are done on the term-level, the resulting models will not necessarily be *bag of words* models. The binning strategies may use contextual information, such as surrounding words, formatting, document structure, etc. Therefore, unlike traditional bag of words models, where a random permutation of the terms will result in the same term weights, our framework provides a simple mechanism for contextual term weighting.

Additionally, it should be clear that Anh and Moffat's model is a special case of this model, where $w_{bin(t,Q),bin(t,D)} = w_{bin(t,Q)} w_{bin(t,D)}$ and binning is done according to their proposed methods. However, as we will soon show, this dimensionality reduction framework can be applied to term weighting in a variety of interesting ways.

In order to use our proposed term weighting scheme we must define a query term binning strategy, a document term binning strategy, and a weighting $\hat{W}$. We will now describe the details of each of these steps.

Query Term Binning There are various ways of perform query-side binning, including:

- Anh and Moffat query binning, which bins the query terms into $|Q|$ bins. The query term with the largest IDF is assigned to bin $|Q|$, the term with the next largest IDF is assigned to bin $|Q| - 1$, and so on, with the term with the smallest IDF being assigned to bin 1.
- Query-independent IDF binning. Rather than sorting terms within the query, we can bin terms according to their IDF. For example, we can assign the 25% of terms with the largest IDF out of the entire vocabulary to bin 4, the

terms with the 25% next largest IDF to bin 3, and so on, with the 25% of terms with the lowest IDF to bin 1. There may be other ways of performing this binning, based on the number of documents that each term occurs in.
- Lexical binning. One may also use lexical information to assign words to bins in different ways. For example, some frequent and important words may be assigned their own bin, or bins could be assigned to types of words based on their length, their part of speech, their lexical semantics, etc.

Document Term Binning Furthermore, several possible ways to bin document terms are:

- Anh and Moffat document binning, as described in Section 2.3.
- Binning based on existing term weighting schemes. For example, one can sort all of the terms within a document according to BM25, then assign terms to bins geometrically or uniformly. This is similar to the Anh and Moffat approach, except sorting is done in a slightly different manner.
- Lexical binning. As described in the query term binning section, it may be possible to assign terms to bins based on natural language processing and/or other linguistic properties.

3.2 Non-Parametric Term Weighting

After query and document terms have been assigned bins, the final step is to determine the weightings $w_{bin(t,Q),bin(t,D)}$ for each combination of bins. As we described before, this problem has $k_Q k_D$ parameters. Depending on the number of bins, it may actually be possible to learn the term weighting directly, without the need to impose any parameterized form on the problem. When parameters are estimated in this way, we call the resulting weighting *non-parametric*, since the weights are learned directly from the data and have no pre-determined functional form.

Figure 1 summarizes the non-parametric term weighting problem. In this example, there are 3 query term bins ($k_Q = 3$) and 4 document term bins ($k_D = 4$). This results in a total of 12 parameters that can be directly estimated. We will describe methods for solving this estimation problem in Section 4.1.

The benefit of such a term weighting scheme is that it assumes no functional form for the term weights and learns directly from the data. However, this method relies on having very reliable, high quality query and document term binning functions. It may be the case that many bins will be necessary to accurately represent the importance of terms within queries and documents, potentially resulting in too many parameters to reliably estimate. The optimal binning strategy and number of bins is an open question that requires further investigation.

3.3 Semi-Parametric Term Weighting

In non-parametric term weighting, no functional form was assumed for the weights. As we discussed, depending on the binning strategies applied, this may

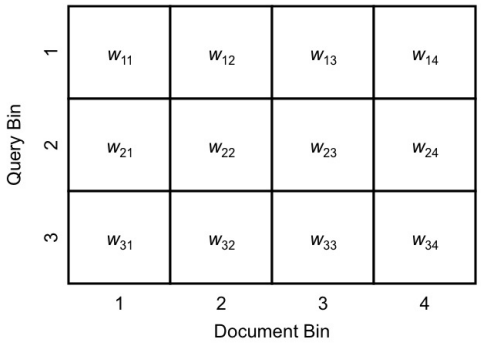

Fig. 1. Summary of the non-parametric term weighting problem after dimensionality reduction.

result in too many parameters. One way of combating this issue is to solve a more constrained version of the term weighting problem that assumes some functional form for $\hat{w}_{bin(t,Q),bin(t,D)}$ but has parameters that depend on $bin(t, D)$ and $bin(t, D)$. We call this class of term weighting schemes *semi-parametric*, since the weighting function takes on a parametric form, but the parameters are not fixed across all term, query, document pairs, as in traditional parametric models. This scheme allows different classes of query and document terms to be weighted vastly differently, based on their characteristics.

As a concrete example, let us consider a semi-parametric form of BM25 weighting, which we propose as follows:

$$\hat{w}_{bin(t,Q),bin(t,D)} = \frac{tf_{t,D}}{k_{bin(t,Q)}(1 - b_{bin(t,D)} + b_{bin(t,D)}\frac{|D|}{|D|_{avg}}) + tf_{t,D}} idf_t$$

Here, it is important to notice that the term frequency saturation parameter k depends on the $bin(t, Q)$ and the document length normalization parameter b depends on $bin(t, D)$. In this way, we can model the fact that different types of terms tend to saturate in importance differently than others. For example, it may be that a single occurrence of a named entity is enough to saturate the term frequency, whereas many occurrences of a medium-IDF term may be required. Similarly, $bin(t, D)$ may be defined to be term independent and simply act to partition documents along some dimension, such as their length. In this way, we could have a document length-dependent setting for b.

A similar type of idea could be applied within the language modeling framework. For example, the following semi-parametric version of Dirichlet smoothing could be used:

$$\hat{w}_{bin(t,Q),bin(t,D)} = \alpha_{bin(t,Q)} \log \frac{tf_{t,D} + \mu_{bin(t,D)} P(t|C)}{|D| + \mu_{bin(t,D)}}$$

Within this formulation we have a different smoothing parameter μ for every bin $bin(t, D)$. This could allows us to use a different smoothing parameter for different classes of document lengths (e.g., short, medium, long), in a similar manner to the semi-parametric b just proposed to be used in conjunction with BM25. We also define a parameter α that depends on $bin(t, Q)$. This can be used to weight different classes of query terms differently. For example, we may want to upweight nouns and downweight certain adjectives, definitives, etc. This can all be accomplished by defining appropriate query and document term binning strategies.

It may be possible to learn more generic weighting functions in this way, as well. For example, a linear or non-linear regression model may be used as the parametric form, with the model parameters depending on $bin(t, Q)$ and $bin(t, D)$. Similar semi-parametric forms can be used to estimate the weight of a term in the query or even a joint weight that depends on both the query and the document (i.e., $w_{t,D,Q}$).

3.4 Parametric Term Weighting

It should now be clear that standard parametric term weighting functions are special cases of our framework that trivially assign all query and document terms to the same bin (i.e., $k_D = k_Q = 1$). Therefore, our framework can be used to expand the expressiveness of any existing term weighting scheme by providing a mechanism to use more fine-grained parameters, which we hypothesize will lead to improvements in retrieval effectiveness.

4 Estimating $\hat{W}$

4.1 Non-Parametric Weight Estimation

The ideal situation is to estimate $\hat{W}$ in a supervised manner using training data in the form of human judgments or click data. Estimating $\hat{W}$ can be transformed into a standard linear "learning to rank" problem. It can be shown that $S(Q, D)$, as defined above, can be rewritten as:

$$S(Q, D) = \sum_{i=1}^{k_Q} \sum_{j=1}^{k_D} |\{w \in Q : bin(t, Q) = i, bin(t, D) = j\}| \hat{w}_{i,j}$$

which is a linear function with respect to the weights $\hat{w}_{i,j}$. If we treat the $|\{w \in Q : bin(t, Q) = i, bin(t, D) = j\}|$ values as "features", then this is a standard linear learning to rank problem, by which we want to find the weights $\hat{w}_{i,j}$ that optimize for some evaluation metric, such as mean average precision or NDCG. A variety of techniques have been described for solving this problem [20, 21].

The weights $\hat{w}_{i,j}$ learned as the result of this optimization process are then used for scoring. It is important to note that while $S(Q, D)$ is parametric (i.e., linear), the term weights $\hat{w}_{i,j}$ are not, since they may take on any possible value.

Finally, we point out that although the scoring function is linear with respect to $\hat{w}_{i,j}$, the weights may be non-linear with respect to term frequency and inverse document frequency, depending on the binning strategy.

If little or no training data is available, then we can use existing term weighting functions, such as BM25, to estimate $\hat{W}$, as follows:

$$\hat{w}_{i,j} = \frac{\sum_{Q \in \mathcal{Q}} \sum_{D \in \mathcal{D}} \sum_{w \in Q \cap D : bin(t,Q)=i, bin(t,D)=j} BM25(t, D)}{\sum_{Q \in \mathcal{Q}} \sum_{D \in \mathcal{D}} |\{w \in Q \cap D : bin(t, Q) = i, bin(t, D) = j\}|}$$

where $\mathcal{Q}$ and $\mathcal{D}$ is the universe of queries and documents, respectively. This unsupervised estimate simply averages (over the entire universe of queries and document) the BM25 term weights for each entry in $\hat{W}$. Of course, it is infeasible to compute this value exactly since $|\mathcal{Q}||\mathcal{D}|$ is essentially unbounded. Instead, a reasonable estimate can be derived by sampling a small number queries and documents to compute the average over. This unsupervised estimate can also be used as a prior, or starting point, when estimating $\hat{W}$ in a supervised manner.

4.2 Parametric and Semi-Parametric Weight Estimation

Parameter estimation for parametric and semi-parametric term weighting schemes is slightly more difficult, since the functional forms are likely to be non-linear with respect to the parameters. This may pose a challenge when using standard optimization techniques. It may be possible to use an approach, such as the one described by Taylor et al. to optimize a proxy loss function, as long as the weight function is differentiable with respect to the parameters [22].

Depending on the complexity of the underlying parametric form, the evaluation metric of interest, the size of the test collection, and the number of parameters, a simple grid search or greedy search technique may work just fine. However, if the number of parameters is large, as will be the case when k_D and/or k_Q is large, then more careful optimization procedures must be devised to avoid possible overfitting and excessive computational costs.

5 Conclusions and Future Work

This paper described a spectrum of term weighting schemes for information retrieval that go beyond traditional parametric weighting. In particular we proposed semi-parametric and non-parametric weighting schemes that we hypothesize could result in more robust, more effective retrieval models. Table 1 provides a summary of the different weighting schemes that were discussed.

As we showed, *non-parametric* weighting schemes are the most generic of the three types. These weighting schemes do not assume any functional form for the weights. Instead, the weights are estimated directly. While this is the most general approach, we suspect that a large number of parameters may be necessary to provide good retrieval effectiveness in practice, and therefore a very large amount of training data may be necessary to effectively learn such

Type of Weighting	Functional Form	Parameters
Parametric	Parametric Function	Global
Semi-Parametric	Parametric Function	Dependent on w, Q, D
Non-parametric	No Functional Form	Dependent on w, Q, D

Table 1. Summary of the different types of term weighting schemes, their functional form, and their parameters.

models. However, if such data is available, we suspect that these models have the potential to yield significant improvements in retrieval effectiveness.

The next most generic class of weighting schemes are *semi-parametric*. Under this scheme, weighting functions have some parametric form, but the parameters of the weighting functions depend on the query term and document term binnings. In this way, the weights are parametric, but depending on the binning, can be adapted better to the data due to the less constrained parameterization.

Finally, *parametric* weighting schemes, which account for most, if not all, of the currently used term weighting functions are the most restrictive. In this class, weighting functions are parametric, but the parameters (if any) of the weighting scheme are global. That is, the same set of parameters are applied to all queries and documents. While the global parameters are estimated from data, the underlying weights may not be very adaptable to a wide variety of query terms and document types, thereby hindering effectiveness.

This paper was devoted entirely to the theory underlying different classes of term weighting functions. However, an important direction of future work is to understand the implications of the theory on practical information retrieval systems. In particular, we plan to explore the effectiveness of the different classes of term weighting schemes. We also plan to develop a better grasp on the usefulness of different binning strategies and the feasibility of using completely non-parametric term weighting.

References

1. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information Processing and Management **24**(5) (1988) 513–523
2. Singhal, A., Buckley, C., Mitra, M.: Pivoted document length normalization. In: Proc. 19th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval. (1996) 21–29
3. Robertson, S.E., Walker, S.: Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In: Proc. 17th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval, Springer-Verlag New York, Inc. (1994) 232–241
4. Ponte, J., Croft, W.B.: A language modeling approach to information retrieval. In: Proc. 21st Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval. (1998) 275–281
5. Amati, G., van Rijsbergen, C.J.: Probabilistic models of information retrieval based on measuring the divergence from randomness. ACM Transactions on Information Systems **20**(4) (2002) 357–389

6. Fang, H., Zhai, C.: An exploration of axiomatic approaches to information retrieval. In: Proc. 28th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval. (2005) 480–487
7. Fan, W., Gordon, M.D., Pathak, P.: A generic ranking function discovery framework by genetic programming for information retrieval. Inf. Process. Manage. **40**(4) (2004) 587–602
8. Anh, V.N., Moffat, A.: Simplified similarity scoring using term ranks. In: Proc. 28th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval. (2005) 226–233
9. Robertson, S.: The probability ranking principle in IR. Journal of Documentation **33**(4) (1977) 294–303
10. Robertson, S.E., Spärck Jones, K.: Relevance weighting of search terms. Journal of the American Society for Information Science **27**(3) (1976) 129–146
11. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to information retrieval. ACM Trans. Inf. Syst. **22**(2) (2004) 179–214
12. Lavrenko, V., Croft, W.B.: Relevance-based language models. In: Proc. 24th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval. (2001) 120–127
13. Zhai, C., Lafferty, J.: Model-based feedback in the language modeling approach to information retrieval. In: Proc. 10th Intl. Conf. on Information and Knowledge Management. (2001) 403–410
14. Berger, A., Lafferty, J.: Information retrieval as statistical translation. In: Proc. 22nd Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval. (1999) 222–229
15. Anh, V.N., Moffat, A.: Collection-independent document-centric impacts. In: Proc. Australian Document Computing Symposium. (2004) 25–32
16. Metzler, D., Strohman, T., Croft, W.B.: A statistical view of binned retrieval models. In: Proc. 30th European Conf. on Information Retrieval. (2008) 175–186
17. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. Journal of the American Society for Information Science **41** (1990) 391–407
18. Hofmann, T.: Probabilistic latent semantic indexing. In: Proc. 22nd Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval, New York, NY, USA, ACM (1999) 50–57
19. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3** (2003) 993–1022
20. Gao, J., Qi, H., Xia, X., Nie, J.Y.: Linear discriminant model for information retrieval. In: Proc. 28th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval, New York, NY, USA, ACM (2005) 290–297
21. Joachims, T.: A support vector method for multivariate performance measures. In: Proc. 22nd Proc. Intl. Conference on Machine Learning, New York, NY, USA, ACM (2005) 377–384
22. Taylor, M., Zaragoza, H., Craswell, N., Robertson, S., Burges, C.: Optimisation methods for ranking functions with multiple parameters. In: Proc. 15th Intl. Conf. on Information and Knowledge Management, New York, NY, USA, ACM (2006) 585–593