Estimating Retrieval Performance Bound for Single Term Queries

Peilin Yang
University of Delaware
Newark, DE 19716
United States
franklyn@udel.edu

Hui Fang
University of Delaware
Newark, DE 19716
United States
hfang@udel.edu

ABSTRACT

Various information retrieval models have been studied for decades. Most traditional retrieval models are based on bag-of-term representations, and they model the relevance based on various collection statistics. Despite these efforts, it seems that the performance of "bag-of-term" based retrieval functions has reached plateau, and it becomes increasingly difficult to further improve the retrieval performance. Thus, one important research question is whether we can provide any theoretical justifications on the empirical performance bound of basic retrieval functions.

In this paper, we start with single term queries, and aim to estimate the performance bound of retrieval functions that leverage only basic ranking signals such as document term frequency, inverse document frequency and document length normalization. Specifically, we demonstrate that, when only single-term queries are considered, there is a general function that can cover many basic retrieval functions. We then propose to estimate the upper bound performance of this function by applying a cost/gain analysis to search for the optimal value of the function.

1. INTRODUCTION

Developing effective retrieval models has been one of the most important and well-studied topics in Information Retrieval (IR). Various retrieval models have been proposed and studied [9, 10, 14]. Many of them are based on "bag-of-term" representation and leverage only basic ranking signals such as TF, IDF and document length normalization [4]. Although more advanced ranking signals, such as term proximity [11] and term semantic similarity [4, 7], have been integrated into the retrieval functions to improve the retrieval performance, it remains unclear whether we have reached the performance upper bound for retrieval functions using only basic ranking signals. If so, what is the upper bound performance? If not, how can we do better?

To find the performance upper bound is quite challenging: although most of the IR ranking models deal with basic signals, how they combine the signals to compute the relevance scores are quite diverse due to different implementations of IR heuristics [4]. This kind of variants makes it difficult to generalize the analysis. Moreover, typically there are one or more free parameters in the ranking models which can be tuned via the training collections. These free parameters make the analysis more complicated.

This paper aims to tackle the challenge through the simplest problem setup. In particular, we focus on single-term queries and study how to estimate the performance bound for retrieval functions utilizing only basic ranking signals. With only one term in a query, many retrieval functions can be greatly simplified. For example, Okapi BM25 and Pivoted normalization functions have different implementations for the IDF part, but this part can be omitted in the functions for single-term queries because it would not affect the ranking of search results. All the simplified functions can then be generalized to a general function form for single-term queries. As a result, the problem of finding the upper bound of retrieval function utilizing basic ranking signals becomes that of finding the optimal performance of the generalized retrieval function. We propose to use cost/gain analysis to solve the problem [1, 3, 2]. As the estimated performance upper bound of simplified/generalized model is in general better than the existing ranking models, our finding provides the practical foundation of the potentially more effective ranking models for single term queries.

2. PERFORMANCE BOUND ANALYSIS

2.1 A General Form of Retrieval Functions for Single-Term Queries

The implementations of retrieval functions are quite diverse, and it is often difficult to develop a general function form that can cover many retrieval functions. However, if we consider only single-term queries (i.e,. those with only one query term), the problem can be greatly simplified.

Let us start with a specific example. Dirichlet prior function is one of the representative functions derived using language modeling approaches [14], and is shown as follows:

$$f(Q,d) = \sum_{t \in Q} \ln\left(\frac{c(t,d) + \mu \cdot p(t|C)}{|d| + \mu}\right), \tag{1}$$

where $c(t,d)$ is the frequency of term t in document d, $|d|$ is the document length; $p(t|C)$ is the maximum-likelihood of the term frequency in the collection and μ is the model parameter. When a query contains only a term t, the retrieval

ICTIR '16, September 12–16, 2016, Newark, Delaware, USA.
© 2016 ACM. ISBN 978-1-4503-4497-5/16/09…$15.00
DOI: http://dx.doi.org/10.1145/2970398.2970428

Table 1: Instantiations of the general retrieval form

Retrieval Functions	$g(\cdot)$	α	c_1	γ	β	c_2
DIR	1	1	$\mu \cdot p(t\|C)$	0	1	μ
BM25 & BM25+	1	$k_1 + 1$	0	1	$\frac{k_1 \cdot b}{avdl}$	$k_1 \cdot (1-b)$
PIV & PIV+	$1 + ln(1 + ln(\cdot))$	1	0	0	$\frac{s}{avdl}$	$1 - s$
F1EXP & F1LOG	$1 + ln(1 + ln(\cdot))$	$avdl + s$	0	0	s	$avdl$
F2EXP & F2LOG	1	1	0	1	$\frac{s}{avdl}$	s
BM3	1	1	$\mu \cdot p(t\|C)$	μ	k_1	$k_1 \cdot \mu + \mu^2 \cdot p(t\|C)$
DIR+	1	$\mu \cdot p(t\|C) + \delta$	$\mu^2 \cdot p^2(t\|C) + \delta \cdot \mu \cdot p(t\|C)$	0	$\mu \cdot p(t\|C)$	$\mu^2 \cdot p(t\|C)$

function can be simplified to:

$$f(\{t\}, d) = \frac{c(t,d) + \mu \cdot p(t|C)}{|d| + \mu} \qquad (2)$$

Note the natural logarithm function in Equation (1) is omitted since it is a monotonically increasing function and would not affect the ranking results. Since $p(t|C)$ is a collection-dependent constant, the function can be further simplified as:

$$f(t, d) = \frac{c(t,d) + c_1}{|d| + c_2}. \qquad (3)$$

Similarly, Okapi BM25 [9] can be simplified to:

$$\begin{aligned} f(t,d) &= \frac{(k_1+1) \cdot c(t,d)}{c(t,d) + k_1 \cdot (1 - b + b \cdot |d|/avdl)} \\ &= \frac{\alpha \cdot c(t,d)}{c(t,d) + \beta \cdot |d| + c_2}, \end{aligned} \qquad (4)$$

where α absorbs $k_1 + 1$, and $\beta = k_1 \cdot b/avdl$ is a collection-dependent variable and $c_2 = k_1 \cdot (1-b)$ is a parameter.

Furthermore, the pivoted normalization function (PIV) [10] can also be simplified to:

$$\begin{aligned} f(t,d) &= \frac{1 + ln(1 + ln(c(t,d)))}{(1 - s + s \cdot |d|/avdl)} \\ &= \frac{g(c(t,d))}{(\beta \cdot |d| + c_2)}, \end{aligned} \qquad (5)$$

where $g(\cdot) = 1 + ln(1 + ln(\cdot))$ and can be further generalized as an arbitrary non-linear function. $\beta = s/avdl$ is a collection related variable and $c_2 = 1 - s$ is a parameter.

All of the above three simplified functions (i.e., Eq. (3), Eq. (4) and Eq. (5)) can be generalized as the following form:

$$F(c(t,d), |d|) = \frac{\alpha \cdot g(c(t,d)) + c_1}{\gamma \cdot c(t,d) + \beta \cdot |d| + c_2}, \qquad (6)$$

where $g(\cdot)$ is an arbitrary non-linear function and $\alpha, \beta, \gamma, c_1, c_2$ are free parameters. This generalized function form is essentially a linear transformation of a non-linear transformation of term frequency divided by a linear transformation of document length. The denominator optionally adds adjusted term frequency as a method to dampen the impact of increasing term frequency. Note that IDF is not part of the function because it would not affect the document ranking for single-term queries.

In fact, we find that the generalized retrieval function as shown in Eq. (6) can cover at least 11 retrieval functions. In addition to the above three retrieval functions, the following functions can also be generalized: (1) F1EXP, F1LOG, F2EXP and F2LOG from the axiomatic retrieval models [5], (2) BM3 derived from the Dirichlet Priors for term frequency normalization model [6], and (3) BM25+, DIR+, PIV+ derived from the lower bounding term frequency normalization models [8]. Table 1 summarizes the instantiations for each of the retrieval functions.

2.2 Upper Bound Estimation for MAP

Given the general form as shown in Equation (6), one straightforward solution to estimate the performance bound for single-term queries would be to simply try all possible values/instantiations for the parameters and functions and then report the best performance. Thus, the problem of estimating performance bound boils down to the problem of searching for optimal parameter settings in terms of the retrieval performance. More specifically, given Eq. (6), we need to find parameter settings for $\alpha, \beta, \gamma, c_1, c_2$ that can optimize the retrieval performance measured (i.e., MAP in this paper). We do not consider the instantiation of $g(\cdot)$ here, and leave it as our future work.

Since it is infeasible to try all possible parameter values and find the optimal setting, we propose to apply the cost/gain analysis to find the optimal parameter setting.

Let us explain the notations first. d_i and d_j are a pair of documents. Given a query, $s_i = f(\{t\}, d_i)$ and $s_j = f(\{t\}, d_j)$ denote the relevance score of these two documents computed using a retrieval function.

For a given query, each pair of documents d_i and d_j with different relevance labels (currently we only consider the binary case, i.e. whether the document is relevant or non-relevant) a ranking model computes the scores $s_i = f(d_i)$ and $s_j = f(d_j)$. Follow the previous studies about RankNet [1, 2], we define the cost function as the pairwise cross-entropy cost applied to the logistic of the difference of the relevance scores:

$$C_{ij} = \frac{1}{2}(1 - S_{ij})\sigma(s_i - s_j) + \log(1 + e^{-\sigma(s_i - s_j)}) \qquad (7)$$

where $S_{ij} \in \{0, \pm 1\}$ denotes the ground-truth ranking relationship of document pair d_i and d_j: 1 if d_i is relevant and d_j is non-relevant, -1 if d_i is non-relevant and d_j is relevant, 0 if they have the same label. The gradient of the cost function is then:

$$\frac{\partial C_{ij}}{\partial s_i} = \sigma\left(\frac{1}{2}(1 - S_{ij}) - \frac{1}{1 + e^{\sigma(s_i - s_j)}}\right) = -\frac{\partial C_{ij}}{\partial s_j} \qquad (8)$$

If we only consider the total cost of ranking non-relevant documents before the relevant documents, S_{ij} is always 1. We will always consider that d_i is relevant and d_j is non-

Table 2: collections and queries

	disk12	Robust04	WT2G	GOV2
#queries	4	11	3	2
qid	57,75, 77,78	312,348,349, 364,367,379, 392,395,403, 417,424	403,417, 424	757,840

relevant from now on. The Eq. (8) is then simplified as:

$$\frac{\partial C_{ij}}{\partial s_i} = \frac{-\sigma}{1 + e^{\sigma(s_i - s_j)}} \tag{9}$$

The upper bound of the performance is then obtained when the cost is minimized by parameters optimization. The parameters $\mathrm{p_k} \in \mathbb{R}$ used in the ranking model could be updated so as to reduce the cost via stochastic gradient descent:

$$\mathrm{p_k} \rightarrow \mathrm{p_k} - \eta \frac{\partial C}{\partial \mathrm{p_k}} = \mathrm{p_k} - \eta \left(\frac{\partial C}{\partial s_i} \frac{\partial s_i}{\partial \mathrm{p_k}} + \frac{\partial C}{\partial s_j} \frac{\partial s_j}{\partial \mathrm{p_k}} \right) \tag{10}$$

Unfortunately, the cost defined in Eq. (9) is actually the "optimization" cost instead of the target cost (the actual cost) [1] and thus minimizing the cost may not necessarily lead to the optimal MAP. However, MAP is either flat or non-differentiable everywhere which makes the direct optimization toward it difficult [13]. To overcome this we modify Eq. (9) by multiplying the derivative of the cost by the size of the change in MAP gain from swapping a pair of differently labeled documents for a given query q. The pairwise λ (we change cost C to λ and λ is the gain instead of cost) can be written as:

$$\lambda_{ij} = \frac{\sigma}{1 + e^{\sigma(s_i - s_j)}} \frac{1}{|R|} \left(\left| \frac{n}{r_j} - \frac{m}{r_i} \right| + \sum_{k=r_j+1}^{r_i-1} \frac{\mathrm{I}(k)}{k} \right) \tag{11}$$

where r_i and r_j are the ranking positions of d_i and d_j; m and n are the number of relevant documents before position r_i and r_j; $\mathrm{I}(k) = 1$ if the document at kth position of the ranking list is relevant and 0 otherwise; $|R|$ is the number of relevant document for the query. The model parameters are adjusted based on the aggregated λ for all pairs of documents for the query using a small (stochastic gradient) step.

The optima are local optima with 99% of the confidence by following the Monte-Carlo method with model parameters chosen from 459 random directions [3].

3. EXPERIMENTS

3.1 Testing Collections

We use four TREC collections: disk12, Robust04, WT2G and Terabyte (GOV2) to conduct the experiments. For the queries, only the title fields of the query topics with only one query term are used (20 in total). We use Dirichlet language model with default $\mu = 2500$ to retrieve at most top 10,000 documents as the documents pool for the pairwise comparison for each query. For relevance labels that less or equal to zero is treated as non-relevant and labels greater than zero are treated as relevant. An overview of the involved collections and queries are listed in Table 2.

3.2 Experiment Setup

We tested both using the cost function only and using the cost function together with λ component of MAP. The results are very close and the cost with λ seems to be a little bit superior so we just report that part of the results. We basically tried several different models based on Eq. (6):

Table 3: Upper Bound of MAP

		disk12	Robust04	WT2G	GOV2
Models with Basic Signals	DIR	0.4009	0.3823	0.3660	0.2083
	BM25	**0.4016**	**0.3824**	**0.4038**	0.2896
	PIV	0.3987	0.3812	**0.4038**	**0.3079**
	F2EXP	0.4000	0.3682	0.3183	0.1950
	BM3	0.4015	0.3823	0.3792	0.2554
	DIR+	0.4009	0.3823	0.3794	0.2083
Upper Bounds	DIR^U	0.4244†	0.4136†	0.4055	0.2724
	TFDL1^U	0.4273†	0.4209†	0.4095	0.3193†
	TFDL2^U	**0.4273**†	**0.4209**†	**0.4095**	**0.3255**†

- **DIR**^{U}: Dirichlet Language Model, denoted as $\frac{c(t,d)+\mu \cdot p(t|C)}{|d|+\mu}$
- **TFDL1**^{U}: which only contains c_1 and c_2 as model parameters, denoted as $\frac{c(t,d)+c_1}{|d|+c_2}$
- **TFDL2**^{U}: which takes α, β, c_1, c_2 as parameters, denoted as $\frac{\alpha \cdot c(t,d)+c_1}{\beta \cdot |d|+c_2}$

For other possible format of Eq. (6) they are essentially covered by TFDL2^U so we do not report the results for them [1].

For all of our experiments, we varied the learning rate η between 10^0 to 10^{10} with step size 10 times to previous value. We have found that optimal learning rate brings marginal gain in terms of overall performance. So we just report the performance on the optimal learning rate. For the starting point, we choose α, β, c_1, c_2 from $[0.1, 10000]$ with step size 10 times to previous value. We set the learning iteration at most 500 epochs and it stops if the gain was constant over 20 epochs.

3.3 Results

Table 3 lists both the optimal performances of previously proposed ranking models with optimal parameters chosen from a wide range (e.g. for DIR and DIR+ $\mu \in [0, 5000]$ with step size 500; for BM25, BM3, PIV, F2EXP b or $s \in [0, 1]$ with step size 0.1) and optima of proposed models. The values listed in the table are the MAPs of single term queries only (not the whole set of the queries). It is shown that the generalized models are better than classic ranking models for the most cases (indicated by the † which means the two-tailed paired t-test at p value of 0.05 comparing with the optimal performances of selected models which are boldfaced). Furthermore, different collections have different gains. Robust04 has the largest gain between the two results which indicating that possibly the previously proposed ranking models do not capture the critical ranking signals well or the statistics they use contradicts with the actual properties of relevant documents. Also, for WT2G we get very little gain by applying our analysis (the performances are even not significant better than the selected models). This probably

[1] Actually they are possibly covered by Eq. (6). But if we choose wide spectrum of the starting points then they are covered by large chance.

Table 4: Parameters

Model	Paras	disk12	Robust04	WT2G	GOV2
DIR^U	μ	4.66e3	3.54e7	1.43e6	0
TFDL1^U	$\frac{c_1}{c_2}$	2.49e-3	1.0	6.87e-5	6.0e-1
TFDL2^U	$\frac{c_1}{c_2}$	1.55e-2	5.86e-2	1.08e1	1.39e-1
	$\frac{\alpha}{\beta}$	1.37e-4	1.43e-2	1.01e-2	1.13e-2

means that if we would like to further improve the performance on WT2G we need to find other forms of the ranking models which may look different than Eq. (6).

3.4 Parameters

Next, we would like to investigate the parameters that lead to the optima for the proposed models. The parameters are worthy to look at since they might inspire or provide intuition of better performing models in the future. Table 4 lists those parameters. As we can see, for DIR^U the optimal parameters μ obtained for Robust04 and WT2G are much larger than 10^3 which is suggested value by the original authors of DIR [14]. For TFDL1^U we choose to report the ratio $\frac{c_1}{c_2}$. The values vary between collections. For example, the optimal values for Robust04 is 1.0 which indicates that the better performed models would have larger dampen factor for document length than other collections. For TFDL2^U both $\frac{c_1}{c_2}$ and $\frac{\alpha}{\beta}$ are reported. We find that α is in several magnitude levels smaller than β. But this is not always the truth for $\frac{c_1}{c_2}$. We would expect more impact on $\frac{\alpha}{\beta}$ than $\frac{c_1}{c_2}$ and the values of $\frac{\alpha}{\beta}$ could be better incorporated by better performing models in the future.

4. RELATED WORK

Although there are lots of effective ranking models proposed by researchers, there are fewer studies dedicated to the theoretical analysis of their performances upper bound. One related domain is the constraint analysis [4] which proposes formal constraints that a reasonable ranking model should bear. Examples of the constraints including how should a ranking model incorporate TF, how to regulate the interaction of TF and DL, how to penalize long document in the collection, etc. The constraint analysis provides a general guide of how a reasonable ranking model should be designed. Our work further explores this direction by providing the practical performance upper bound as well as the optimal parameters which helps to fine tune the constraint theory.

Our estimation method is mostly inspired by the RankNet [1, 2] and the LambdaRank [2, 3] which are successful in the learning to rank domain. In their works they apply the pair-wise documents comparison for a specific query which is also adopted by our work. However, we did two different things in our work: (1) the aforementioned techniques apply neural network as the underlying model while we follow the rationale proposed by some classic ranking model, i.e. the ranking score should be positively correlated with TF and inversely correlated with DL, to find the local optimum of the generalized ranking models. (2) we aim to optimize MAP instead of NDCG and we proposed a simplified equation for calculating the difference of MAP if two documents are swapped in the ranking list which can make the analysis more efficiency. There is another work which indeed directly optimizes MAP called SVMMAP [13]. SVMMAP is actually another learning to ranking algorithm based on support vector machine. It performs optimization only on a working set of constraints which is extended with the most violated constraint at each optimization step. Taylor et al. [12] used the cost analysis to predicate a family of BM25 ranking models. They however did not apply the gain analysis which has shown to be superior in our experiments.

5. CONCLUSIONS AND FUTURE WORK

In this paper we have applied cost/gain analysis to the performance upper bound of single term queries for TREC collections. The found upper bounds of MAP provide sound foundation of potentially better performed ranking models in the future. Moreover, the parameters that lead to the local optimums provide more insight about how the future models could better incorporate proper statistics/signals.

Future work may include expanding the analysis to multiple terms queries and finding more mathematically restricted way to prove the performance upper bounds.

6. ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under Grant Number IIS-1423002. We thank the reviewers for their useful comments.

7. REFERENCES

[1] C. Burges, R. Ragno, and Q. Le. Learning to rank with non-smooth cost functions. In *Advances in Neural Information Processing Systems 19*. MIT Press, Cambridge, MA, January 2007.

[2] C. J. Burges. From ranknet to lambdarank to lambdamart: An overview. Technical Report MSR-TR-2010-82, June 2010.

[3] P. Donmez, K. M. Svore, and C. J. Burges. On the local optimality of lambdarank. In *SIGIR*. Association for Computing Machinery, Inc., July 2009.

[4] H. Fang, T. Tao, and C. Zhai. A formal study of information retrieval heuristics. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 49–56, New York, NY, USA, 2004. ACM.

[5] H. Fang and C. Zhai. An exploration of axiomatic approaches to information retrieval. SIGIR '05, pages 480–487, New York, NY, USA, 2005. ACM.

[6] B. He and I. Ounis. A study of the dirichlet priors for term frequency normalisation. SIGIR '05, pages 465–471, New York, NY, USA, 2005. ACM.

[7] X. Liu and H. Fang. Latent entity space: a novel retrieval approach for entity-bearing queries. *Information Retrieval Journal*, 2015.

[8] Y. Lv and C. Zhai. Lower-bounding term frequency normalization. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 7–16, New York, NY, USA, 2011. ACM.

[9] S. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. pages 109–126, 1996.

[10] A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. SIGIR '96, pages 21–29, New York, NY, USA, 1996. ACM.

[11] T. Tao and C. Zhai. Regularized estimation of mixture models for robust pseudo-relevance feedback. SIGIR '06, pages 162–169, New York, NY, USA, 2006. ACM.

[12] M. Taylor, H. Zaragoza, N. Craswell, S. Robertson, and C. Burges. Optimisation methods for ranking functions with multiple parameters. CIKM '06, pages 585–593, New York, NY, USA, 2006. ACM.

[13] Y. Yue, T. Finley, F. Radlinski, and T. Joachims. A support vector method for optimizing average precision. SIGIR '07, pages 271–278, New York, NY, USA, 2007. ACM.

[14] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22(2):179–214, Apr. 2004.