A Statistical View of Binned Retrieval Models

Donald Metzler[1], Trevor Strohman[2], and W. Bruce Croft[2]

[1] Yahoo! Research, Santa Clara, CA
[2] University of Massachusetts, Amherst, MA

Abstract. Many traditional information retrieval models, such as BM25 and language modeling, give good retrieval effectiveness, but can be difficult to implement efficiently. Recently, document-centric impact models were developed in order to overcome some of these efficiency issues. However, such models have a number of problems, including poor effectiveness, and heuristic term weighting schemes. In this work, we present a statistical view of document-centric impact models. We describe how such models can be treated statistically and propose a supervised parameter estimation technique. We analyze various theoretical and practical aspects of the model and show that weights estimated using our new estimation technique are significantly better than the integer-based weights used in previous studies.

1 Introduction

Most of the information retrieval models developed recently fall into a class of models known as parameterized retrieval models. Examples of these models are BM25 [1], language modeling [2], the axiomatic model [3], the divergence from randomness model [4], and linear discriminative models [5, 6]. At the very core of these models is some term weighting function that is composed of one or more free parameters and standard information retrieval features, such as term frequency, inverse document frequency, and document length. These term weighting functions are responsible for *quantitatively* assigning importance values to document and query terms. The standard procedure for training or tuning a parameterized model of this form is to learn a set of parameters using either supervised or unsupervised methods that maximizes some information retrieval metric.

It is important to note that the importance values (weights) are quantitative variables, and therefore, their absolute and relative values are indeed important. If term A is given double the weight of term B then we must conclude that A is two times as important as term B. This is very different than concluding that term A is *more* important than term B. Such a conclusion would require us to assume that term importance is an *ordinal* variable, rather than a quantitative one. While term weighting functions impose an implicit ordering on terms according to importance, they do not explicitly model the ordinal nature of term importances.

Recently, Anh and Moffat introduced the document-centric impact model, which represents a paradigm shift in the design of retrieval models [7]. The

model, which was experimentally shown to be both effective and highly efficient, moves away from complex parameterized term weighting functions. Instead, a method is proposed by which document terms are partitioned into a small number (e.g., fewer than 16) of bins. Each bin contains a set of terms of equal importance. For example, there may be a bin that contains all of the most important terms, another that contains less important terms, and a third that contains the least important terms. This imposes an explicit *ordering* of sets of terms (bins), instead of the implicit ordering of terms imposed by classic term weighting functions.

In this work, we present a model that can be considered to be a statistical interpretation of the document-centric impact model. Like the document-centric impact model, our model also requires binning of document terms in order to estimate term weights. However, in our model, we take a probabilistic approach that allows many of the techniques available in the language modeling literature to be used. Without a statistical interpretation, such techniques would not be as easily applied. Furthermore, such an interpretation allows us to use non-integral impacts, and estimate parameters more formally in a supervised fashion, thereby eliminating one of the more heuristic pieces of the document-centric impact model. As we will show, this statistical interpretation, along with the newly devised estimation technique consistently and significantly improves retrieval effectiveness relative to existing impact-based retrieval models.

The remainder of this paper is laid out as follows. In Section 2 we describe related models. Section 3 lays out the theoretical foundations of our model. In Section 4 the results from our experimental evaluation are presented. Finally, Section 5 concludes the paper and presents potential areas of future work.

2 Related Models

In this section we review the language modeling framework for information retrieval and the document-centric impact model, both of which are closely related to our proposed model.

2.1 Language Modeling for IR

The language modeling framework for information retrieval was first proposed by Ponte and Croft [2]. Language models attempt to model *language* or *topicality*. Although there are many different variants of language modeling, we will only describe one of the most robust and commonly used formulations [8]. In this formulation, we are tasked with estimating document and query models. Models are defined as multinomial distributions over some fixed vocabulary $\mathcal{V}$. Due to their very nature, document and query models are often estimated differently. Documents are typically estimated using some smoothed maximum likelihood estimate [9]. Query models are either estimated according to their maximum likelihood estimate or using a more complex pseudo-relevance feedback-based

formulation, such as model-based feedback [10, 11] or relevance-based models [12, 13].

Given a query, documents are ranked according the negative KL divergence between the query and document model, which is computed as:

$$\begin{aligned} -KL(\theta_Q||\theta_D) &= H(\theta_Q) - CE(\theta_Q, \theta_D) \\ &\stackrel{rank}{=} \sum_{w \in \mathcal{V}} \theta_{w,Q} \log \theta_{w,D} \end{aligned} \tag{1}$$

where H is the entropy, CE is cross-entropy, θ_Q is the query model, and θ_D is the document model. Here, $\theta_{w,Q}$ and $\theta_{w,D}$ are shorthand for $P(w|\theta_Q)$ and $P(w|\theta_D)$, respectively. Although this sum is shown to go over the entire vocabulary $\mathcal{V}$, it is very often the case that terms that do not occur in the query are assigned a zero probability in the query model, thus significantly reducing the number of terms in the sum.

Pros Language modeling has several appealing characteristics. First, the model is formally motivated and based on a strong statistical foundation. This allows estimation and learning techniques from statistics and machine learning to be easily applied. Examples of such techniques are Bayesian smoothing [9], translation models [14], mixture models [15], cluster-based models [16, 17], and topic models [18]. Second, the model has proven to be highly effective over a wide range of retrieval tasks. Finally, the model is relatively easy to understand and implement.

Cons As with all models, language modeling also has several unappealing characteristics. One of the most fundamental theoretical issues with the framework concerns how query and document models are estimated and compared. Document models are estimated using techniques such as maximum likelihood estimation and Bayesian smoothing. These models, at their core, are modeling term *occurrences*. When sampling terms from a model estimated in this way, we expect our sample to exhibit term occurrence statistics similar in nature to those observed in the document they were estimated from. However, queries and documents exhibit very different term occurrence statistics, as was pointed out in the past [19]. For example, documents contain many function words, while queries rarely do. Therefore, it is theoretically unsettling to compare a query model with a document model, given that the statistical properties of term occurrences in queries and documents are fundamentally different.

Smoothing and high model complexity are also concerns. It is well known that smoothing does more than overcome the zero frequency problem. It also results in an implicit IDF factor in the query likelihood retrieval model, which ultimately results in a very *tf.idf*-like ranking function [9]. In terms of model complexity, a general multinomial model over a vocabulary of size $|\mathcal{V}|$ requires $|\mathcal{V}|-1$ parameters to be estimated. This is a very large number of free parameters to estimate for a model. Fortunately, most types of smoothing wash away these

many degrees of freedom (although they are still implicitly there) down to one or two parameters.

2.2 Document-Centric Impact Model

Anh and Moffat's document-centric impact retrieval model has been shown to be relatively effective and highly efficient [20, 7]. The model moves away from using quantitative parameterized term weighting functions. Instead, the model ranks terms according to their importance and imposes a very simple, pre-defined term weighting function to the sorted terms. As we discussed earlier, this type of model captures the notion of ordinal importance between terms, rather than trying to explicitly quantify importance, as is done in most other retrieval models.

Term weighting within the model is accomplished in two stages. First, terms are sorted according to some importance criteria. After sorting, the terms are then partitioned and assigned to bins. Each bin is assigned an integral impact ranging from 1 to k, where k is the total possible number of bins. The result of this process is that every term in every document is assigned a term weight in the set $\{1, \ldots, k\}$. Typical values of k include 4, 8 and 16 [7]. Each document is binned in the same way. We describe the details of Anh and Moffat's sorting and binning technique in Section 3.2.

Query terms are weighted differently, for several reasons. Anh and Moffat suggest that applying the strategy just described to queries will fail, due to the small number of query terms [7]. In addition, properly setting the query term weights is critical in order to achieve reasonable effectiveness. Our preliminary experiments showed that using uniform term weights results in poor effectiveness. The details of Anh and Moffat's query binning technique are given later.

Ranking within the model is done via a simple dot product between the document and query impacts (weights). This is computed according to:

$$RSV(D;Q) = \sum_{w \in \mathcal{V}} I_{w,Q} I_{w,D} \tag{2}$$

where $I_{w,Q}$ is the impact value assigned to query term w and $I_{w,D}$ is the impact value assigned to document term w. Terms not occurring in the query are often assigned an impact value of zero, although this is not required.

Pros Previous studies have shown that document-centric impact models are highly efficient, especially on large collections [21]. Impact-ordered indexes can reduce the amount of disk storage necessary compared to standard inverted list indexes. Furthermore, the model is amenable to efficient query processing [22]. This makes the model more attractive, from an efficiency standpoint, than language modeling and BM25.

Cons Despite the efficiency of the model, the effectiveness is often not as strong as language modeling or BM25 baselines. This trade-off between efficiency and

effectiveness can be controlled by choosing an appropriate number of bins. As expected, as fewer bins are used, efficiency increases, but effectiveness decreases.

Another issue with the model is the fact that there is no formal justification or motivation for the various binning strategies previously proposed in the literature. These strategies are typically heuristic and built from intuition.

Furthermore, using integral impact values is a matter of convenience and efficiency. However, there again is no formal motivation for choosing impacts in such a way. In Section 3.3 we describe a less heuristic estimation technique for choosing our model's equivalent of impact values.

3 Model

Our model is designed to combine the best aspects of language modeling and the document-centric impact model. The model is probabilistic like language modeling, thus allowing it to be incorporated into more complex statistical techniques, such as those described in Section 2.1. However, unlike language modeling, we do not model the generation of text. Instead, we model the importance of bins of terms (or arbitrary features).

The first step of our model, much like the document-centric impact model, requires us to bin the terms according to their importance. We assume that there is some fixed set of bins defined by $\mathcal{B}$, where each $B \in \mathcal{B}$ is an ordinal variable indicating relative importance. For example, B_1 may denote "most important", B_2 may denote "medium importance", and B_3 may denote "least important". Binning is performed on each document. There are many different ways to bin terms. We describe several approaches in the next section. The final result of this process is, for each document, a partitioning of the vocabulary into $|\mathcal{B}|$ bins.

After binning, we must estimate a model for each document. Rather than estimating text generation models, as is done in language modeling, we define importance models. These models attempt to capture the likelihood that the terms in a certain bin are important. We define a document importance model as a multinomial distribution over bins. As a matter of shorthand, we write $P(B|\theta_D)$ as $\theta_{B,D}$ and interpret $\theta_{B,D}$ as the probability that the terms in bin B (for document D) are important. This is fundamentally different than the language modeling interpretation. Indeed, we believe this interpretation is philosophically more appealing, as it does not assume that queries and documents are generated from the same underlying model. Instead, we model a fundamental, yet difficult to define, notion of term importance which is consistent across models.

Now that we have all of the pieces of our model, the final step is to describe how documents are ranked in response to a query. We rank documents using a generalized likelihood ranking function that allows query term weighting. We call this the *weighted likelihood* ranking function. It is defined as:

$$\begin{aligned} P(Q|D) &= \prod_{w \in \mathcal{V}} \theta_{b_D(w)}^{wt_{w,Q}} \\ &\overset{rank}{=} \sum_{w \in \mathcal{V}} wt_{w,Q} \log \theta_{b_D(w),D} \end{aligned} \tag{3}$$

where $wt_{w,Q}$ defines a weight for query term w and $b_D(w)$ is the bin that term w is assigned to in document D. A more formal definition of $b_D(w)$ is provided in the next section. This ranking function assigns high weights to documents that contain query terms that are both highly weighted and highly likely to be important. We note that this ranking function is reduced to the standard likelihood function when query term weights are set proportionally to the number of times they occur in the query. In the remainder of this section we describe various binning and weighting strategies for both queries and documents.

3.1 Query Binning and Weighting

IDF-Weighted Anh and Moffat propose a query binning strategy based on query term $idfs$. Their strategy has two steps. First, each query term is assigned a weight according to:

$$wt_w = (1 + \log tf_{w,Q}) \log \left(1 + \frac{maxtf_w}{cf_w}\right) \tag{4}$$

where $tf_{w,Q}$ is the number of times w occurs in the query, $maxtf_w$ is the maximum number of times w occurs in any document in the collection and cf_w is the total number of times w occurs in the collection.

The final step bins terms linearly according to their weight. This results in query terms with very large idf values being assigned to the "important" bins and those with low idf being assigned to bins of lower importance. Query term weights are then assigned according to:

$$wt_w = I_{b_Q(w),Q} \tag{5}$$

where it is assumed that some *a priori* set of impacts have been assigned to each query bin and $I_{b_Q(w),Q}$ is the impact assigned to query bin $b_Q(w)$. In this work, we follow Anh and Moffat and assume integral impacts. That is, query terms in the least important bin are assigned an impact value of 1, those in the next least important bin are assigned impact value 2, and so on.

Other Methods Other methods for computing query weights are possible, although not explored in detail here. For example, relevance-based language models estimate query weights by mixing together the language models of a set of relevant or pseudo-relevant documents [13]. An analogous technique could be used within our model to estimate better query weights.

One criticism of relevance-based language models is that they assign large probabilities to function words due to their prevalence in the top ranked documents. Such models do not try to separate out the meaningful terms from the background, as is done with parsimonious language models [23]. Indeed, we suspect that relevance-based query models estimated using our model will behave as the parsimonious language models do. This is due to the fact that function terms will be given very low probability, as they are assigned to "unimportant" bins, and give topical terms larger probability, as they are assigned to "important" bins. Investigating this phenomenon further is part of future work.

3.2 Document Binning

For each document, we define a binning function $b_D : \mathcal{V} \rightarrow \mathcal{B}$ that maps the original vocabulary ($\mathcal{V}$) onto a set of bins $\mathcal{B}$. For a document D, the binned document representation is generated by applying b_D to each term. This results in a new document that only consists of bins from $\mathcal{B}$.

The bin vocabulary $\mathcal{B}$ can be thought of as a surrogate vocabulary that captures some latent aspect of the original vocabulary. The purpose of binning is to cluster or combine terms that are similar under some criteria. In this work we aim at binning terms according to their importance. However, it is possible that other binning criteria may be more appropriate for other applications.

Another important consequence of binning is the significant reduction of the effective vocabulary size when we choose $|\mathcal{B}| \ll |\mathcal{V}|$. The binning process reduces the dimensionality of our document representation. This results in significantly reduced model complexity which can minimize the effects of overfitting and significantly improve query processing efficiency.

(TF,IDF) Binning Anh and Moffat propose a number of document-centric binning strategies [7]. Each of their proposed strategies have a *sorting* and *assignment* stage. In the sorting stage, document terms are sorted according to some criteria. In the assignment stage, the sorted terms are assigned to bins.

Anh and Moffat report that the *(TF, IDF)* sorting method results in the best effectiveness [7]. This method sorts terms in descending order using term frequency as the primary key and inverse document frequency as a tie breaking secondary key.

The sorted terms are assigned to bins according to a geometric sequence. That is, a small number of terms (i.e. those at the beginning of the sorted list) are assigned to the "most important" bin, a larger number of terms are assigned to the next most important bin, with the least important bin containing the largest number of terms (i.e. those terms at the end of the sorted list). More formally, the number of terms in bin b_i is given by:

$$x_i = (|D| + 1)^{1/k} x_{i+1} \tag{6}$$

$$x_{|\mathcal{B}|} = (|D| + 1)^{1/k} - 1 \tag{7}$$

where the least important terms are assigned to bin b_1 and the most important to bin $b_{|\mathcal{B}|}$.

Other Binning Strategies Although not explored in this work, we note that there are a number of reasonable strategies for binning terms. In particular, index pruning strategies [24, 25] and probabilistic indexing techniques [26] may be useful. These methods share the same goal as binning by term importance. However, instead of explicitly creating a binning, these methods only choose to index those terms that are likely to be important within a given document.

3.3 Document Model Estimation

Document-Centric Impact Estimate If some pre-defined impact (integral or real-valued) value is assigned to each term bin, as in the document-centric impact model [7], then we can convert the impacts to probabilities as follows:

$$\hat{\theta}_{B,D} = \frac{\exp\left[I_{B,D}\right]}{\sum_{B' \in \mathcal{B}} \exp\left[I_{B',D}\right]} \tag{8}$$

Unfortunately, it is unclear how to optimally set the impact weights given some binning. While the integral assignment proposed by Anh and Moffat is simple, it is likely not to be optimal. Therefore, a more informed, well-founded method for estimating the document model probabilities is required.

It is straightforward to show that when document models are estimated in this way and query term weights are computed using the IDF-Weighted method described in Section 3.1 our ranking function (Equation 3) is equivalent to the impact ranking function (Equation 2). This provides a probabilistic interpretation of the document-centric impact model.

Discriminative Estimation As described previously, one paradox of the language modeling approach to information retrieval is that document models are estimated so as to maximize the likelihood (or the *a posteriori*) of generating the terms in the document, while the overarching goal is to maximize some evaluation metric, such as mean average precision. Therefore, we propose to choose document model probabilities in such a way that they maximize some retrieval metric, instead of properly modeling term occurrence statistics. We acknowledge that it is common practice in language modeling to train a model by tuning the smoothing parameter in order to maximize some metric. However, this is typically a single, coarse grained parameter that has very specific interactions with the model. We are proposing to tune the model in a radically different way that allows finer control and results in parameter settings that can be interpreted more intuitively than Dirichlet or Jelinek-Mercer smoothing parameters.

Given a set of bins $\mathcal{B}$, our goal is to estimate $\theta_{B,D}$ by maximizing some retrieval metric. This optimization problem involves setting $|\mathcal{B}| - 1$ parameters for each document in the collection. Even when a small number of bins is chosen, this problem is infeasible. However, if we make the simplifying assumption that $\theta_{B,D} = \theta_{B,D'}$ for all D and D' in the collection, then the problem becomes more reasonable. This assumption ties all of the bin importance probabilities together. That is, it assumes that the likelihood a term in some bin j is important is the same across all documents. While this assumption may be overly simplistic, it significantly reduces the number of free parameters in our optimization problem to $|\mathcal{B}| - 1$, which is easily solved for most reasonable bin settings. Another side effect of our assumption is that it allows for very efficient query processing.

Formally, our discriminative estimation technique requires the following optimization problem to be solved:

$$\left[\hat{\theta}_1 \ldots \hat{\theta}_{|\mathcal{B}|}\right] = \arg\max E(\mathcal{R}; \theta_1 \ldots \theta_{|\mathcal{B}|}) \tag{9}$$

where $\mathcal{R}$ is the set of relevance judgments and E is some evaluation metric.

Since most information retrieval metrics are not amenable to standard optimization techniques, we choose to solve this optimization problem using greedy hill climbing, which is a local search technique. This hill climbing approach is reasonable for small numbers of bin, even on very large collections, because of the low cost of evaluating large numbers of queries.

4 Evaluation

In this section we evaluate our proposed binning and estimations techniques in terms of effectiveness. Although efficiency is important, the evaluation of the efficiency of integral vs. non-integral impacts is beyond the scope of this work.

Collection	# Docs	Train Topics	Test Topics
TREC Disks 1,2	741,856	51-150	151-200
TREC Disks 4,5	528,155	301-450	601-700
WT10g	1,692,096	451-500	501-550

Table 1. Overview of collections and topics used.

All binning-related experiments are carried out using Galago[3], a new indexing and retrieval system developed to test our new probabilistic model. In addition, the Indri search system is used for the query likelihood runs [27]. We evaluate our methods on three TREC data sets with varying characteristics. Table 1 provides an overview of each data set. The TREC Disks 1 and 2 (TREC12) and TREC Disks 4 and 5 (TREC45) data sets consist of newswire articles from several sources. The WT10G data set is much larger and is made up of web documents. The queries associated with each data set are split into a training and test set. The training set is used to tune parameters (smoothing parameters and document importance model probabilities). The test set is used solely for evaluation purposes.

Documents are stemmed using the Porter stemmer and stopped using the same list of stopwords used by Anh and Moffat [7]. Queries are constructed using only the title portion of the TREC topic. Finally, we use 8 bins when IDF-weighted query term binning is employed.

4.1 Integral vs. Discriminative Weights

We now scrutinize the optimality of choosing document model probabilities based on integral impacts. Therefore, we wish to compare the results of (TF,IDF) binning and integral document estimates with (TF,IDF) binning and discriminative document model estimates. Recall that the integral weights are set in a completely unsupervised fashion, whereas the discriminative weights are learned from training data.

[3] `http://www.galagosearch.org`

Data	θ_D Estimation	$wt_{w,Q}$ Estimation	2 bins	4 bins	8 bins	16 bins
TREC12	Integral	IDF	0.2067	0.2241	0.2273	0.2273
	Discriminative	IDF	0.2105	0.2269	0.2315	$0.2336^{\dagger}$
	Language Modeling		0.2633			
TREC45	Integral	IDF	0.2325	0.2417	0.2427	0.2459
	Discriminative	IDF	$0.2430^{\dagger}$	$0.2494^{\dagger}$	$0.2577^{\dagger}$	$0.2567^{\dagger}$
	Language Modeling		0.2920			
WT10g	Integral	IDF	0.1522	0.1598	0.1863	0.1886
	Discriminative	IDF	0.1570	$0.1692^{\dagger}$	$0.1879^{\dagger}$	0.1887
	Language Modeling		0.1861			

Table 2. Mean average precision for various combinations of document model estimation techniques, query weight estimation strategies, and number of document bins. A query likelihood language modeling run using Dirichlet smoothing is also included for comparison. A † superscript indicates statistically significant improvements in effectiveness over the cell immediately above it using a one-tailed t-test with $p < 0.05$.

As we see from Table 2, the discriminatively trained weights are consistently better than the integral weights across various document bin sizes. These improvements are statistically significant for over half of our test cases.

While this result is not necessarily surprising, it does allow us to quantitatively evaluate the optimality of the naïvely chosen integral weights. Indeed, the results of our experiments show that integral weights, while not being optimal, achieve results that are often close to optimal. This is a more interesting and surprising result, as it was expected that such weights would be far from optimal. The reason why such weights may be so close to optimal may be the result of the particular binning strategy used, and therefore our analysis does not extend beyond (TF,IDF) binning. It is unclear whether these results will hold for more complex binning strategies. It is likely that in more complex cases the divide between the discriminatively trained model and the integral weight model will increase.

4.2 Language Modeling vs. Impact-Based Models

We now briefly investigate how well the impact-based models perform when compared to a strong language modeling baseline. The language modeling baseline significantly outperforms the best impact-based formulation for the TREC12 and TREC45 data sets. Interestingly, the two models demonstrate comparable effectiveness on the WT10G collection.

Our results seem to contradict those described by Anh and Moffat [7], which showed that the impact-based model significantly outperformed language modeling and BM25. However, most of the language modeling results outlined in their work were quoted from previous work that had very weak language modeling baseline numbers. Indeed, our rigorously tuned language modeling approach shows significantly stronger performance on the newswire data sets compared to the impact-based model.

Our experience with impact-based models suggest that they strongly prefer documents that contain all of the query terms. We believe that this is an asset to the model in large collections where there are likely to be many documents that contain all the query terms [28]. Furthermore, some recent work suggests that relevance judgments in large TREC collections are biased toward those documents that contain all of the query terms. We believe that this may explain why impact methods perform strongly compared to language modeling on larger collections while there is a large effectiveness gulf on smaller collections, which presumably have a higher percentage of relevant documents that do not contain all of the query terms.

5 Conclusions

In this paper, we presented a probabilistic retrieval model that can be considered a statistical view of the document-based impact model. Our model achieves good effectiveness and efficiency by combining the strengths of the language modeling and document-centric impact models.

In addition, we described a supervised method for discriminatively learning document importance model weights. Rather than using integral weights, as was done in previous work, we find the set of weights that maximize some underlying retrieval metric. Our results showed consistent and significant improvements in effectiveness when weights were learned in this way.

Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval, in part by NSF grant #CNS-0454018, and in part by Advanced Research and Development Activity and NSF grant #CCF-0205575. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

References

1. Robertson, S.E., Walker, S.: Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In: Proc. 17th SIGIR, Springer-Verlag New York, Inc. (1994) 232–241
2. Ponte, J.M., Croft, W.B.: A language modeling approach to information retrieval. In: Proc. 21st SIGIR. (1998) 275–281
3. Fang, H., Zhai, C.: An exploration of axiomatic approaches to information retrieval. In: Proc. 28th SIGIR. (2005) 480–487
4. Amati, G., Rijsbergen, C.J.V.: Probabilistic models of information retrieval based on measuring the divergence from randomness. ACM Transactions on Information Syststems **20**(4) (2002) 357–389
5. Nallapati, R.: Discriminative models for information retrieval. In: Proc. 27th SIGIR. (2004) 64–71

6. Gao, J., Qi, H., Xia, X., Nie, J.Y.: Linear discriminant model for information retrieval. In: Proc. 28th SIGIR. (2005) 290–297
7. Anh, V.N., Moffat, A.: Simplified similarity scoring using term ranks. In: Proc. 28th SIGIR. (2005) 226–233
8. Song, F., Croft, W.B.: A general language model for information retrieval. In: Proc. 8th CIKM. (1999) 316–321
9. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to ad hoc information retrieval. In: Proc. 24th SIGIR. (2001) 334–342
10. Tao, T., Zhai, C.: Regularized estimation of mixture models for robust pseudo-relevance feedback. In: Proc. 29th SIGIR. (2006) 162–169
11. Zhai, C., Lafferty, J.: Model-based feedback in the language modeling approach to information retrieval. In: Proc. 10th CIKM. (2001) 403–410
12. Diaz, F., Metzler, D.: Improving the estimation of relevance models using large external corpora. In: Proc. 29th SIGIR. (2006) 154–161
13. Lavrenko, V., Croft, W.B.: Relevance based language models. In: Proc. 24th SIGIR. (2001) 120–127
14. Berger, A., Lafferty, J.: Information retrieval as statistical translation. In: Proc. 22nd SIGIR. (1999) 222–229
15. Ogilvie, P., Callan, J.: Combining document representations for known-item search. In: Proc. 26th SIGIR. (2003) 143–150
16. Liu, X., Croft, W.B.: Cluster-based retrieval using language models. In: Proc. 27th SIGIR. (2004) 186–193
17. Kurland, O., Lee, L.: Corpus structure, language models, and ad hoc information retrieval. In: Proc. 27th SIGIR. (2004) 194–201
18. Wei, X., Croft, W.B.: Lda-based document models for ad-hoc retrieval. In: Proc. 29th SIGIR. (2006) 178–185
19. Jones, K.S.: Language modelling's generative model: Is it rational? Technical report, University of Cambridge (2004)
20. Anh, V.N., Moffat, A.: Collection-independent document-centric impacts. In: Proc. Australian Document Computing Symposium. (2004) 25–32
21. Anh, V.N., Moffat, A.: Melbourne university 2004: Terabyte and web tracks. In: Proceedings of TREC 2004. (2004)
22. Anh, V.N., Moffat, A.: Pruned query evaluation using pre-computed impacts. In: Proc. 29th SIGIR. (2006) 372–379
23. Hiemstra, D., Robertson, S., Zaragoza, H.: Parsimonious language models for information retrieval. In: Proc. 27th SIGIR. (2004) 178–185
24. Büttcher, S., Clarke, C.L.A.: A document-centric approach to static index pruning in text retrieval systems. In: Proc. 15th CIKM. (2006) 182–189
25. Carmel, D., Cohen, D., Fagin, R., Farchi, E., Herscovici, M., Maarek, Y.S., Soffer, A.: Static index pruning for information retrieval systems. In: Proc. 24th SIGIR. (2001) 43–50
26. Fuhr, N.: Two models of retrieval with probabilistic indexing. In: Proc. 9th SIGIR. (1986) 249–257
27. Strohman, T., Metzler, D., Turtle, H., Croft, W.B.: Indri: A language model-based serach engine for complex queries. In: Proceedings of the International Conference on Intelligence Analysis. (2004)
28. Buckley, C., Dimmick, D., Soboroff, I., Voorhees, E.: Bias and the limits of pooling. In: Proc. 29th SIGIR. (2006) 619–620