Multi-Style Language Model for Web Scale Information Retrieval

Kuansan Wang
Microsoft Research
One Microsoft Way
Redmond, WA 98052 USA
kuansan.wang@microsoft.com

Xiaolong Li*
Bing Search Ranking & Intent Group
One Microsoft Way
Redmond, WA 98052 USA
xiaolong.li@microsoft.com

Jianfeng Gao
Microsoft Research
One Microsoft Way
Redmond, WA 98052 USA
jfgao@microsoft.com

ABSTRACT

Web documents are typically associated with many text streams, including the body, the title and the URL that are determined by the authors, and the anchor text or search queries used by others to refer to the documents. Through a systematic large scale analysis on their cross entropy, we show that these text streams appear to be composed in different language styles, and hence warrant respective language models to properly describe their properties. We propose a language modeling approach to Web document retrieval in which each document is characterized by a mixture model with components corresponding to the various text streams associated with the document. Immediate issues for such a mixture model arise as all the text streams are not always present for the documents, and they do not share the same lexicon, making it challenging to properly combine the statistics from the mixture components. To address these issues, we introduce an "open-vocabulary" smoothing technique so that all the component language models have the same cardinality and their scores can simply be linearly combined. To ensure that the approach can cope with Web scale applications, the model training algorithm is designed to require no labeled data and can be fully automated with few heuristics and no empirical parameter tunings. The evaluation on Web document ranking tasks shows that the component language models indeed have varying degrees of capabilities as predicted by the cross-entropy analysis, and the combined mixture model outperforms the state-of-the-art BM25F based system.

Categories and Subject Descriptors

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

General Terms

Theory, Experimentation.

Keywords

Information Retrieval, Mixture Language Models, Smoothing, Parameter Estimation, Probabilistic Relevance Model.

*This work was done when Xiaolong Li was with Microsoft Research.

SIGIR'10, July 19–23, 2010, Geneva, Switzerland.

1. INTRODUCTION

Inspired by the success in speech recognition, Ponte and Croft [23] introduced the language modeling (LM) techniques to information retrieval (IR) that have since become an important research area. The motivation is very simple: just as we would like a speech recognition system to transcribe speech into the most likely uttered texts, we would like an IR system to retrieve documents that have high probabilities of meeting the information needs encoded in the query. Over a decade of studies on this topic, it has been now widely understood that LM is a principled realization of the statistical approach envisioned by Maron and Kuhns at the dawn of IR [19], and that its underlying statistical framework provides mathematically sound explanations to why many proven heuristics, such as TF/IDF weightings and document length normalization, have been working so well [1][6][9][18][31]. As is in the case of speech recognition, LM for IR can be formulated as a Bayesian risk minimization problem [16], for which the optimal performance can be achieved by following the maximum *a posteriori* decision rule that was first shown in [7] and reiterated for IR by Zhai and Lafferty [33]. Specifically, given a query Q, a minimum risk retrieval system should rank the document D based on product of the likelihood of the query under the document language model, $P_D(Q)$, and the prior of the document $P(D)$:

$$s(D,Q) = P_D(Q)P(D) \quad (1)$$

An enthusing question still in the center of the LM for IR research is what the document language model is and how it could be obtained [6][18][31]. While it is intuitive to use the text body to train the document language model as in the majority of the work [18][31], it has been widely recognized that queries are often composed in a different language style than the document body, and a poor query likelihood can thus occur for relevant documents because of the style mismatch. To this end, Miller *et al.* [21] has proposed a hidden Markov model in which an additional latent stage is included to model the query generation process. Lafferty *et al.* have argued for an explicit model of the query language itself [16], and proposed the machine translation techniques to bridge the gap between the body and the query [1]. Jin *et al.* [14], for example, used the title and the body of a document as the target and the source languages, respectively, and demonstrated that the "translated" title LM can be a viable choice as the P_D for IR. The two-stage LM by Zhai and Lafferty [33] proposed yet another idea of using smoothing techniques. There, the document LM is first created by smoothing the document body with a body background model, which is then followed by a second stage of smoothing ideally with a query background model.

In practice, documents often have more fields than just the title and the body. This is particularly true in the Web environment where, in addition to the textual contents created by the document authors, Web documents are also annotated with inbound anchor

text by other document authors, as well as the user queries leading to clicks on the documents. Traditional IR has viewed the multiple-field document retrieval as a structured document retrieval problem, and some established retrieval models, such as BM25, have been generalized to multi-field document retrieval [26]. A straightforward generalization for LM is to view the document as being described by multiple text streams. As shown in Sec. 2, a quantitative analysis on the documents indexed by a commercial Web search engine does confirm that all these text streams seem to be written in their own language styles and have varying degrees of mismatch with the query, justifying the idea to model their linguistic characteristics separately. Empirical studies on applying the LMs for different task domains have also confirmed that mixing textual sources with different language styles can significantly degrade the quality of LMs (e.g., [2]).

When a probability space is divided into disjoint partitions, the probability of an event can be evaluated as the sum of the conditional probability of the event occurring in each partition, weighted by the prior of that partition. Apply this principle to the document modeling and let D_i denote the i^{th} stream of D and P_{Di} the corresponding component LM for the stream, we have

$$P_D = \sum_i P(D_i \mid D) P_{D_i} = \sum_i w_{D_i} P_{D_i} \quad (2)$$

Such a mixture distribution has been widely used for LM in speech and language processing [11] as well as in IR (e.g., [22]). However, beneath the simple linear interpolation form of (2) lies the serious question of the conditions under which the component LMs can be combined properly. Since the foundation of mixture modeling is derived from the probability space partitioning, all the mixture components should therefore be modeling the same underlying probability space. It is widely known [11] that LMs having different sets of vocabulary should be viewed as modeling different domains and therefore their scores cannot be directly compared, let alone combined into a mixture distribution. When applying LM for IR, for example, it is critical to ensure that all document LMs have the same vocabulary so that the document LMs do not selectively treat different portion of the query as out of vocabulary (OOV) terms. The common approach to smooth the document LMs with a shared background model effectively makes all documents use the same vocabulary of the background model. Still, running into OOVs is quite common. For IR using non-mixture LM, encountering OOVs in a query is not a severe problem because the impact in computing the ranking scores (1) is the same for all the documents. This is not true for mixture LM described by (2) since OOVs of one mixture component are not necessarily OOVs for others, making how to properly compute the combined probability of the query a critical question.

To address this problem, we in this work undertake a so-called "open-vocabulary" LM approach that is prevalent in the language processing community (e.g., [11]) but has not been extensively studied for IR. At the core of the open-vocabulary LM is a formidable challenge to assess the probability mass for OOVs. LMs that can yield non-trivial probabilities for OOVs can be viewed as modeling a language with infinite vocabulary. All open-vocabulary LMs thus at least have the same cardinality for their vocabulary, and their probabilistic scores are on a more solid ground to be comparable and combined. As surveyed in Sec. 3, the open vocabulary LM is far from a solved research problem because it inevitably requires one to "guess" the unseen. All the techniques proposed in the past five decades have all involved some kinds of heuristics or parameter tunings that make it challenging to deploy the model outside of research labs. This is because the application domains usually have different environmental conditions that are either not expected by the heuristics or are incongruent to the properties of the tuning data. The scale of the Web typically amplifies the difficulty of these issues, as demonstrated in the machine learning results reported in [28] that show the retrieval performance can be highly volatile depending on how the parameters in BM25F are acquired.

In this paper, we propose an information theoretically motivated method towards open vocabulary LMs. The emphasis here is to obtain an analytically tractable and fully automated system that alleviates the problems arising from heuristic parameter tunings. Typically, such an approach can only yield "statistically optimal" outcome and cannot guarantee the performance be better than fine-tuned systems in all cases. We apply this spirit to both the smoothing of the mixture component LM P_{Di} and the estimation of the mixture weights in (2). In Sec. 4, we present the detailed mathematical derivation that shows how the smoothing parameters can be obtained by computing how N-gram is predicted by (N-1)-gram. In particular, the OOV probability mass, which is equivalent to unigram discount, can therefore be estimated by inspecting how the unigram is predicted by the zero-gram. In Sec. 5 we describe the methods to compute mixture coefficients, and in Sec. 6 we describe the experimental results.

The contributions of the paper are as follows: First, we provide a large scale quantitative analysis to verify how the query language is different in style from document body. We confirm and generalize the prevalent informal observations that, on the Web scale, various fields associated with the documents do have significantly different properties. From a modeling perspective, the analytical outcomes suggest these text sources are better modeled separately. Based on the analysis, we propose a mixture LM approach to IR. The mixture model has to address two immediate and formidable challenges. First, it requires an open-vocabulary LM that has no known solution without heuristics until this work. We propose a mathematically tractable close form solution to realize open-vocabulary LMs. Secondly, the mixture model increases the number of parameters, and we show IR results are very sensitive to tuning. We show that our proposed analytical method can achieve high quality performance without empirical tuning.

2. WEB LANGUAGE STYLE ANALYSIS

The observations that the query language is different in styles from document body and may be closer to titles are intuitive. To formalize the analysis, we conduct a large scale analysis on a June 2009 snapshot of the Web documents in the EN-US market. We examine the language usages in the document text body, the title, the anchor text, as well as the queries against a commercial search engine at the same time. To quantify the language usages in these streams, we first build a statistical N-gram LM for each of them and study the complexity of the language using information theoretic measurements such as entropy or cross-entropy. The LMs used in this section, with the exception of query LMs, are all publicly accessible through [20]. Formally, the cross-entropy between model P_A and P_B is

$$H(P_A \| P_B) = -\sum_t P_A(t) \log P_B(t)$$

Since the logarithmic function is convex, it can be easily shown that the cross entropy is smallest when the two models are identical. The cross entropy function can therefore be viewed as a measurement that quantifies how different the two models are. The entropy of a model P_A, $H(P_A) = H(P_A \| P_A)$. To avoid the confusion on the base of the logarithm, we further convert the

entropy into the corresponding linear scale perplexity measurement, namely,

$$PPL(P_A \| P_B) = e^{H(P_A \| P_B)}$$

Previously, it has been estimated that the trigram perplexity of general English has an upper bound of 247 words per position based on a 1 million word corpus of American English of varying topics and genres [2].

In contrast, in the Web snapshot the vocabulary size is at least 1.2 billion for the document body, and 60 million, 150 million, and 252 million for the title, anchor text and the user query streams, respectively. As our main objective is to investigate how these language sources can be used to model user queries, we study the cross-entropy between the query LM to others, the results of which are shown in Figure 1.

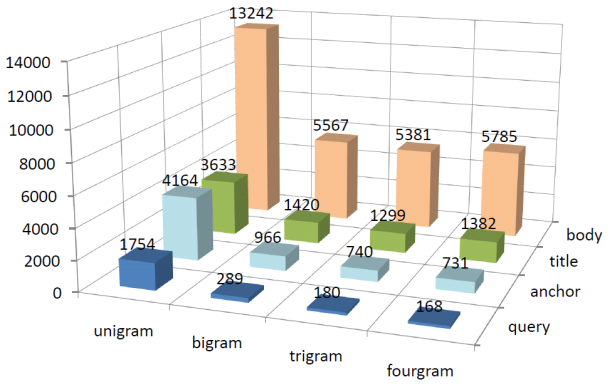

Figure 1: Cross-stream perplexities against queries for various streams and N-gram (N=1, 2, 3, 4)

As can be seen, when the LM grows more powerful with increasing order, the query language perplexity keeps dropping, from 1754 for unigram down to 180 for trigram and 168 for 4-gram. It thus appears that the query language falls within the previously estimated upper bound for perplexity, 247, for general English.

The cross-stream perplexities give hints on the efficacy of using various streams to model the query language. The document body has shown consistently the largest mismatch with the queries, while anchor text seems the best choice among the three to model the queries with powerful enough LM (i.e., N > 1). We note that, starting at bigram, both title and anchor text models have a smaller perplexity than the unigram model of query itself. The study lends some support to the hypothesis that document title is a better source than the body to build a LM for IR.

Up to trigram, the heightened modeling power with an increasing order uniformly improves the perplexities of all streams for queries, although this increased capability can also enhance the style mismatch that eventually leads to the perplexity increase at higher order. For the document body and title, the payoff of using more powerful LMs seems to taper off at bigram, whereas trigram may still be worthwhile for the anchor text.

As are in many applications of LMs, the perplexity measure is not the ultimate metric for applications, in other words, models with lower perplexities do not necessarily lead to a better performance. However, the perplexity analysis is still informative in that higher perplexity models can seldom outperform the lower perplexity ones.

3. CURRENT STATE OF OPEN VOCABULARY LANGUAGE MODEL

For any unigram LM P_C with vocabulary V, the probabilities of all the in-vocabulary and OOV tokens sum up to 1. An open-vocabulary LM is a model that reserves non-zero probability mass for OOVs:

$$pUnk \equiv \sum_{t \notin V} P_C(t) = 1 - \sum_{t \in V} P_C(t) > 0$$

When an open vocabulary model is used to evaluate a text corpus and encounter additional k distinct OOV tokens, the maximum entropy principle [13] is often applied to evenly distribute $pUnk$ among these newly discovered OOVs, i.e., $P_C(t) = pUnk / k$ for $t \notin V$. The key question is how much mass one should take away from V and assign it for $pUnk$. The "discount" strategy, as is often called, remains an unsolved research problem since Shannon invented N-gram as part of the information theory.

Since its publication in 1953, the Good-Turing formula is still a widely used or served as the foundation for many discounting strategies [11]. It states that, if there are n_r tokens that appear exactly r times in a corpus, then for the purpose of calculating probability we should "pretend" these tokens appear r^* times where

$$r^* = (r+1)\frac{n_{r+1}}{n_r}$$

Accordingly, the probability of encountering such a token is given by

$$P_r = \frac{r^*}{\sum_{r=0}^{\infty} r n_r} = \frac{r^*}{|T|}$$

where $|T|$ denotes the total number of tokens in the corpus. By applying the Good-Turing formula for $r = 0$, we have the total probability mass that should be reserved for all the unseen tokens is

$$pUnk = \frac{n_0 \cdot 0^*}{|T|} = \frac{n_1}{|T|}$$

Namely, the probability mass for the unseen is equal to that of the single occurrence tokens. Obviously, how good this discounting strategy is depends heavily on how accurate the Good-Turing formula characterizes the statistical properties of the application in question. Although the Good-Turing has been shown to be useful and superior to many other heuristics for a wide range of applications, the formula is still seen as enigmatic and finding an intuitive explanation to its underlying heuristics remains an active research question [23].

4. COMPONENT MODEL SMOOTHING USING CALM

In this paper, we adopt a more analytically tractable approach to open-vocabulary discount. The key element in our method is a model adaptation algorithm called CALM first proposed by Wang and Li [29]. A close examination of the original presentation reveals that the adaptation framework in CALM can be explained in an alternative manner using the widely known vector space paradigm. As the original CALM was developed for N-gram, we try to keep the discussion in this section general even though we only report experimental data for unigram (N = 1) in this paper.

4.1 Adaptation as Vector Interpolation

First, we note that a LM can be thought of as a vector from an underlying functional space in which all the admissible language models for a given lexicon V form a simplex of the space, namely, $\Lambda = \{P : V \to [0,1], \sum_{v \in V} P(v) = 1\}$. For example, any LM for a trivial binary lexicon can be represented by a point within the line segment enclosed by (1, 0) and (0, 1) on a two dimensional Euclidean space as illustrated in Figure 2. Let P_B denote the background LM and P_O the statistics of a set of newly observed data we would like to adapt the background LM to, respectively. The goal of adaptation is to find a target $P_T \in \Lambda$, $P_T = P_B + \Delta P$, such that P_T and P_O are reasonably close and ΔP, the modification on the background LM, is minimized. Because the resultant model P_T has to reside on the simplex, one cannot simply use the Euclidean norm to compute distances and determine the adapted LM without constraints. However, it can be easily verified that such a constraint can be met if we choose the adjustment vector ΔP along the direction of the difference vector of $P_O - P_B$, namely, $\Delta P = \alpha_O (P_O - P_B)$ where α_O is the adaptation coefficient. Naturally, we want to pick a non-negative α_O so as to point the adjustment towards the right direction, and to choose $\alpha_O < 1$ so as to avoid overshoot.

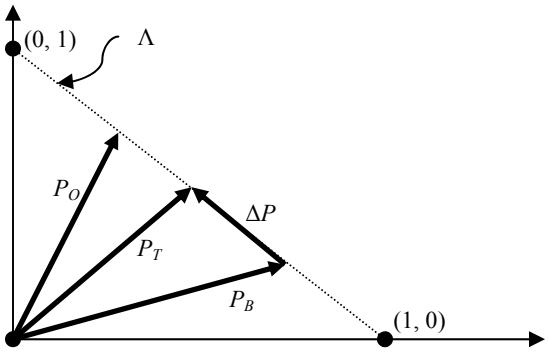

Figure 2: A 2-dimesional illustration of model adaptation in the probability space

Putting it together, we have

$$P_T = P_B + \alpha_O (P_O - P_B) = \alpha_O P_O + (1 - \alpha_O) P_B \quad (3)$$

LM adaptation can therefore be achieved by linear interpolation, assuming the same mathematical form as smoothing.

A significant contribution of CALM is to derive how the adaptation coefficient can be calculated mathematically when the underlying LM is based on N-gram assuming a multinomial distribution. Following the work of [29], $1 - \alpha_O$ can be interpreted as the *prior* probability of P_B being the correct model and whose closed form formulation can be obtained using Stirling's approximation. In the Appendix we show that the adaptation coefficient for a given set of observations O can be computed as

$$\log(1 - \alpha_O) = \sum_{t \in O} \frac{n(t)}{L_O} \log \frac{P_B(t)}{n(t)/L_O} = -KL(P_O \| P_B) \quad (4)$$

where L_O and $n(t)$ denote the document length and the term frequency of the term t, respectively. In short, the adaptation coefficient has a closed form relationship to the Kullback-Leibler (KL) divergence between the background model and the ML estimation of the LM of the document $P_O(t) = n(t) / L_O$. It can be verified that, if $P_B(t)$ agrees with $n(t) / L_O$ for all terms, the adaptation coefficient α_O is 0 indeed. The more the background model disagrees with the observation, the more negative the right hand side of (4) will become, which leads α_O to approach 1.

4.2 Open Vocabulary LM through N-gram Discount

The CALM interpolation formula of (3) indicates that in the target LM P_T, only α_O portion of the probability comes from the observation. In other words, the observation is "discounted" because a probability mass of $1 - \alpha_O$ in the target LM is set aside for sources external to the observation. One can therefore use (4) to compute the discount factor of an N-gram by choosing the corresponding (N-1)-gram as the background. For N > 1, (4) coincides with the formulation of the well-known Stolcke heuristics [27] that has been widely used in the N-gram LM pruning: N-grams that can be reasonably predicted by (N-1)-gram can be pruned out of the model. For the purpose of this work, we further extend the idea down to N = 1, where the observation P_O and the background P_B are the unigram and zero-gram LMs, respectively. Conventionally, the zero-gram LM refers to the least informative LM that treats every token as OOV, namely, its probability mass is exclusively allocated for OOV. Given an observation with a vocabulary size $|V_O|$, a zero-gram LM would just equally distribute its probability mass equally among the vocabulary, leading (4) into

$$\log(1 - \alpha_O) = \sum \frac{n(t)}{L_O} \log \frac{1/|V_O|}{n(t)/L_O} = H(P_O) - \log |V_O| \quad (5)$$

As in Sec. 2, $H(P_O)$ here denotes the (empirical) entropy of the observation LM P_O. We can further convert (5) from the logarithmic into the linear scale and express the discount factor in terms of perplexity and vocabulary size:

$$pUnk = 1 - \alpha_O = PPL(P_O) / |V_O| \quad (6)$$

The interpretation of this outcome is quite intuitive. As well understood, perplexity is the expected number of alternatives when a language model is used to generate a token each time. The ratio of the perplexity to the vocabulary size characterizes how equivocal the language model is. The result of (6) suggests that the higher the ratio, the less certain the language model is and hence the larger the discount should be. At the extreme case when the perplexity equals the vocabulary size, the language model is basically generating tokens in the random pattern as the zero-gram, and hence the discount factor becomes 1.

4.3 Open Vocabulary Component LM

In this paper, we compose the smoothed stream component LM P_{Di} with (3), using an open-vocabulary LM trained from the stream collection as the background model. To be more specific, we first for each document D and each stream i create a closed-vocabulary maximum likelihood model as the observation $P_{O,Di}$. The vocabulary for the stream $V_{O,Ci}$ and the closed-vocabulary stream collection model is thus obtained as

$$P_{O,C_i} = \sum_D P(D) P_{O,D_i}$$

The discount factor is computed with (6) and is used to attenuate the in-vocabulary probability as

$$P_{T,C_i}(t) = \alpha_{O,C_i} P_{O,C_i}(t), \quad t \in V_{O,C_i}$$

The $P_{T,Ci}$ is the open-vocabulary stream collection model. Finally, the stream collection model is used as the background to obtain the smoothed document stream model through linear interpolation

$$P_{D_i} = \alpha_{D_i} P_{O,D_i} + (1-\alpha_{D_i}) P_{T,C_i} \quad (7)$$

Here, the smoothing with the stream collection model ensures each document LM has the same number of mixture components even though the document does not have some stream observations. This smoothing alleviates the dilemma that some streams are sporadic and sparse for many Web documents.

Although the interpolation coefficient α_{Di} in (7) can in practice be kept as a free parameter to be empirically tuned (e.g., [32]), a major objective of this work is to explore alternatives that are tuning-free and thus more desirable when an IR system leaves a lab environment. In addition to the methods described in the next section, we note that the interpolation coefficient in (7) can also be computed using (4) in a document dependent yet query independent fashion. Several observations can be made from this approach. First, the adaptation coefficient of (4) is document dependent as desired. Unlike the Dirichlet smoothing used in [32] that can also yield document dependent estimation of α_{Di}, CALM achieves this without having to make a strong assumption that the family of the prior distributions is conjugate to the multinomial distribution. The estimation is fully automatable in that it does not leave us with a free parameter that can vary and has to be empirically determined from applications to applications. CALM can therefore be implemented at the index time not only in a batch mode but also in an online fashion that model adaptation takes place as soon as the document enters the collection (e.g. crawled from the Web). Secondly, since CALM uses a linear interpolation method, the "IDF effect" pointed out by Zhai and Lafferty [32] to explain why LM performs well for IR as the traditional TF/IDF approach also applies to CALM. Third, we note that the computation of (4) is light weight. Its complexity grows only linearly with the unique terms in the observation.

5. MIXTURE LANGUAGE MODEL

The mixture weights for the component LMs play a central role in the multi-style LM approach. As is in the previous section, we note that we can apply CALM adaptation formula to compute the weights of the multi-component mixture (2) by first re-arranging the distribution as two-component mixture:

$$P_D = w_0 P_{D_0} + (1-w_0) \sum_{i>0} w'_i P_{D_i} \quad (8)$$

As (4) can be applied to obtain w_0, the process can be recursively repeated to obtain other coefficients. Since the goal of the document LM is to evaluate queries, one would like the model to be close to the query language. Accordingly, it seems appropriate to choose the query stream as D_0 in (8) so that the CALM formula functions as adapting other mixture components to the query. This method leads to document-dependent mixture weights, leaves no parameter to tune, and is enticing in terms of engineering Web scale IR because the mixture coefficients can be pre-computed when the documents are being indexed.

The query independent nature of the mixture weights, however, is not as intellectually satisfying as the query dependent ones. While the perplexity studies suggest the average closeness of web document streams to the queries, we observe that the styles of individual queries vary dramatically: As some queries can benefit from large weights on the anchor text or the user query streams, it is not the case for others, especially those whose target documents are new and yet to be widely linked to or sought after with search engines. Indeed, our pilot studies suggest that query dependent weights outperform query independent ones and thus the latter results are omitted in this paper.

The query dependent portion of the ranking function is the query likelihood $P_D(Q)$ in (2). The objective of choosing the optimal mixture weights is to maximize this likelihood. As shown in (7), each mixture component itself is a two-component mixture that has parameters to be determined. We can obtain the re-estimation formula under Expectation Maximization (EM) algorithm as

$$w'_i = \frac{1}{|Q|} \sum_{q \in Q} \frac{w_i P_{Di}(q)}{\sum_i w_i P_{Di}(q)} \quad (9)$$

and

$$\alpha'_{D_i} = \frac{1}{|Q|} \sum_{q \in Q} \frac{\alpha_{D_i} P_{O,D_i}(q)}{P_{Di}(q)} \quad (10)$$

6. WEB SEARCH EXPERIMENTS

To assess the effectiveness of the proposed tuning-free methods for the web document retrieval, we conduct the experiments on the same query set generating the Web test collection previously described by Svore and Burges [28]. The collection consists of 11,845 distinct queries and a retrieval base of more than 1.2 million documents with 5-scale relevance judgments that can be used to compute NDCG as the metric for the retrieval function. The data set has the following five streams associated with each web document: document text body (B), title (T), URL (U), anchor text (A), and the user queries (C) that have one or more clicks on the document recorded in the search engine logs. The percentages of documents with non-empty streams are as shown in Table 1.

Table 1: Portions of documents in the test collection with non-empty text body (B), title (T), URL (U), anchor text (A), and user queries (C)

B	T	U	A	C
81.46%	74.21%	74.21%	75.79%	37.76%

Documents that contain only graphic contents, for example, will be regarded as having empty text body. Since the user query stream is sparsely populated, we exclude it from the study in this paper.

The test collection, having been studied by multiple institutions, comes with a few well-established retrieval results. In the following, we report two pertinent experimental data as baselines for comparison. The first is based on Okapi BM25 [25] and its multi-field extension [26] (referred to as BM25F below), both of which parameters are taken from the published results in [28]. We note that neither set of the Okapi experiment takes into account the document prior, which we have found to be critical in downplaying the roles of undesirable contents such as spam. Similar observations on the importance of document prior have been made for other applications [15]. As such, we adopt the machine learning technique described in [28] and train a neural network ranker based system that uses NDCG@10 as the objective function to combine BM25/F with the document prior, and its results are reported based on leave-one-out cross validation on the test collection below. The results, shown in the "mean% (standard deviation %)" format, are labeled as "Oracle 1" since the machine learning is conducted on the test collection with all the relevance judgments.

There is no reason to believe the results reported here cannot be reproduced elsewhere, such as the recent TREC Web Track data set [4], provided that the document prior can be computed with methods that effectively confront the prevalent spamming activi-

ties on the Web. Specifically, the test collection used in this paper includes a technique described in [30] that identifies spammers based on the HTTP redirection patterns. We have found such crawling time features critical and can augment other link graph analysis and content based methods and lead to an effective prior estimation that makes the IR metrics more meaningful.

6.1 Single Style LM

We first conduct a series of single style LM experiments to understand the merits of the adaptive LM (Sec. 4) against the well-known Dirichlet smoothing based LM. To be precise, the single style LM here means that the document is represented by a single stream.

The rationale for the experimental design is as follows. Aside from the open-vocabulary, which does not play a role for single style LM in the IR tasks, the "tuning-free" method uses the same form, i.e., linear interpolation, to smooth the LM. The novelty here is in the manner of how the interpolation coefficients, which can be interpreted as the prior of the distributions to be interpolated (Sec. 4.1), are chosen. The Dirichlet approach makes the assumption of conjugate prior, which is only contingent upon the distribution family of the LM and not on the empirical observations. Accordingly, the Dirichlet smoothing leaves a free parameter that has to be empirically tuned based on the application data. In contrast, the adaptive LM makes no assumption on the distribution family of the prior. Rather, it capitalizes on the data observed in the document to derive an analytical yet data-driven estimate of the prior, thereby achieving the objective of no free parameters.

Table 2: NDCG for single stream retrieval experiments

		NDCG@1	NDCG@3	NDCG@10
Baseline: BM25	B	26.72	30.19	37.77
	T	26.46	29.64	36.24
	U	29.77	31.68	37.40
	A	33.59	35.90	41.78
CALM	B	28.74	32.02	39.09
	T	33.95	36.33	42.19
	U	36.81	38.06	43.19
	A	35.42	37.50	43.03
EM	B	28.87	32.23	39.30
	T	33.85	36.41	42.52
	U	36.80	38.03	43.09
	A	36.13	38.44	44.20
Oracle 1: BM25 + *P*(*D*) w/ML	B	27.87 (0.58)	30.98 (0.55)	38.48 (0.40)
	T	30.45 (0.18)	33.59 (0.21)	40.44 (0.28)
	U	34.66 (0.14)	35.98 (0.08)	41.99 (0.18)
	A	37.37 (0.25)	38.76 (0.30)	44.14 (0.36)
Oracle 2: Grid-search Dirichlet Smoothing	B	29.37 (0.18)	32.47 (0.11)	39.43 (0.17)
	T	32.05 (0.58)	34.84 (0.54)	41.38 (0.41)
	U	33.67 (0.96)	35.54 (0.71)	41.46 (0.46)
	A	37.62 (0.46)	39.26 (0.30)	44.56 (0.16)

Table 2 shows the experimental results with an emphasis to understand the parameter tuning effects. The experimental condition labeled "CALM" implements (7) for smoothing, whereas the experiments labeled "EM" utilize the EM algorithm to find the interpolation coefficient that maximizes the query likelihood for each individual query. As previously described, CALM is an approach where all the parameters can be computed at the document index time, while EM has to be carried out in retrieval time. Even though parameters maximizing query likelihood do not necessarily improve NDCG scores, it appears to be the case between the CALM and the EM cases. We note that, even though the CALM method does not further utilize query specific information for smoothing, its performance has already come close to the "EM" method. Both LM approaches record higher NDCG scores than the baseline (all results are statistically significant based on *t*-test with significance level of 0.05), and come to the high-end performance of the Oracle 1 that utilizes more data to train the parameters. The closeness to the Oracle 1 result is surprisingly encouraging because all the LM methods optimize only the indirect measures of query likelihood, whereas in all Oracle 1 cases NDCG@10 is directly optimized on the test collection.

To further understand the parameter tuning, we run a grid search on the free parameter in Dirichlet smoothing (from 50 to 500 with a step size 50) and tabulate the corresponding retrieval results in Table 2 labeled as "Oracle 2" in the "mean% (standard deviation%)" format. The results confirm that the choices of free parameters can introduce significant variances in NDCG, and that the EM method can produce reasonable results without tuning.

In all cases, the retrieval experiments lend support to the analysis in Sec. 2 that streams other than the text body tend to be a better choice for IR tasks, and their relative efficacy seems to track the perplexity prediction well. For example, anchor text is consistently outperforms the title and the body streams across all experimental conditions.

6.2 Multi-Style LM

Table 3 summarizes that experiments that test to what extent multiple streams can be combined to improve retrieval performance. The "CALM + EM" condition uses the interpolation coefficients for individual streams determined at the index time in the same manner as described in Sec. 6.1, and uses the EM algorithm to compute the mixture weights at the retrieval time when the query is received. In comparison, the "Joint EM" condition uses the EM algorithm to jointly determine the mixture weights and the interpolation coefficients for all the stream at the retrieval time in the manner described in (10) of Sec. 5. As is in the case for the single style LM, the total retrieval time approach "Joint EM" seems to offer consistent better performance than the partially index time method CALM+EM. Both LM methods produce reasonable performance, even though they do not utilize any judgment data and only are indirectly optimized for query likelihood rather directly on NDCG. Regardless the modeling techniques, all experimental conditions consistently show that better retrieval performance can be achieved when more streams are included in the retrieval model.

The motivation behind the emphasis on "tuning free" is based on our empirical observation that many retrieval methods typically yield dramatically unstable performance, the root cause of which can be traced to their sensitivity to the free parameters in the models. As mixture models increase the number of model parameters, the robustness issue is inevitably exacerbated. We demonstrate the sensitivity issues by including an experimental condition "Oracle 0" in which we retrain the neural net on the test collection to obtain the optimal BM25F parameters. As can be seen, the NDCG metrics change dramatically from the baseline where the BM25F parameters were trained on a separate dataset that is created using the same pooling methodology and judgments guidelines but with the collection harvested from the Web 2 months earlier. More

troublingly, such a dramatic swing in performance metric cannot be discovered through cross validation, as the standard deviations in both Oracle 0 and Oracle 1 appear small. Our investigation confirms that single style stream experiments do not exhibit such a big gap in BM25 performance. This leads to our working hypothesis that the combinations of multiple text streams introduce the new performance robustness challenges, a topic that warrants more research in the future.

Table 3: NDCG for mixture LM for retrieval

		NDCG@1	NDCG@3	NDCG@10
Baseline: BM25F	BT	26.36	32.29	39.36
	BTU	32.15	35.12	42.23
	BTUA	36.02	38.34	45.05
CALM + EM	BT	33.90	36.42	42.77
	BTU	34.90	37.69	44.16
	BTUA	36.46	39.39	45.77
Joint EM	BT	33.92	36.57	42.86
	BTU	35.11	37.82	44.29
	BTUA	36.98	39.71	46.04
Oracle 0: BM25F ML retuned	BT	30.27 (0.13)	33.48 (0.10)	40.75 (0.16)
	BTU	34.85 (0.62)	37.07 (0.49)	43.67 (0.42)
	BTUA	43.84 (0.17)	43.47 (0.05)	48.21 (0.13)
Oracle 1: BM25F + *P*(*D*)	BT	33.01 (0.11)	35.91 (0.30)	42.74 (0.37)
	BTU	34.85 (0.62)	37.07 (0.49)	44.82 (1.03)
	BTUA	45.21 (0.28)	44.51 (0.36)	48.92 (0.52)

7. SUMMARY

The key question of using LM for IR is how to create a LM for each document that best models the queries used to retrieve the document. Studying the textual resources with the documents, we first present convincing and quantitative evidence that different language styles are used for composing the document body, title, anchor text, and queries. As such, these different styles are better separately modeled and then combined to form the document language model.

The immediate question is how LMs with different vocabulary sets can be combined in a principled way. Previous attempts to this so-called open-vocabulary LM problem resorts to heuristics many of which are hard to verify. The most famous and widely used, the Good-Turing formula, is recognized as enigmatic and unintuitive. We propose an alternative based on rigorous mathematical derivations with few assumptions. The same mathematical framework, based on LM adaptation, also suggests that once the open-vocabulary issue is resolved the model combination can be achieved by simple linear interpolation. Such a simple form allows us to employ the EM algorithm to dynamically compute the query-document matching scores without tuning free parameters. Our experiments show that the proposed approach can produce retrieval performance close to the high-end oracle results.

8. ACKNOWLEDGMENT

The authors would like to thank Chris Thrasher, Paul Hsu, Evelyne Viegas, Fritz Behr, and Zijian Zheng for the collaboration.

9. APPENDIX

The linear interpolation of (3) indicates the adapted distribution P_T is a mixture of the ML estimation of the observation data P_O and the background model P_B. Note that the probability of an event E is the mixture sum of the event taking place under various conditions C_i weighted by the respective priors $P(C_i)$:

$$P(E) = \sum_i P(E \mid C_i) P(C_i)$$

We can view the mixture coefficient in (1) as the prior probability of the respective mixture component being the "real" distribution in describing the probabilistic events whose statistical property is characterized by P_T. In the case of adaptation, the probability of the background being the real distribution can be estimated by computing how effective the background model predicts the observation where token t occurs $n(t)$ times among a total of L_O tokens, namely,

$$L_O = \sum\nolimits_{t \in O} n(t)$$

With the assumption that the background model P_B being a multinomial distribution, the probability of the observation evaluated against P_B is

$$P_B(O) = \frac{L_O!}{\prod_{t \in O} n(t)!} \prod\nolimits_{t \in O} P_B(t)^{n(t)}$$

Equivalently,

$$\log P_B(O) = \log L_O! - \sum_{t \in O} \log n(t)! + \sum_{t \in O} n(t) \log P_B(t)$$

The factorial terms in the above equation can be approximated by the well-known Stirling formula

$$\log n! \approx n \log n - n$$

Accordingly, we have

$$\begin{aligned} \log P_B(O) &\approx L_O \log L_O - L_O - \sum_{t \in O} [n(t) \log n(t) - n(t)] \\ &\quad + \sum_{t \in O} n(t) \log P_B(t) \\ &= L_O \log L_O + \sum_{t \in O} n(t) \log \frac{P_B(t)}{n(t)} \\ &= \sum_{t \in O} n(t) \log \frac{P_B(t)}{n(t)/L_O} \end{aligned}$$

Note that the mixture weight is the per-token probability whereas $P_B(O)$ above is evaluated over a total of L_O tokens. With the statistical independent assumptions of the tokens in the LM, we have

$$\log P_B(O) = \log(1-\alpha_O)^{L_O} = L_O \log(1-\alpha_O)$$

which leads to (4).

10. REFERENCES

[1] Berger, A. and Lafferty, J. 1999. Information retrieval as statistical translation. In *Proc. SIGIR-99*, 222-229.

[2] Brown, P., della Pietra, S. A., della Pietra, V. J., Lai, J., Mercer, R. L. 1992. An estimate of an upper bound for the entropy of English. *Computational Linguistics*, 18(1), 31-40.

[3] Bulyko, I., Ostendorff, M., Siu, M., Ng, T., Stolcke, A., and Cetin, O. 2007. Web resources for language modeling in conversational speech recognition. *ACM Trans. on Speech and Language Processing*, 5(1), December, 2005, 1-25.

[4] Clark, C. L. A., and Craswell, N. 2009. Report on the TREC 2009 Web Track. In *Proc. TREC 2009*.

[5] Collins-Thompson, K. and Callan, J. 2005. Query expansion using random walk models. In *Proc. CIKM' 05*, Bremen, Germany, 704-711.

[6] Croft, W. B., Metzler, D., and Strohman, T. 2009. *Search Engines: information retrieval in practice*, Addison Wesley.

[7] Duda, R. O., Hart, P. E. 1973. *Pattern Classification and Scene Analysis*, Wiley, New York.

[8] Fang, H., Tao, T., Zhai, C. 2004. A formal study of information retrieval heuristics. In *Proc. SIGIR-04*, 49-56.

[9] Gao, J., Nie, J., Wu, G., Cao, G. 2004. Dependence language model for information retrieval. In *Proc. SIGIR-04*, 170-177.

[10] Hiemstra, D. and Kraaij, W. 2005. 21 language models at TREC: A language modeling approach to the text retrieval conference. In *TREC: Experimental and Evaluation in Information Retrieval*, MIT Press, E. M. Voorhees and D. Harman (eds).

[11] Huang, X. D., Acero, A., and Hon, H.-W. 2001. *Spoken Language Processing*, Prentice Hall PTR, New Jersey.

[12] Huang, J., Gao, J., Miao, J., Li, X., Wang, K., and Behr, F. 2010. Exploring web scale language models for search query processing. In *Proc. WWW 2010.*

[13] Jaynes, E. T. 1957. Information theory and statistical mechanics. In *Physical Review Series II*, American Physical Society, 106(4), 620-630.

[14] Jin, R., Hauptmann, and Zhai, C. 2002. Title language model for information retrieval. In *Proc. SIGIR-02*, 42-48.

[15] Kraaij, W., Westerveld, T., and Hiemstra, D., 2002. The importance of prior probabilities for entry page search. In *Proc. SIGIR'02*, Tampere, Finland, 27-32.

[16] Lafferty, J. and Zhai, C. 2001. Document language models, query models, and risk minimization for information retrieval. In *Proc. SIGIR'01*, New Orleans, LA, 111-119.

[17] Lavrenko, V., and Croft, W. B. 2001. Relevance-based language models. In *Proc. SIGIR'01*, New Orleans, LA, 120-127.

[18] Manning, C., Raghavan, P., and Schutze, H. 2008. *Introduction to information retrieval*, Cambridge University Press.

[19] Maron, M, and Kuhns, J. 1960. On relevance, probabilistic indexing and information retrieval. *Journal of ACM,* 7, 216-244.

[20] Microsoft web n-gram services. http://research.microsoft.com/web-ngram

[21] Miller, D., Leek, T., Schwartz, R. M. 1999. A hidden Markov model information retrieval system. In *Proc. SIGIR-99*, 214-222.

[22] Ogilvie, P. and Callan, J. 2003. Combining document representations for known item search. In *Proc. SIGIR-03*, 143-151.

[23] Orlitsky, A., Santhanam, N. P., and Zhang, J. 2003. Always Good Turing: asymptotically optimal probability estimation. *Science*, 302(5644), 427-431.

[24] Ponte, J., and W. B. Croft. 1998. A language model approach to information retrieval. In *Proc. SIGIR-98*, 275-281.

[25] Robertson, S. E., Walker, S., Sparck-Jones, K. S., Hancock-Beaulieu, M. M., and Gatford, M. 1994. Okapi at TREC-3. In *Proc. the third text retrieval conference (TREC-3)*, D. K. Harman (eds.), NIST special publication 500-225, Gaithersburg, MD, 109-126.

[26] Robertson, S. E., Zaragoza, H., and Taylor, M. 2004. Simple BM25 extension to multiple weighted fields. In *Proc. CIKM-2004*, 42-49.

[27] Stolcke, A. 1998. Entropy-based pruning of backoff language models. In *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, 270-274.

[28] Svore, K. M. and Burges, C. J. C. 2009. A machine learning approach for improved BM25 retrieval. In *Proc. CIKM'09*, Hong Kong, China.

[29] Wang, K. and Li, X. 2009. Efficacy of a constantly adaptive language model technique for web-scale applications. In *Proc. ICASSP-2009*, Taipei, Taiwan, 4733-4736.

[30] Wang, Y.-M., Ma, M., Niu, Y., and Chen, H. 2007. Spam double-funnel: connecting web spammers with advertisers. In *Proc. WWW-2007*, 291-300.

[31] Zhai, C. 2008. Statistical language models for information retrieval: a critical review. *Foundations and Trends in Information Retrieval*, Vol. 2(3), 137-215.

[32] Zhai, C. and Lafferty, J. 2001. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proc. SIGIR'01*, New Orleans, LA, 334-342.

[33] Zhai, C., and Lafferty, J. 2002. Two-stage language models for information retrieval. In *Proc. SIGIR'02*, Tampere, Finland, 49-56.