Query Likelihood with Negative Query Generation

Yuanhua Lv
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801
ylv2@uiuc.edu

ChengXiang Zhai
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801
czhai@cs.uiuc.edu

ABSTRACT

The query likelihood retrieval function has proven to be empirically effective for many retrieval tasks. From theoretical perspective, however, the justification of the standard query likelihood retrieval function requires an unrealistic assumption that ignores the generation of a "negative query" from a document. This suggests that it is a potentially non-optimal retrieval function.

In this paper, we attempt to improve the query likelihood function by bringing back the negative query generation. We propose an effective approach to estimate the probabilities of negative query generation based on the principle of maximum entropy, and derive a more complete query likelihood retrieval function that also contains the negative query generation component. The proposed approach not only bridges the theoretical gap in the existing query likelihood retrieval function, but also improves retrieval effectiveness significantly with no additional computational cost.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval models

General Terms

Algorithms

Keywords

Negative query generation, query likelihood, language model, probability ranking principle, principle of maximum entropy

1. INTRODUCTION

The query likelihood retrieval method [12] has recently enjoyed much success for many different retrieval tasks [18]. The query likelihood retrieval method [12] assumes that a query is a sample drawn from a language model: given a query Q and a document D, we compute the likelihood of "generating" query Q with a model estimated based on document D. We can then rank documents based on the likelihood of generating the query.

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.

Although query likelihood has performed well empirically, there was criticism about its theoretical foundation [13, 4]. In particular, Sparck Jones questioned "where is relevance?" [4]. Responding to this criticism, Lafferty and Zhai [6] showed that under some assumptions the query likelihood retrieval method can be justified based on probability ranking principle [14] which is regarded as the foundation of probabilistic retrieval models.

However, from theoretical perspective, the justification of using query likelihood as a retrieval function based on the probability ranking principle [6] requires an unrealistic assumption about the generation of a "negative query" from a document, which states that the probability that a user who dislikes a document would use a query does not depend on the particular document. This assumption enables ignoring the negative query generation in justifying using the standard query likelihood method as a retrieval function. In reality, however, this assumption does not hold because a user who dislikes a document would more likely avoid using words in the document when posing a query. This suggests that the standard query likelihood function is a potentially non-optimal retrieval function.

In order to address this theoretical gap between the standard query likelihood and the probability ranking principle, in this paper, we attempt to bring back the component of negative query generation.

A main challenge in estimating the negative query generation probability is to develop a general method for any retrieval case. Our solution to this problem is to estimate the probability of negative query generation purely based on document D so as to make it possible to incorporate the negative query generation for retrieving any document in response to any query. Specifically, we exploit document D to infer the queries that a user would use to avoid retrieving D based on the intuition that such queries would not likely have any information overlap with D. We then propose an effective approach to estimate probabilities of negative query generation based on the principle of maximum entropy [3], which leads to a negative query generation component that can be computed efficiently. Finally, we derive a more complete query likelihood retrieval function that also contains the negative query generation component, which essentially scores a document with respect to a query according to the ratio of the probability that a user who likes the document would pose the query to the probability that a user who dislikes the document would pose the query.

The proposed query likelihood with negative query generation retrieval function not only bridges the theoretical gap

in the existing query likelihood function, but also improves retrieval effectiveness significantly in our experiments.

2. QUERY LIKELIHOOD METHOD

In the query likelihood retrieval method [12], given a query Q and a document D, we compute the likelihood of "generating" query Q with a model θ_D estimated based on document D, and then rank the document based on its query likelihood:

$$Score(D, Q) = p(Q|\theta_D) \tag{1}$$

The query generation can be based on any language model [12, 11, 2, 19, 10, 9, 16] . So far, using a multinomial distribution [11, 2, 19] for θ_D has been most popular and most successful, which is also adopted in our paper. With the multinomial distribution, the query likelihood is $p(Q|\theta_D) = \prod_w p(w|\theta_D)^{c(w,Q)}$, where $c(w, Q)$ is the count of term w in query Q. According to the maximum likelihood estimator, we have the following estimation of the document language model θ_D for the multinomial model:

$$p_{ml}(w|\theta_D) = \frac{c(w, D)}{|D|} \tag{2}$$

where $c(w, D)$ indicates the frequency of w in document D, and $|D|$ is the document length. θ_D needs smoothing to overcome the zero-probability problem, and an effective method is the Dirichlet prior smoothing [19]:

$$p(w|\theta_D) = \frac{|D|}{|D| + \mu} p_{ml}(w|D) + \frac{\mu}{|D| + \mu} p(w|C) \tag{3}$$

Here μ is the smoothing parameter (Dirichlet prior), and $p(w|C)$ is the collection language model which is estimated as $p(w|C) = \frac{c(w,C)}{\sum_{w'} c(w',C)}$, where $c(w, C)$ indicates the count of term w in the whole collection C.

The query likelihood scoring function essentially ranks documents using the following formula [19]:

$$\begin{aligned} &\log p(Q|\theta_D) \\ &\overset{rank}{=} \sum_{w \in Q \cap D} c(w, Q) \log \left(1 + \frac{c(w, D)}{\mu p(w|C)}\right) + |Q| \log \frac{\mu}{|D| + \mu} \end{aligned} \tag{4}$$

where $|Q|$ represents query length.

3. NEGATIVE QUERY GENERATION

To understand the retrieval foundation of the query likelihood method, Lafferty and Zhai [6] provided a relevance-based derivation of the query likelihood method. Formally, let random variables D and Q denote a document and query, respectively. Let R be a binary random variable that indicates whether D is relevant to Q or not. Following [5], we will denote by ℓ ("like") and $\bar{\ell}$ ("not like") the value of the relevance variable. The probability ranking principle [14] provides a justification for ranking documents for a query based on the conditional probability of relevance, i.e., $p(R = \ell|D, Q)$. This is equivalent to ranking documents based on the odds ratio, which can be further transformed using Bayes' Rule:

$$O(R = \ell|Q, D) = \frac{p(R = \ell|Q, D)}{p(R = \bar{\ell}|Q, D)} \propto \frac{p(Q, D|R = \ell)}{p(Q, D|R = \bar{\ell})} \tag{5}$$

There are two different ways to decompose the joint probability $p(Q, D|R)$, corresponding to "document generation" and "query generation" respectively. With document generation $p(Q, D|R) = p(D|Q, R)p(Q|R)$, we have

$$O(R = \ell|Q, D) \propto \frac{p(D|Q, R = \ell)}{p(D|Q, R = \bar{\ell})} \tag{6}$$

Most classical probabilistic retrieval models [15, 5, 1] are based on document generation.

Query generation, $p(Q, D|R) = p(Q|D, R)p(D|R)$, is the focus of this paper. With query generation, we end up with the following ranking formula:

$$O(R = \ell|Q, D) \propto \frac{p(Q|D, R = \ell)p(R = \ell|D)}{p(Q|D, R = \bar{\ell})p(R = \bar{\ell}|D)} \tag{7}$$

in which, the term $p(R|D)$ can be interpreted as a prior of relevance on a document. Without any prior knowledge, we may assume that this term is the same across all the documents, and obtain the following simplified formula:

$$O(R = \ell|Q, D) \propto \frac{p(Q|D, R = \ell)}{p(Q|D, R = \bar{\ell})} \tag{8}$$

There are two components in this model. $p(Q|D, R = \ell)$ can be interpreted as a positive query generation model. It is essentially the basic query likelihood, which suggests that the query generation probability used in all the query likelihood scoring methods intuitively means the probability that a user who likes document D would pose query Q. The other component $p(Q|D, R = \bar{\ell})$ can be interpreted as the generation probability of a "negative query" from a document, i.e., the probability that a user who dislikes a document D would use a query Q.

However, in order to justify using the basic query likelihood alone as the ranking formula, an unrealistic assumption has to be made about this negative query generation component, which states that the probability that a user who dislikes a document would use a query does not depend on the particular document [6], formally $p(Q|D, R = \bar{\ell}) = p(Q|R = \bar{\ell})$.

This assumption enables ignoring the negative query generation in the derivation of the basic query likelihood retrieval function, leading to the following basic query likelihood method: $O(R = \ell|Q, D) \propto p(Q|D, R = \ell) = p(Q|\theta_D)$.

In reality, however, this assumption does *not* hold because a user who dislikes a document would more likely avoid using words in the document when posing a query, suggesting a theoretical gap between the standard query likelihood and the probability ranking principle. This shows that the standard query likelihood function is a potentially non-optimal retrieval function.

In the following section, we attempt to improve the query likelihood function by estimating, rather than ignoring the component of negative query generation $p(Q|D, R = \bar{\ell})$.

4. QUERY LIKELIHOOD WITH NEGATIVE QUERY GENERATION

What would a user like if he/she does not like D? We assume that there exists a "complement document" $\bar{D}$, and that if a user does not like D, the user would like $\bar{D}$. That is, when generating query Q, if a user does not like D, the user would randomly pick a word from $\bar{D}$. Formally,

$$p(w|D, R = \bar{\ell}) = p(w|\theta_{\bar{D}}) \tag{9}$$

The challenge now lies in how to estimate a language model $\theta_{\bar{D}}$, which we refer to as the "negative document language model" of D. Note that the negative document language model in our paper is still a "document" model, which is completely different from the relevance model $p(w|R=\ell)$ [7] and the irrelevance model $p(w|R=\bar{\ell})$ [17] that capture the probability of observing a word w relevant and non-relevant to a particular information need respectively.

Ideally we should use many actual queries by users who do not want to retrieve document D to estimate the probability $p(w|\theta_{\bar{D}})$. For example, we may assume that if a user sees a document in search results but does not click on it, he/she dislikes the document. Under this assumption, we can use all the queries from the users who "dislike" the document to approximate $\bar{D}$. However, in practice, only very few search results will be shown to users and certainly there are always queries that we would not even have seen. Yet, as a general retrieval model, the proposed method must have some way to estimate $\theta_{\bar{D}}$ for any document with respect to any query.

One straightforward way is using the background language model $p(w|C)$ to approximate $p(w|\theta_{\bar{D}})$, by assuming that almost all other documents in the collection are complementary to D: $p(w|\theta_{\bar{D}}) \approx p(w|C)$. With this estimate, the negative query generation component will not affect the ranking of documents, because the probability of negative query generation will be constant for all documents: it justifies the document independent negative query generation component in the standard query likelihood method. However, the content of document D is ignored in this estimate.

We are interested in estimating $p(w|\theta_{\bar{D}})$ in a general way based on the content of document D so as to make it possible to incorporate a document dependent negative query generation component when retrieving any document. Our idea is based on the intuition that if a user wants to avoid retrieving document D, he/she would more likely avoid using words in the document when posing a query. That is, the user would like a document $\bar{D}$ with little information overlap with D. Therefore, $\bar{D}$ should contain a set of words that do not exist in D, because given only document D available, the sole constraint of $\bar{D}$ is that, if a word w occurs in D, i.e., $c(w,D)>0$, this word should not occur in $\bar{D}$.

$$c(w,\bar{D}) = \begin{cases} 0 & \text{if } c(w,D)>0 \\ ? & \text{otherwise} \end{cases} \tag{10}$$

where "?" means unknown.

How to determine the count of a word in $\bar{D}$ if it does not occur in D? As the probability distribution of such a word is unknown, according to the *principle of maximum entropy* [3], each such a word occurring in $\bar{D}$ should have the same frequency δ, which maximizes the information entropy under the only prior data D. That is, $\bar{D}$ contains a set of words that are complementary to D in the universe word space (i.e., the whole word vocabulary V). Formally,

$$c(w,\bar{D}) = \begin{cases} 0 & \text{if } c(w,D)>0 \\ \delta & \text{otherwise} \end{cases} \tag{11}$$

According to the maximum likelihood estimator, we have the following estimation of the negative document language model $\theta_{\bar{D}}$ for the multinomial model:

$$p_{ml}(w|\theta_{\bar{D}}) = \frac{c(w,\bar{D})}{|\bar{D}|} \tag{12}$$

where $|\bar{D}|$ is the "length" of $\bar{D}$, which can be computed by aggregating frequencies of all words occurring in $\bar{D}$. Because the number of unique words in D is usually much smaller than the number of unique words in the whole document collection C (i.e., $|V|$), the number of unique words in $\bar{D}$ is approximately the same as $|V|$ based on Formula 11. Thus

$$|\bar{D}| = \sum_{w\in V} c(w,\bar{D}) \approx \delta|V| \tag{13}$$

Due to the existence of zero probabilities, $p_{ml}(w|\theta_{\bar{D}})$ needs smoothing. Following the estimation of regular document language models, we also choose the Dirichlet prior smoothing method due to its effectiveness [19]. Formally,

$$p(w|\theta_{\bar{D}}) = \frac{\delta|V|}{\delta|V|+\mu} p_{ml}(w|\theta_{\bar{D}}) + \frac{\mu}{\delta|V|+\mu} p(w|C) \tag{14}$$

where μ is the Dirichlet prior. Since the influence of μ can be absorbed into variable δ obviously, we thus set it simply to the same Dirichlet prior value as used for smoothing the regular document language model (Equation 3).

Now we can bring back the negative query generation component to the query generation process:

$$\begin{aligned} O(R=\ell|Q,D) &\overset{rank}{=} \log \frac{p(Q|D,R=\ell)}{p(Q|D,R=\bar{\ell})} \\ &= \log p(Q|D,R=\ell) - \log p(Q|D,R=\bar{\ell}) \\ &= \log p(Q|\theta_D) - \log p(Q|\theta_{\bar{D}}) \end{aligned} \tag{15}$$

The negative query loglikelihood $\log p(Q|\theta_{\bar{D}})$ can be further written as

$$\log p(Q|\theta_{\bar{D}}) \overset{rank}{=} -\sum_{w\in Q\cap D} c(w,Q) \log\left(1+\frac{\delta}{\mu p(w|C)}\right) \tag{16}$$

The corresponding derivation process has been shown in Formula 17.

Plugging Equations 4 and 16 into Equation 15, we finally obtain a more complete query likelihood retrieval function that also contains the negative query generation component:

$$\begin{aligned} &O(R=\ell|Q,D) \\ &\overset{rank}{=} \sum_{w\in Q\cap D} c(w,Q)\left[\log\left(1+\frac{c(w,D)}{\mu p(w|C)}\right) + \log\left(1+\frac{\delta}{\mu p(w|C)}\right)\right] \\ &+ |Q|\log\frac{\mu}{|D|+\mu} \end{aligned} \tag{18}$$

Comparing Formula 18 with the standard query likelihood in Formula 4, we can see that our new retrieval function essentially introduces a novel component $\log\left(1+\frac{\delta}{\mu p(w|C)}\right)$ to reward the matching of a query term, and it rewards more the matching of a more discriminative query term, which not only intuitively makes sense, but also provides a natural way to incorporate IDF weighting to query likelihood, which has so far only been possible through a second-stage smoothing step [19]. Note that when we set $\delta=0$, the proposed retrieval function degenerates to the standard query likelihood function.

Note that this new component we introduced is a *term-dependent constant*, which means that the proposed new retrieval function only incurs $\mathcal{O}(|Q|)$ additional computation cost as compared to the standard query likelihood function, which can be certainly ignored.

$$\log p(Q|\theta_{\bar{D}}) = \sum_{w \in Q} c(w,Q) \log p(w|\theta_{\bar{D}})$$

$$= \sum_{w \in Q \cap D} c(w,Q) \log \left(\frac{\mu}{\delta|V| + \mu} p(w|C) \right) + \sum_{w \in Q, w \notin D} c(w,Q) \log \left(\frac{\delta}{\delta|V| + \mu} + \frac{\mu}{\delta|V| + \mu} p(w|C) \right)$$

$$= \sum_{w \in Q \cap D} c(w,Q) \log \left(\frac{\mu p(w|C)}{\delta|V| + \mu} \right) + \underbrace{\sum_{w \in Q} c(w,Q) \log \left(\frac{\delta + \mu p(w|C)}{\delta|V| + \mu} \right)}_{\text{document independent constant}} - \sum_{w \in Q \cap D} c(w,Q) \log \left(\frac{\delta + \mu p(w|C)}{\delta|V| + \mu} \right) \quad (17)$$

$$\stackrel{rank}{=} - \sum_{w \in Q \cap D} c(w,Q) \log \left(1 + \frac{\delta}{\mu p(w|C)} \right)$$

Query	Method	WT2G			WT10G			Terabyte			Robust04		
		MAP	P@10	#Rel	MAP	P@10	#Rel	MAP	P@10	#Rel	MAP	P@10	#Rel
Short	QL	0.3088	0.4600	1905	0.1930	0.2796	3812	0.2921	0.5463	19391	0.2521	0.4225	10260
	XQL	**0.3187**^{3}	0.4620	1920	**0.1961**	0.2807	3852	**0.2936**^{3}	0.5503	19404	**0.2530**^{1}	0.4229	10244
Verbose	QL	0.2742	0.4000	1837	0.1790	0.3150	3816	0.2112	0.4718	14468	0.2329	0.3968	9344
	XQL	**0.2871**^{2}	0.4100	1854	**0.1871**^{3}	0.3140	3975	**0.2143**^{1}	0.4718	14734	**0.2440**^{4}	0.3992	9372

Table 1: Comparison of the standard query likelihood (QL) and the proposed query likelihood with negative query generation (XQL) using cross validation. Superscripts 1/2/3/4 indicate that the corresponding improvement is significant at the 0.05/0.02/0.01/0.001 level using the Wilcoxon non-directional test.

More interestingly, the developed query likelihood with negative query generation (Formula 18) leads to the same ranking formula as derived by lower-bounding term frequency normalization in the query likelihood method [8]. However, the formula derived in [8] is based on a heuristic approach, which is inconsistent with the theoretical framework of the query likelihood method. Our method provides a probabilistic approach for appropriately lower-bounding term frequency normalization in the query likelihood method.

5. EXPERIMENTS

We use four TREC collections: WT2G, WT10G, Terabyte, and Robust04, which represent different sizes and genre of text collections. We adopt the same preprocessing and parameter tuning methods as in our recent study [8]. Our goal is to see if the proposed negative query generation component can work well for improving the standard query likelihood method.

We first compare the effectiveness of the standard query likelihood (labeled as **QL**) and the proposed query likelihood with negative query generation (labeled as **XQL**). QL has one free parameter μ, and XQL has two free parameters μ and δ. We use cross validation to train both μ and δ for XQL and μ for QL.

We report the comparison results in Table 1. The results demonstrate that XQL outperforms QL consistently in terms of MAP and also achieves better P@10 and recall (#Rel) scores than QL in most cases. The MAP improvements of XQL over QL are significant in general. These results show that bringing back the negative query generation component is able to improve retrieval performance, and that the proposed approach works effectively.

Regarding different query types, we observe that XQL usually improves more on verbose queries than on short queries. For example, the MAP improvements on WT2G, WT10G, and Robust04 collections are as high as 5% for verbose queries.

We introduce a parameter δ to control the negative query generation component. We plot MAP improvements of XQL over QL against different δ values in Figure 1. It demonstrates that, for verbose queries, when δ is set to a value around 0.05, XQL works very well across different collections. Therefore, δ can be safely "eliminated" from XQL for verbose queries by setting it to a default value 0.05. Although δ tends to be collection-dependent for short queries, setting it conservatively to a small value, e.g., 0.02, can often lead to consistent improvement on all collections.

As XQL and QL share one parameter μ, the Dirichlet prior, we are also interested in understanding how this parameter affects the retrieval performance of XQL and QL. So we draw the sensitivity curves of QL and XQL to μ in Figure 2. It shows that XQL is consistently more effective than QL when we vary the value of μ. Moreover, the curve trends of QL and XQL are very similar to each other. In particular, QL and XQL even often share the same optimal setting of μ. These are interesting observations, which suggest that μ and δ do not interact with each other seriously; as a result, we can tune two parameters separately.

6. CONCLUSIONS

In this paper, we show that we can improve the standard query likelihood function by bringing back the component of negative query generation (i.e., the probability that a user who dislikes a document would use a query). Our work not only bridges the theoretical gap between the standard query likelihood method and the probability ranking principle, but also improves retrieval effectiveness over the standard query likelihood with no additional computational cost for various types of queries across different collections. The proposed retrieval function can potentially replace the standard query likelihood retrieval method in all retrieval applications.

7. ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under grant CNS-1027965, by U.S. Air Force Office of Scientific Research MURI award FA9550-08-1-0265, and by a Sloan Research Fellowship.

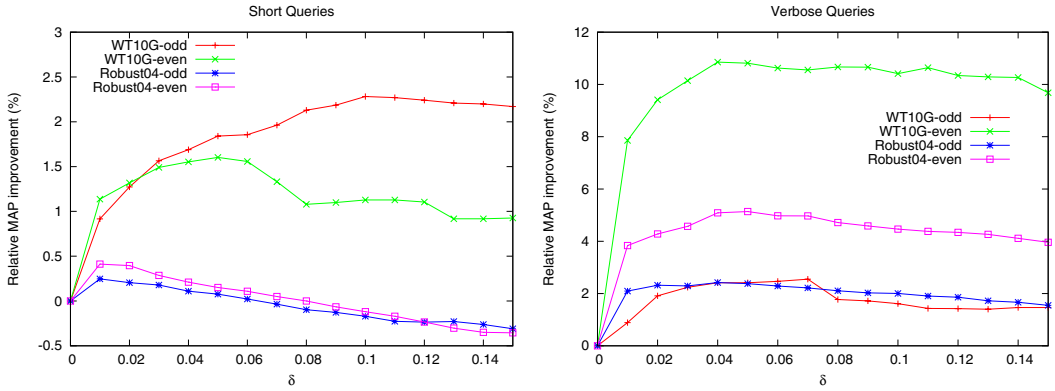

Figure 1: Performance Sensitivity to δ of the proposed query likelihood with negative query generation (XQL) for short (left) and verbose (right) queries. Note that XQL will degenerate to QL when $\delta = 0$. The corresponding settings of parameter μ for each δ value are well tuned. "odd" and "even" in the legend mean that the corresponding curves are based on odd and even-numbered topics respectively.

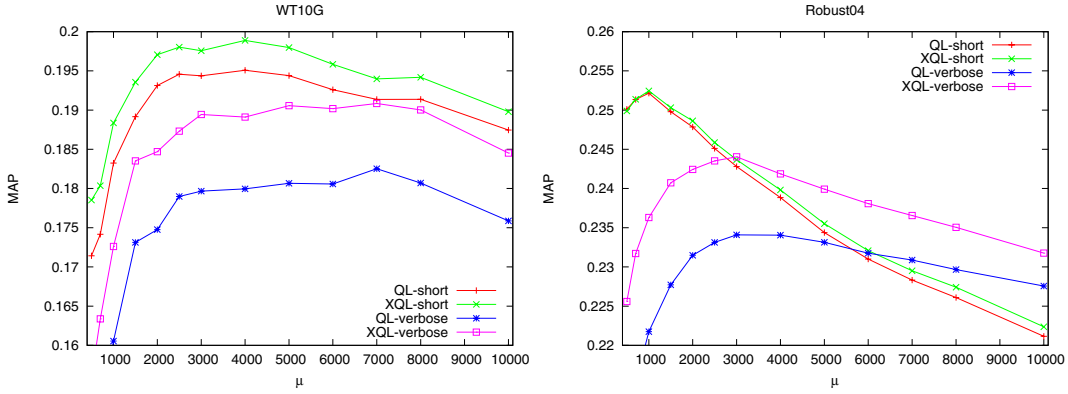

Figure 2: Performance Sensitivity to parameter μ of the standard query likelihood (QL) and the proposed query likelihood with negative query generation (XQL) on WT10G and Robust04. δ is fixed to 0.05 in XQL.

8. REFERENCES

[1] N. Fuhr. Probabilistic models in information retrieval. *The Computer Journal*, 35:243–255, 1992.

[2] D. Hiemstra. *Using Language Models for Information Retrieval*. PhD thesis, University of Twente, January 2001.

[3] E. T. Jaynes. Information theory and statistical mechanics. *Phys. Rev.*, 106(4):620–630, May 1957.

[4] K. S. Jones and S. E. Robertson. LM vs PM: Where's the Relevance? In *Proceedings of the Workshop on Language Modeling and Information Retrieval*, pages 12–15. Carnegie Mellon University, 2001.

[5] K. S. Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: development and comparative experiments. *Inf. Process. Manage.*, 36:779–808, November 2000.

[6] J. Lafferty and C. Zhai. Probabilistic relevance models based on document and query generation. In *Language Modeling and Information Retrieval*, pages 1–10. Kluwer Academic Publishers, 2002.

[7] V. Lavrenko and W. B. Croft. Relevance based language models. In *SIGIR '01*, pages 120–127, 2001.

[8] Y. Lv and C. Zhai. Lower-bounding term frequency normalization. In *CIKM '11*, pages 7–16, 2011.

[9] Q. Mei, H. Fang, and C. Zhai. A study of poisson query generation model for information retrieval. In *SIGIR '07*, pages 319–326, 2007.

[10] D. Metzler, V. Lavrenko, and W. B. Croft. Formal multiple-bernoulli models for language modeling. In *SIGIR '04*, pages 540–541, 2004.

[11] D. R. H. Miller, T. Leek, and R. M. Schwartz. A hidden markov model information retrieval system. In *SIGIR '99*, pages 214–221, 1999.

[12] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *SIGIR '98*, pages 275–281, 1998.

[13] S. Robertson and D. Hiemstra. Language models and probability of relevance. In *In Proceedings of the first Workshop on Language Modeling and Information Retrieval*, pages 21–25. Carnegie Mellon University, 2001.

[14] S. E. Robertson. The probability ranking principle in ir. *Journal of Documentation*, 33(4):294–304, 1977.

[15] S. E. Robertson and K. S. Jones. Relevance weighting of search terms. *Journal of the American Society of Information Science*, 27(3):129–146, 1976.

[16] M. Tsagkias, M. de Rijke, and W. Weerkamp. Hypergeometric language models for republished article finding. In *SIGIR '11*, pages 485–494, 2011.

[17] X. Wang, H. Fang, and C. Zhai. A study of methods for negative relevance feedback. In *SIGIR '08*, pages 219–226, 2008.

[18] C. Zhai. Statistical language models for information retrieval a critical review. *Found. Trends Inf. Retr.*, 2(3):137–213, 2008.

[19] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR '01*, pages 334–342, 2001.