

## A Machine Learning Approach to Improve Language Model Retrieval on Multiple Content Fields

Yuan LIN<sup>†</sup>, Hongfei LIN, Zheng YE, Sui SU

*Department of Computer Science and Engineering, Dalian University of Technology, Dalian 116023, China*

### Abstract

Language model is widely used on information retrieval field. In spite of its widely use, There have been few studies examining its effectiveness on a document description over multiple field combinations. In this paper, we develop a machine learning approach to language model retrieval, using RankBoost, from the results of language model methods on multiple fields. The experimental results show that our approach can consistently improve retrieval effectiveness across several LETOR 3.0 data sets.

*Keywords:* Information Retrieval; Language Model; Machine Learning; RankBoost.

### 1. Introduction

Language model [1, 2] approaches to information retrieval are attractive and promising because they connect the problem of retrieval with that of language model estimation. The basic idea of these approaches is to estimate a language model for each document, and to rank documents by the likelihood of query according to the estimated language model. Yet this framework is very promising because of its foundations in statistical theory. A central issue in language model estimation is smoothing, the problem of adjusting the maximum likelihood estimator to compensate for data sparseness. It proposed several popular smoothing methods: Jelinek-Mercer method, Bayesian smoothing using Dirichlet priors' method and Absolute discounting method as three important approaches for language model. These language model approaches have served as strong baselines widely used in the information retrieval community and the fact that very simple language modeling retrieval methods have performed quite well effectively.

Recent study suggests that it is not the best way to index the document as a whole unit. Many researches have taken the multiple fields for a document especially the Web document into account to improve the accuracy [3, 4]. Some search engine also using fields' information to improve the retrieval results. The popular fields used widely are content fields, such as the title, abstract, body and etc. The retrieval scores of all the fields may reflect the relevance of the document to the query.

The general methods to utilize these results are that one is empirically weighted the scores of the fields; the second is simple seem these fields as one unit to give relevance scores; but it is difficult to choose the

---

<sup>†</sup> Corresponding author.

Email addresses: [yuanlin@mail.dlut.edu.cn](mailto:yuanlin@mail.dlut.edu.cn) (Yuan LIN)

perfect parameters by empirical methods. However, a machine learning approach may easily find an applicable ranking model to different data sets for multiple fields after learning from its training set.

RankBoost [5] is a pair-wise method widely used on field of learning to rank. Many researches are based on this approach [6, 7, 8]. The researches show that the RankBoost approach is very effective for ranking. In this paper our main interest is using RankBoost as the machine learning tool in order to improve the effectiveness for language model approaches on multiple content fields.

The rest of paper is divided as follows: Section 2 introduces the related work about retrieval on multiple fields. In Section 3 we review the content fields of documents. Then Section 4 provides an overview of language model. In Section 5 we introduce RankBoost approach and discuss how to use it to learn a ranking model using language model retrieval features. Section 6 we describe our experiment and present our results. Finally, we conclude this work and point out some directions for future research in Section 7.

## 2. Related Work

For multiple fields' retrieval, some researches have made an effort to improve the effectiveness of results. Lucene [9] has proposed an approach that weighted fields on the indexing phase, and the weight parameters can be modified artificially. As a language model approaches based retrieval system, Indri [10] also provides the multiple fields retrieval. Different from Lucene, it gives the weight parameters of fields on the phase of query searching, which is set by users. Both of these two search systems set the fields' parameters in the empirical and artificial way, and this way may not adapt to the different data sets, and it is difficult to choose a perfect set of parameters for any data sets in this way. Recently, some other information retrieval models have also been used in multiple fields' retrieval. Stephen Robertson et al.[3] develop a BM25 extensional model:BM25F to multiple weighted fields, according to setting  $3K+1$ ( $K$  is the number of fields) to fused the term frequencies, document frequencies and field lengths using BM25 model to give every document a relevance scores. Krysta M. Svore et al. [4] propose a BM25-style retrieval model using a data-driven approach based on the idea of BM25F. Differently, they using a machine learning method: LambdaRank [11] to learn a two layers-neural net model using the same input of BM25F for ranking; The experimental results explain this way can improve the performance of ranking. However, the learning features that they choose are term frequencies for terms in positions 1-10, the document frequencies for terms in positions 1-10, and field length for all fields. It seems the approach of choosing features in this way is not sufficiency. So in this paper we accept its concept of data-driven to use a machine learning method to learn a ranking model from multiple fields ranking list directly based on language model retrieval approaches, to improve the ranking results. We use different language model smooth methods of different fields as learning features. Our main aim is using machine learning method to obtain a ranking model, the performance of which is better than general methods using language model on multiple fields. Finally, we want to develop an approach to improve other information retrieval approaches on multiple fields.

## 3. Document Fields

There may be many fields in a Web document, such as title, URL, body, abstract, anchor text and etc. In this paper we choose language model as retrieval method to get ranking scores, so we use content text

fields as test fields to exert language model approaches. The content fields of a document include title text, URL text, body text, abstract text, anchor text, which we use in this paper. The title field contains the title of document. The URL field contains the text of the page's web address. The body field consists of the html content of page. The abstract text contains the main idea of a document. Anchor text is the text associated with a link in a source document, which is assumed to describe the target document. All the information from the fields may be related to the relevance, so we extract the ranking features based on these fields using language model method.

#### 4. Language Model

The basic idea of the language model approach [1] can be explained as follows: A query  $q$  is generated by a probabilistic model based on a document  $d$ . Given a query  $q=q_1q_2\dots q_n$  and a document  $d = d_1d_2\dots d_m$ . It is important to estimate the conditional probability  $p(d|q)$ , the probability that  $d$  generates the observed  $q$ . After applying the Bayes' formula and dropping a document-independent constant (since we are only interested in ranking documents), we have:

$$p(d | q) \propto p(q | d)p(d)$$

In this paper we are interested in the smoothing approaches which give the relevance scores of the documents. The term smoothing refers to the adjustment of the maximum likelihood estimator of a language model so that it will be more accurate. The main idea of smoothing is to assign a non-zero probability to unseen words using the information of the document collection. Based on this idea, there are three methods as follows.

Jelink-Mercer method, which involves a linear interpolation of the maximum likelihood model with the collection model, using a coefficient  $\lambda$  to control the influence of each model. Bayesian smoothing using Dirichlet priors: a language model is a multinomial distribution, for which the conjugate prior for Bayesian analysis is the Dirichlet distribution. Absolute discounting: the idea of the absolute discounting method is to lower the probability of seen words by subtracting a constant from their counts [12]. The approach discounts the seen word probability by subtracting a constant. For every word in query, we get the value of  $p(w|d)$  in the document, after accumulate their logarithms, we can get the final relevance scores.

The language model approaches is helpful to deal with unseen query words, especially for these content fields with a few words, the smoothing methods take global information into account is a particularly important. The global information is different with respect to different field. We use these three smoothing approaches to obtain the language model retrieval scores, and we can obtain three types of ranking features by these methods, and we can examine whether it is helpful to improve the retrieval results by learning with different retrieval approaches for ranking model. Then we seem single field of every document as an independent unit for extracting language model ranking features. After getting the features, we choose RankBoost as our machine learning approach.

#### 5. RankBoost

Learning to rank is a research field whose interest focused on finding a ranking model which take the performance of the ranking features into account to improve the retrieval results. The ranking features may

be the information retrieval methods or anything that is related to the relevance of the documents to the query. RankBoost is a kind of pairwise learning to rank approach which can reduce ranking to classification on document pairs with respect to the same query, no longer assume absolute relevance. Its primary task is to make document pairs based on relevance judgments with respect to the same query.  $\langle x_0, x_1 \rangle$  denotes a document pair, and  $x_0$  is unrelevance document,  $x_1$  is relevance document. A high weight assigned to a pair of documents indicates a great importance that the weak learner orders that pair correctly. We also assign to the weak learner to show its performance of predicting the document relevance. Once iterative it generates a weak learning. Reserving weak learning of each iterative and accumulating them multiplied by their weights; we can obtain the final ranking model. RankBoost Algorithm [5] we use is as table 1 showing.

Weak rankings have the form  $h_t(x)$ . We think of these as providing ranking information in the same manner as ranking features and the final ranking. The weak learners we used in our experiments are based on the original given ranking features and new features extracted by retrieval approaches. We focus in this section and in our experiments on  $\{0, 1\}$ -valued weak rankings that use the ordering information provided by the ranking features, but ignore specific scoring information. In particular, we will use weak rankings  $h$  of the form: if  $f_i > \theta$ ,  $h(x) = 1$ ; else  $h(x) = 0$ . That is a weak ranking derived from a ranking feature  $f_i$  by comparing the score of  $f_i$  on a given instance to a threshold  $\theta$ .  $\alpha_t$  is the weight of the weak learning which can be generated at each iteration and computed by RankBoost. It shows the importance of the corresponding weak learning, with respect to predicting the relevance of document.  $Z_t$  is a normalization factor. In this paper, we seem the results of language model approaches as training features, and utilize RankBoost to obtain a model that includes multiple ranking features based on language model to decide the relevance of the documents to query, which finally be indicated as ranking scores.

Table 1 RankBoost for Learning Ranking Model

Algorithm 2 RankBoost for learning ranking model
Input: Training set T; Relevance judgments R;
Output: Ranking Function H(x);
Start:
1: Pair(T,R) => Document pair set D;
2: Initialize(D) => $D_t : \forall D_1(x_0, x_1) = 1/ D $ ;
3: For t=1,...,T:
1) Train weak learner using distribution $D_t$ ;
2) Get weak ranking $h_t$ ;
3) Choose $\alpha_t \in \mathbb{R}$ ;
4) Update: $D_{t+1}(x_0, x_1) = \frac{D_t(x_0, x_1) \exp(\alpha_t (h_t(x_0) - h_t(x_1)))}{Z_t}$ ;
4: Output the final ranking: $H(x) = \sum_{t=1}^T \alpha_t h_t(x)$ .
End

## 6. Methodology

### 6.1. Experimental Methods

In the experiment, we firstly consider the scores of one smoothing method of language model on multiple fields as features which is used to learn a model to predict the relevance of documents and queries. There are three smoothing approaches used by language model: Jelinek-Mercer method (JM), Bayesian smoothing using Dirichlet priors' method (Dir) and Absolute discounting method (Dis), for example, we use Jelinek-Mercer as ranking method to obtain scores from the content fields, such as title, body and etc. then for every fields, there is a ranking list with scores, and we seem the ranking scores from one field as a ranking feature. If there are K content fields in the document, we can get K ranking features. Then we can use RankBoost to learn a model from the features with the labeled data, and we can predict the ranking scores from unlabeled data, and examine whether the model is helpful to improve the ranking results. After that we deal with the documents using other two smoothing approaches in same way. Finally, we add all the features generated from all the content fields using these three smoothing approaches of language model, to the training process to learn a new model to examine whether it is useful to improve the results further.

### 6.2. Data Set

We evaluate our method on the Letor3.0 [13] data set released by Microsoft Research Asia. This data set contains two data sets: the OHSUMED data set, the GOV data set. The OHSUMED data set is derived from medicine retrieval task, while GOV comes from TREC task.

Letor3.0 is based on many query-document features. The features may be tf-idf, BM25、HITS、Page Rank and LMIR [2] etc. The data set also provides the Meta information about documents and queries to mine ranking features further. Although the data set include additional attributes, we train our model only use the ranking features from multiple fields obtained by using language model approaches, because we would like to maintain a fair comparison to the general language model approaches for the multiple retrieval as they are so widely used.

The sub-sets that we use for our research are TD2003, HP2003 and NP2003 from GOV data set and OHSUMED data set. For GOV data set there are four content fields: title, body, anchor, URL; while there are two content fields: title, abstract for OHSUMED, and we extract the ranking features based on these fields for different data sets. Our goal is to have our RankBoost model demonstrate improved accuracy on multiple fields.

### 6.3. Experimental Results

In order to evaluate the performance of the proposed approach, we adopt MAP [14] as evaluation method. The average precision of a query is the average of the precision scores after each relevant document retrieved. Average precision (AP) takes the positions of relevance documents on ranking list into account to give scores of the list with respect to one query. Finally, MAP is obtained by the mean of the average precision over a set of queries. There are four data subsets chosen from letor3.0, and 5 groups of training set and test set in each subset. Our experiments give the results of RankBoost model learn from language model ranking features compared with two baseline language model approaches based on the whole

document (Baseline1) and linear interpolation approach with the same weight to every field retrieval scores(Baseline2). According to the experiment, we can get the performance of the baseline1 、baseline2 and the results of the prediction of RankBoost model, and we can examine whether it is availability to use the ranking model training with the ranking feature of the different smoothing approaches.

Table 2 shows the comparing RankBoost approach with the baseline approaches on the Letor3.0 data set. For every sub-set, JM、Dir and Dis are the smoothing approaches of language model. For the OHSUMED we set the parameter of JM  $\lambda = 0.5$ ; the parameter of Dir  $\mu = 50$ ; the parameter of Dis  $\delta = 0.5$ ; this is the parameter setting for the OHSUMED set. The parameters we choose for GOV data set are  $\lambda = 0.1$ ;  $\mu = 2000$ ;  $\delta = 0.7$ . The results show the average values to the five group test for sub-datasets.

Table 2 MAP of the Test Sets in Letor3.0

Data Set	Method	Baseline1	Baseline2	RankBoost
OHSUMED	JM	0.3890	0.4271	0.4451
	Dir	0.4436	0.4251	0.4430
	Dis	0.4361	0.4261	0.4444
TD2003	JM	0.1176	0.1298	0.1292
	Dir	0.0668	0.1031	0.1348
	Dis	0.1230	0.1476	0.1624
HP2003	JM	0.4110	0.4700	0.5160
	Dir	0.3109	0.4380	0.5259
	Dis	0.4804	0.4921	0.5335
NP2003	JM	0.5264	0.5483	0.6277
	Dir	0.4782	0.4601	0.5248
	Dis	0.5848	0.5380	0.6355

We design the different ranking features for different data sets based on the number of their content fields. For OHSUMED, because it only has two fields: title and abstract, we can only obtain two ranking features by language model smoothing approach. And there are four fields in the GOV documents: title, anchor, URL, and body, so we can get four ranking features from these data sets.

#### 6.4. Experimental Analysis

The experiment results show that the model trained by RankBoost exhibits superior accuracy for content fields on the four sub-sets of letor3.0 data set. For each data set, we made three groups of experiments using different language model smoothing methods. We find that the performance of ranking model is better than the general methods using language model retrieval on multiple fields. For baseline 1, it describes the whole document as a single field; it is easy to get the ranking scores in short time. However, it loses the information of fields, while the scores come from the field, which can decide the relevance of the document independently. But for the limitation of information that the single field contains, a single

ranking list based on one field can not represent the actual relevance perfectly, so it is necessary to find a model based on these fields ranking features to decide the final ranking. It is a widely used method to assign the weights to the different fields using linear model; including some famous retrieval systems such as Lucene and Indri which use this approach. We take the linear interpolation approaches into account to construct baseline 2, for simplicity we set the weight of different fields with the same value; otherwise, the bigger weight means the greater risk, so we give all the fields the equal chance to devote themselves to the final ranking list. However, it is difficult to find a group of perfect weight parameters in this way; especially the actual ranking model may be not linear. Even if, it can be evaluated from validation set but it can be take too much time and not perfectly represent the actual performance of the field ranking. Derived from the idea of data-driven, we try to find a ranking model that learns from data itself to avoid the problem of parameter tuning. Learning to rank provides us a good machine learning way to find a ranking model automatically. We use RankBoost to maximize the number of the preference document pairs with correctly order based on relevance [5] on the training set (if all the order of the pairs in the ranking list is correct, we can get the best accuracy ranking.), then we can get a model based on field ranking features to predict the relevance more accurately. The main factor that we choose language model approaches for extracting ranking feature is that language model retrieval approaches can provide more ranking methods based on different smoothing approaches, which can be used to verify that our approach is availability to different methods used to multiple retrieval. From the results of experiment, we can calculate the gain of our ranking model comparing with the baseline methods, On the OHSUMED set we get 4% increased on average level for the three methods, which we evaluate by MAP. There are round 10% promotion for the other data set. The performance of our ranking model verifies the machine learning method for multiple fields' retrieval is effective.

Finally, since it can get good performance on results by using ranking model, we want to examine whether is helpful to extend the feature space. The features space is constructed by the ranking list obtained by multiple retrieval methods on multiple content fields. For example, there are N kinds of retrieval methods and M types of content fields. The number of the features must be  $M*N$ . Now we set the retrieval methods as the language model approaches, so we can get 6 ranking features from OHSUMED set; and 12 ranking features from GOV set. We also designing two contrast experiments: multi-baseline1, we also make a linear interpolation model for comparing; multi-baseline2, we select the results from all the above experiment with the best performance, in order to find out whether the new feature space contributes to improve the results further. For simplicity we exhibit the average results of five test set as the performance of the retrieval methods (baseline1, baseline2, RankBoost) on the four data sets. The approach we use more ranking features from multiple language smoothing method is called multi-RankBoost. Table 3 shows the results. The type of smoothing method for extracting ranking features is shown in parentheses.

According to the table 7, we can see that the performance of linear interpolation model degrading further as well as feature space expended, comparing with the RankBoost model. But the Multi-RankBoost model can't exhibit the super accuracy over the best ranking model from single smoothing approach feature space. We conclude that for these data set, extending the feature space in this way makes little different. However, the results of multi-baseline2 is due to best performance of the single language model approach, and they are obtained after comparing other two approaches, while Muti-RankBoost need not to take the performance of different retrieval methods into account, which can provide a ranking model whose

performance is similar to the best of the single method. But it can't be ignored that learning for Multi-RankBoost model need more features, which increases the training time. In general case, the single retrieval method for extracting multiple fields ranking features to learning ranking model is enough to improve the ranking results.

Table 3 Average MAP of the test sets in OHSUMED、TD2003 、HP2003 and NP2003

	Multi-baseline1	Multi-baseline2	Multi-RankBoost
OHSUMED	0.4262	0.4451(JM)	0.4455
TD2003	0.1479	0.1624(Dis)	0.1622
HP2003	0.5190	0.5535(Dis)	0.6026
NP2003	0.5412	0.6355(Dis)	0.6291

## 7. Conclusion

In this paper, we propose an approach whose idea is derived from data-driven. We apply RankBoost to learn a ranking model based on ranking features from multiple fields. Considering the diversity of retrieval, we choose the language model as basic retrieval method for ranking features extracted from multiple content fields. The experiment results show that the machine learning model is effective to improve the performance of ranking based on three smoothing approaches for language model retrieval. In final experiment we try to learn a ranking model with all the features we get from multiple fields using language model approaches; the result is similar to that of single retrieval method with best performance, so it may be a good method to deal with the retrieval task including multiple content fields and multiple retrieval approaches. Finally our work also display that the approach of learning to rank may be still effective in small ranking feature space for unique retrieval task. Possible extensions and future work include: the experimental results shown that our approach is availability which apply language model as ranking method, we will test whether other retrieval methods would be effective in this way; we will also try other learning to rank model to multiple fields' retrieval, especially for the final experiment it may improve the performance further.

## Acknowledgement

This work is supported by grant from the Natural Science Foundation of China (No.60673039 and 60973068), the National High Tech Research and Development Plan of China (2006AA01Z151) and the Ph.D. Programs Foundation of Ministry of Education of China (No. 20090041110002 ).

## References

- [1] Chengxiang Zhai and John Lafferty. A Study of Smoothing Methods for Language Models Applied to Information Retrieval. In: ACM Transactions on Information Systems, vol.22 no.2, 2004, pp. 179-214
- [2] Chengxiang Zhai and John Lafferty. A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval. In: Proceedings of SIGIR2001, pp. 334-342
- [3] Stephen Robertson, Hugo Zaragoza and Michael Taylor. Simple BM25 Extension to Multiple Weighted Fields. In: Proceedings of CIKM2004, pp.42-49
- [4] Krysta M. Svore and Christopher J. C. Burges. A Machine Learning Approach for Improved BM25 Retrieval. In:

ACM International Conference on Information and Knowledge Management, Hong Kong,2009,1811-1814

- [5] Freund, Y., Iyer, R., Schapire R., Singer, Y.: An efficient boosting algorithm for combining preferences. In: Journal of Machine Learning Research, vol.4,2003, pp. 933-969
- [6] Amini, M.-R., Truong, T.-V., Goutte, C.: A Boosting Algorithm for Learning Bipartite Ranking Functions with Partially Labeled Data. In: ACM Special Interest Group on Information Retrieval, Singapore ,2008, pp. 99-106
- [7] Duh, K., Kirchhoff, K.: Learning to Rank with Partially-Labeled Data. In: ACM Special Interest Group on Information Retrieval, Singapore ,2008, pp. 251-258
- [8] Zhou, K., Xue, G.-R., Zha, H.-Y., Yu, Y.: Learning to Rank with Ties. In: ACM Special Interest Group on Information Retrieval, Singapore ,2008,pp. 275-282
- [9] Zhe Qiu, Taotao Fu. Lucene 2.0 + Heritrix Search Engine. In: Post&Telecom Press. 2007.6
- [10] <http://www.lemurproject.org/indri/>
- [11] Chritoper J.C. Burges, Robert Ragno, Quoc Viet Le. Learning to Rank with Nonsmooth Cost Functions. In: Neural Information Processing Systems, Vancouver, Canada ,2006, pp.193-200
- [12] H.Ney, U.Essen, and R.Schwartz. On structuring probalilistic dependencies in stochastic language modeling, In: Computer Speech and Language,1994,pp. 1-38
- [13] <http://research.microsoft.com/en-us/um/beijing/projects/letor/letor3dataset.aspx>
- [14] Järvelin, K., Kekäläinen, J.: IR evaluation methods for retrieving highly relevant documents. In: the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, USA ,2000,pp. 41–48