Optimizing Two–Stage Bigram Language Models for IR

Sara Javanmardi
University of California, Irvine
sjavanma@ics.uci.edu

Jianfeng Gao,Kuansan Wang
Microsoft Research
{jfgao, kuansan.wang}@microsoft.com

ABSTRACT

Although higher order language models (LMs) have shown benefit of capturing word dependencies for Information retrieval (IR), the tuning of the increased number of free parameters remains a formidable engineering challenge. Consequently, in many real–world retrieval systems, applying higher order LMs is an exception rather than the rule. In this study, we address the parameter tuning problem using a framework based on a linear ranking model in which different component models are incorporated as features. Using unigram and bigram LMs with 2–stage smoothing as examples, we show that our method leads to a bigram LM that outperforms significantly its unigram counterpart and the well–tuned BM25 model.

1. INTRODUCTION

Over the last decade, language modeling approaches to IR have shown very promising empirical results on benchmark datasets. On the one hand, comparing to traditional retrieval models, such as VSM and BM25, the LM approach provides a more principled framework to model various kinds of information useful for document retrieval. For example, the smoothing methods, used to deal with the sparseness in LM estimation, play a similar role to that of the document length normalization and the IDF in traditional retrieval models. On the other hand, LMs present particular modeling challenges. Among the most critical ones is parameter tuning. These parameters include smoothing factors at different levels, and can affect the retrieval effectiveness significantly. Most of the previous researches optimize the parameters empirically via trial and error in an ad hoc manner, which is sup–optimal and time–consuming, especially for the n–gram models with $n > 1$. This paper presents an alternative, more principled, parameter optimization method, inspired by the work of the linear ranking model [1] and the MRF model [2]. We derive a set of features, each from one component model of the smoothed language model. Then we form a linear ranking model to incorporate these features, where the free parameters (i.e., smoothing factors) in the original language model are cast as weights of these features. We thus formulate parameter tuning of LM approaches to IR as a multi–dimensional optimization problem. In this study, we use unigram and bigram LMs with 2–stage smoothing as examples. The bigram LM has a larger set of tuning parameters than its unigram counterpart [3]. Despite its reported superior performance on TREC collections, the parameter tuning of the model is not very much clarified in the original paper [4]. As we will show in our experiments that the proposed method is easy to follow and leads to a bigram LM that outperforms significantly its unigram counterpart and the well–tuned BM25 model on a large scale real–world dataset.

WWW2010, April 26-30, 2010, Raleigh, North Carolina.

2. PARAMETER OPTIMIZATION

With the intention of decoupling the two different roles of smoothing: improving the accuracy of the estimated document language model, and accommodating the generation of common and non–informative words in the query [5], we have implemented 2–stage smoothing unigram and bigram LMs on a large scale real–world dataset. In the first stage, the document LM is smoothed with a Dirichlet prior, and in the second stage, the smoothed document model is linearly interpolated with a background model trained on the collection [3]. Assuming that query Q is specified as a sequence of k words $(q_1, ..., q_k)$, the proposed 2–stage smoothing unigram LM is defined as follows:

$$P(q_i|D) = (1-\lambda)\frac{f_{q_i,D} + \mu/|V|}{|D| + \mu} + \lambda\frac{f_{q_i,C}}{|C|} \quad (1)$$

where λ and μ are free parameters; $|V|$ is the dictionary size and $|C|$ is the sum of the length of the documents in the background. $f_{q_i,D}$ and $f_{q_i,C}$ show the the frequency of q_i in the document and background, respectively. Inspired by [4], we extend the 2–stage smoothing to bigram LM, which is defined as follows:

$$\begin{aligned} P(q_i|q_{i-1}, D) &= (1-\lambda_1)[(1-\lambda_2)\frac{f_{q_i,D} + \mu_1/|V|}{|D| + \mu_1} \\ &+ \lambda_2\frac{f_{q_{i-1}q_i,D} + \mu_2/|V|^2}{f_{q_{i-1},D} + \mu_2}] \\ &+ \lambda_1[(1-\lambda_3)\frac{f_{q_i,C} + \mu_3/|V|}{|C| + \mu_3} \\ &+ \lambda_3\frac{f_{q_{i-1}q_i,C} + \mu_4/|V|^2}{f_{q_{i-1},C} + \mu_4}] \end{aligned} \quad (2)$$

The main challenge of using the unigram and bigram LMs is how to tune the free parameters. In the unigram LM we have 2 free parameters, while the bigram LM contains 7 free parameters. Hence, it is difficult to obtain the optimal parameter setting by a brute–force method. To tune the parameters we use a linear ranking model [1] , which is a variant of the MRF model. The linear ranking model assumes a set of M features, f_m for $m = 1 \dots M$. Each

BM25	$k_1 = 1.0$, $b = 0.5$
Unigram LM	$\lambda = 0.25$, $\mu = 1700$
Bigram LM	$\lambda_1 = 0.24$, $\lambda_2 = 0.29$, $\lambda_3 = 0.94$, $\mu_1 = 1800$, $\mu_2 = 400$, $\mu_3 = 792$, $\mu_4 = 900$

Table 1: Value of the tuned parameters

	BM25	Unigram LM	Bigram LM
NDCG@1	0.2556	0.2586	0.2630
NDCG@3	0.2852	0.2903	0.2912
NDCG@10	0.3552	0.3591	0.3614

Table 2: NDCG values for each retrieval model

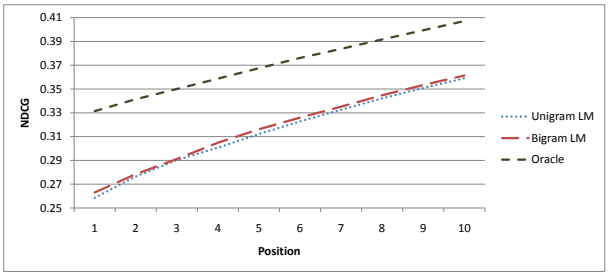

Figure 1: Performance gain of an Oracle ranker.

feature is an arbitrary function that maps (Q, D) to a real value, $f(Q, D) \in R$. The model has M parameters, λ_m for $m = 1 \ldots M$, each for one feature function. The relevance score of a document D of a query Q is calculated as

$$Score(Q|D) = \sum_{i=1}^{M} \lambda_m f_m(Q, D) \quad (3)$$

We consider 2 and 4 features for the unigram and the bigram LMs, respectively. Each feature is derived from a component model in the 2–stage smoothing LM. For example in Equation 1, we consider logarithm of the the coefficient of λ as the first feature and the logarithm of the the coefficient of $(1 - \lambda)$ as the second feature. Similarly we extract 4 features for the bigram LM. Because NDCG is used to measure the quality of the retrieval system in this study, we optimize λs for NDCG directly using the Powell Search algorithm on development data [6].

3. EVALUATION

We evaluated the retrieval models on a large scale real world dataset, containing 11,916 English queries sampled from one–year query log files of a commercial search engine. On average, each query is associated with 185 Web documents (URLs). Each query–document pair has a relevance label. The label is human generated and is on a 5–level relevance scale, 0 to 4, with 4 meaning document D is the most relevant to query Q and 0 meaning d is not relevant to Q. To estimate the value of the free parameters, we randomly split the dataset into training and testing datasets with the same number of queries in each partition. We tuned the parameters on the training dataset and then we used the estimated values to measure performance on the testing dataset. Table 1 shows the estimated values of the free parameters after tuning. For comparison, we also developed BM25 whose free parameters are optimized using the grid search proposed in [7]. The performance of all the retrieval models is measured by mean Normalized Discounted Cumulative Gain (NDCG) [8]. We report NDCG scores at truncation levels 1, 3, and 10 in Table 2.

4. CONCLUSION AND FUTURE WORK

In this paper we discuss the task of fine tuning the free parameters of LMs for IR. In our solution, we first derive features from generative models, and then form a linear ranking model, so that free parameters (smoothing factors) can be optimized with respect to NDCG directly e.g., by using Powell–Search. Using 2–stage smoothing bigram LM as an example, we achieve better results compared to its unigram counterpart, as shown in Table 2. To investigate the impact of word dependencies for LM, we compare the bigram and unigram models via a query–by–query analysis. We find that for about 40% of the queries, the bigram LM works better than the unigram LM. We therefore assume the existence of an oracle ranker. For each query, the ranker can pick the model, either bigram or unigram, which gives the better a NDCG score. Figure 1 shows the NDCG results of such a ranker. As expected, the performance gain is much more significant than that of unigram or bigram LMs. As the first step towards the oracle ranker, we developed a query segmentation model to segment a query into unigrams and bigrams. Notice that for a query, there are multiple segmentations. In our experiments, we only retain the segmentation that gives the maximum query–likelihood probability, based on the product of probabilities of the segments. Results so far have shown some improvement using the query segmentation method. For example, using the method and considering segments with popular bigrams, NDCG scores increase significantly to 0.2675, 0.2979 and 0.3685.

5. REFERENCES

[1] J. G. nd Haoliang Qi, X. Xia, and J.-Y. Nie, "Linear discriminant model for information retrieval," in *SIGIR*, 2005, pp. 290–297.

[2] D. Metzler and W. B. Croft, "A markov random field model for term dependencies," in *SIGIR*, 2005, pp. 472–479.

[3] C. Zhai and J. Lafferty, "Two-stage language models for information retrieval," in *SIGIR*, 2002, pp. 49–56.

[4] J. Gao, J.-Y. Nie, G. Wu, and G. Cao, "Dependence language model for information retrieval," in *SIGIR*, 2004, pp. 170–177.

[5] C. Zhai and J. Lafferty, "A study of smoothing methods for language models applied to information retrieval," *ACM Trans. Inf. Syst.*, vol. 22, no. 2, pp. 179–214, 2004.

[6] M. J. D. Powell, "An efficient method for finding the minimum of a function of several variables without calculating derivatives," *The Computer Journal*, vol. 7, no. 2, pp. 155–162, 1964.

[7] M. Taylor, H. Zaragoza, N. Craswell, S. Robertson, and C. Burges, "Optimisation methods for ranking functions with multiple parameters," in *CIKM*, 2006, pp. 585–593.

[8] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of ir techniques," *ACM Trans. Inf. Syst.*, vol. 20, no. 4, pp. 422–446, 2002.