# An Investigation of Dirichlet Prior Smoothing's Performance Advantage

Mark D. Smucker[1]    James Allan[1]

Submitted to *Information Retrieval* on November 14, 2006.
Revision submitted on December 13, 2007.

Technical Report IR-548, Center for Intelligent Information Retrieval (CIIR),
Department of Computer Science, University of Massachusetts Amherst,
November 2006, Revised December 2007.

## Abstract

In the language modeling approach to information retrieval, Dirichlet prior smoothing frequently outperforms Jelinek-Mercer smoothing. Both Dirichlet prior and Jelinek-Mercer are forms of linear interpolated smoothing. The only difference between them is that Dirichlet prior determines the amount of smoothing based on a document's length. Theory suggests that Dirichlet prior's advantage should be the result of better document model estimation, for Dirichlet prior sensibly smooths longer documents less. In contrast, our hypothesis was that Dirichlet prior's performance advantage comes primarily from a penalization of shorter documents' scores. We conducted two experiments to test our hypothesis. In our first experiment, when we transformed the test collections to have a uniform probability of relevance given document length, $P(Rel|Len)$, Dirichlet prior's performance advantage disappeared. If Dirichlet prior's advantage came from better estimation, it should have retained that advantage even with a uniform $P(Rel|Len)$. In our second experiment, we gave the known $P(Rel|Len)$ as a document prior to the retrieval method. With the document prior, Jelinek-Mercer's performance increased to match Dirichlet prior and Dirichlet prior showed some degradation in performance. These results confirm our hypothesis. While better estimation was formerly a plausible explanation of Dirichlet prior's performance advantage, we now know that Dirichlet prior smoothing's advantage appears to come from its penalization of shorter documents.

Keywords: Smoothing, Dirichlet prior, Jelinek-Mercer, language modeling, document prior, document length normalization.

# 1 Introduction

The language modeling approach to information retrieval (IR) represents documents as generative probabilistic models (Ponte and Croft 1998, Miller et al. 1998, Hiemstra and Kraaij 1998, Berger and Lafferty 1999, Song and Croft 1999). Documents with higher probabilities for query words are preferred over other documents. A document's score is computed to be the probability that it would generate the query. This probability, known as the query likelihood, is the product of each query word given the document's probabilistic model.

The easiest way to estimate a model for a document is to assign a probability to each word appearing in the document equal to the number of times it occurs divided by the number of word occurrences in the document – this is known as maximum likelihood estimation. Words not in the document will be assigned a probability of zero. Zero probabilities are a problem; a document must contain all query words to avoid a score of zero.

To better estimate document models and eliminate zero probabilities, document models are *smoothed* to produce non-zero probabilities for all words. Common smoothing methods mix the document model with the collection model. The collection can be thought of as one large document consisting of all documents concatenated together. Mixing the document model with the collection model will produce a new document model that has some probability for all words. Query words not in the collection are dropped from the query. Smoothing techniques are commonly parameterized to control the amount of mixing between the document and collection model.

Zhai and Lafferty (2001) investigated the use of three types of smoothing in information retrieval. They reported on Jelinek-Mercer, Dirichlet prior, and absolute discounting smoothing methods. They looked at the performance attainable by these methods on nine collections using both short and very long queries. They used the TREC topic's keyword-like title field for short queries and a concatenation of the title, description, and narrative fields for the long queries. Jelinek-Mercer and Dirichlet prior were the better performing methods.

Zhai and Lafferty (2001) found that on the short queries, Dirichlet prior smoothing was the best performing on eight of the nine collections with absolute discounting being the best on one collection. The performance difference on the short queries was large with an average mean average precision (MAP) of 0.256 for Dirichlet prior vs. 0.227 for Jelinek-Mercer across the nine collections. On the long queries, Dirichlet prior (DP) was the best on six collections and Jelinek-Mercer (JM) smoothing was best on the other three, but their average performance was essentially equivalent. On the long queries, the average MAP for DP was 0.279 vs. 0.280 for JM. In a later work, Zhai and Lafferty reported on JM vs. DP performance when the sentence-length description field was used as a query (Zhai and Lafferty 2002). Out of six collections, DP performed better than JM on five with an average MAP of 0.211 compared to JM's average MAP of 0.187. The title and description queries represent query lengths one could realistically expect from a user, and on these lengths Dirichlet prior considerably outperforms Jelinek-Mercer smoothing.

We wanted to understand why Dirichlet prior smoothing performs better than Jelinek-Mercer smoothing. Both Dirichlet prior and Jelinek-Mercer linearly combine the maximum likelihood estimated (MLE) document model with the MLE model of the collection. Both are discounting smoothing methods that reduce the probability of the words seen in the document and reallocate the probability mass to words not seen in the document. The only difference between the two smoothing methods is that Dirichlet prior discounts longer documents less than shorter documents. In other words, Dirichlet prior smooths longer documents less. Jelinek-Mercer smooths all documents to the same degree.

Smoothing longer documents less, and conversely smoothing shorter documents more, should produce better estimated document models. Long documents give much more evidence with which to accurately estimate the probabilities of words. Intuitively, we should not trust the MLE model of a short document as much as the MLE model of a long document.

Zhai and Lafferty (2001) suggest that Dirichlet prior performs better because it produces better estimated document models. Theoretically, better estimated models will result in better document retrieval.

In contrast, we found that Dirichlet prior's performance advantage comes more from its penalization of shorter documents than from its potentially better estimation. In the TREC collections, longer documents are more likely to be relevant (Singhal et al. 1996). It appears that by smoothing shorter documents more than longer documents, Dirichlet prior is able to suppress the scores of short documents, which is advantageous given the nature of the TREC collections.

This is a significant finding for two reasons. First, theory says that Dirichlet prior will perform better than Jelinek-Mercer because of better probability estimates. We found that the Dirichlet prior's better estimation has little to do with its observed performance advantage. Second, this finding points to the importance of incorporating into language modeling retrieval a component that can preferentially weight documents more likely to be relevant given factors such as document length. Such preferential weighting is cleanly incorporated as a prior probability.

This paper extends our earlier work (Smucker and Allan 2005) with additional experiments and analysis.

# 2 Language Model Retrieval and Smoothing

In this section we review the language modeling approach to information retrieval and the Jelinek-Mercer and Dirichlet prior smoothing methods.

## 2.1 Notation

The vocabulary, $V$, is the set of words in the collection. The number of words in $V$ is $|V|$. Documents contain zero or more occurrences of each word in the vocabulary. The count of word $w$ in document $D$ is $D(w)$. The document length is the cardinality of $D$, $|D|$ and is defined as follows:

$$|D| = \sum_{w \in V} D(w)$$

The collection, $C$, is also a multiset over $V$ and $|C|$ is the total number of word occurrences in $C$. Query $Q$ is also represented as a multiset over the vocabulary.

The probabilistic model of document $D$ will be represented as $M_D$. The probability of a word $w$ given a document model $M_D$ is $P(w|M_D)$. For convenience, we write the maximum likelihood estimated probability of a word $w$ given a piece of text $T$ as $P(w|T)$.

## 2.2 Probabilistic Models of Documents

We use the multinomial as our probabilistic model of text. A multinomial model of text specifies a probability for each word in the vocabulary $V$. The probabilities of the multinomial are its parameters and thus there are $|V|$ parameters. The probabilities of the multinomial sum to one. A common way to think about the multinomial is as a biased die. A die has $|V|$ faces with each word having some probability of being *generated* by the die on a roll.

For a given text document, $D$, the parameters of the multinomial representing $D$, $M_D$, need to be determined. This process of computing the probability of a word $w$ given the model $M_D$, $P(w|M_D)$, is called estimation. A standard approach to parameter estimation is maximum likelihood estimation (MLE). MLE maximizes the likelihood of the observed data given the model. Treating the words of $D$ as independent samples, the likelihood of $D$ is defined to be:

$$L(D) = \prod_{w \in D} P(w|M_D)^{D(w)} \tag{1}$$

The maximum likelihood estimate for the probability of a word turns out to be the count of that word divided by the total number of occurrences in $D$:

$$P(w|D) = P(w|M_D) = \frac{D(w)}{|D|} \tag{2}$$

We can create MLE models of any piece of text $T$. As mentioned in the previous section, we will write $P(w|T)$ to represent the MLE probability of $w$ given $T$.

Note that the MLE model has zero probabilities for all words not in the document.

## 2.3 Retrieval Model

Documents are ranked by the probability of a document given a query, which is given by Bayes' theorem as:

$$P(D|Q) = \frac{P(Q|D)P(D)}{P(Q)} \tag{3}$$

We drop $P(Q)$ from the above equation since it is the same for all documents and will not affect the ranking. The prior probability of a document is given by $P(D)$.

The probability that a document model could generate a query $Q$, which is known as query likelihood, is given by:

$$P(Q|M_D) = \prod_{w \in Q} P(w|M_D)^{Q(w)} \tag{4}$$

where $M_D$ is the probabilistic model of $D$. Substituting Equation 4 into Equation 3 (with $P(Q)$ dropped) gives us our scoring function for a document:

$$P(D|Q) \propto P(D) \prod_{w \in Q} P(w|M_D)^{Q(w)} \tag{5}$$

If for some word $w \in Q$, $P(w|M_D) = 0$, then the document will be given a score of zero. This is the zero probability problem. Especially for long queries, it unreasonable to require documents to contain an occurrence of all query words. The zero probabilities of the MLE model of a document are particularly poor estimates since it is reasonable for there to be some probability that any word occurs in a document. We can eliminate the zero probabilities from the document models using smoothing.

## 2.4 Document Smoothing Methods

A solution to the problem of zero probabilities and poor probability estimates is to bring prior knowledge to the estimation process. A natural fit as a prior for the multinomial is the Dirichlet density (Sjölander et al. 1996). A Dirichlet density can be thought of as an urn containing multinomial dies. All the multinomials are of the same size with $|V|$ parameters. The Dirichlet density has the same number of parameters as the multinomials for which it is a prior. The vector $\vec{\alpha}$ represents the parameters of the Dirichlet density. For each word $w$ in the vocabulary, there is a corresponding element $\alpha_w$ of $\vec{\alpha}$, and all $\alpha_w > 0$.

When we use the Dirichlet density as the prior for the multinomial, the estimate of the probability of word given a document is the weighted average of the word's probability in all multinomials. Each multinomial is weighted by its probability given the observed document and the Dirichlet density. This estimate is the mean posterior estimate:

$$P(w|M_D) = \int_M P(w|M) P(M|\vec{\alpha}, D) dM \tag{6}$$

which reduces to:

$$P(w|M_D) = \frac{D(w) + \alpha_w}{|D| + |\vec{\alpha}|} \tag{7}$$

as shown by Sjölander et al. (1996).

The longer the document, the less influence the Dirichlet prior has in determining the parameter estimates for the multinomial $M_D$. The mean of the Dirichlet density is for each $\alpha_w$, $\alpha_w/|\vec{\alpha}|$. As the text becomes shorter, the parameter estimates for $M_D$ regress to the mean of the Dirichlet density.

The parameters of a Dirichlet density can be represented as a multinomial probability distribution $M$ and a weight $m = |\vec{\alpha}|$. Thus, with $P(w|M) = \alpha_w/|\vec{\alpha}|$, Equation 7 becomes:

$$P(w|M_D) = \frac{D(w) + mP(w|M)}{|D| + m} \tag{8}$$

The machine learning community terms this formulation of Dirichlet prior smoothing the *m-estimate* (Mitchell 1997). The parameter $m$ is the *equivalent sample size*. The Dirichlet density when used as a prior for the multinomial can be understood as taking $m$ samples according to $P(w|M)$ prior to observing the data in $D$.

The parameters of the Dirichlet density can be determined using maximum likelihood estimation (MLE). MLE finds the density parameters that produce the highest likelihood for a collection of documents when the density is used as a prior. The MLE can be computed numerically using a Newton-Raphson method (Narayanan 1991) or via an expectation maximization (EM) like method (Sjölander et al. 1996).

In contrast, common practice in information retrieval, and the one we follow, is to let $P(w|M) = P(w|C)$, i.e. use the MLE model of the collection for $M$. This results in the common expression of Dirichlet prior smoothing as:

$$P(w|M_D) = \frac{D(w) + mP(w|C)}{|D| + m} \tag{9}$$

The value of $m$ is a fixed value and is determined empirically. Typically $m$ is set to maximize a retrieval metric like mean average precision for a set of queries and a collection of documents.

A closely related smoothing method is *linear interpolated smoothing.* Linear interpolated smoothing linearly combines two models to produce a smoothed model. Documents are typically smoothed with the collection. The document $D$ is smoothed with the collection $C$ as follows:

$$P(w|M_D) = (1-\lambda)P(w|D) + \lambda P(w|C) \tag{10}$$

The amount of smoothing increases as $\lambda$ increases from 0 to 1, which matches the behavior of an increase in Dirichlet prior's parameter $m$. $P(w|D)$ and $P(w|C)$ are the MLE models of $D$ and $C$ respectively.

Dirichlet prior smoothing is a form of linear interpolated smoothing. For a given document length $|D|$ and parameter $m$, an equivalent $\lambda$ exists. By setting $\lambda$ in Equation 10 to:

$$\lambda = 1 - \frac{|D|}{|D|+m} \tag{11}$$

Equation 9 can be written in the form of Equation 10 (Johnson 1932):

$$\begin{aligned}
P(w|M_D) &= (1 - 1 + \frac{|D|}{|D|+m})P(w|D) + (1 - \frac{|D|}{|D|+m})P(w|C) \\
&= \frac{P(w|D)|D|}{|D|+m} + (\frac{|D|+m}{|D|+m} - \frac{|D|}{|D|+m})P(w|C) \\
&= \frac{D(w)}{|D|+m} + \frac{mP(w|C)}{|D|+m} \\
&= \frac{D(w) + mP(w|C)}{|D|+m}
\end{aligned}$$

Because Dirichlet prior is equivalent to linear interpolated smoothing with a $\lambda$ parameterized on document length, both methods smooth a document exactly the same way for a given $\lambda$.

As document length increases, Dirichlet prior smoothing gives less weight to the collection, $P(w|C)$, and more weight to the document, $P(w|D)$. Longer documents' maximum likelihood estimates of probabilities are trusted more than shorter documents' estimates.

We studied the difference between Dirichlet prior smoothing and linear interpolated smoothing with a fixed $\lambda$. Following Chen and Goodman (1998) and Zhai and Lafferty (2001), we will refer to linear interpolated smoothing with a fixed $\lambda$ as Jelinek-Mercer smoothing.

# 3 Queries and Collections

In this section we describe the queries and collections that we used for our experiments.

We used the TREC 3, 7, and 8 ad-hoc retrieval tasks for our experiments. These tasks respectively consist of topics 151-200, 351-400, and 401-450. Each topic consists of a title, description, and narrative. The titles best approximate a short keyword query while the description is typically formulated as a single well formed sentence describing the information need of the user. The narratives are directions to potential future relevance assessors and are often paragraph length descriptions of what should be considered on-topic and off-topic.

We used only titles and descriptions in isolation of each other to represent queries. For Zhai and Lafferty (2001), the short queries are the titles and the long queries are the concatenation of title, description and narrative fields. We agree with the formulation to use titles as keyword-like non-verbose queries and descriptions as verbose queries (Zhai and Lafferty 2002). A verbose query is likely to contain many more common and non-informative words as opposed to the more focused title queries.

The collection for TREC 3 consists of TREC volumes (discs) 1 and 2. The collection for TREC 7 and 8 consists of TREC volumes 4 and 5 minus the Congressional Record (CR) subcollection. We preprocessed the collections and queries in the same manner. We stemmed using the Krovetz stemmer (Krovetz 1993) and removed stopwords using an in-house stopword list of 418 noise words. We used Lemur 4.3.2 (Lemur 2003) for all experiments. The calculations in Section 5.2 and the data plotted in Figure 6 were created using Lemur 2.0.3, which has minor differences in the way it tokenizes text.

# 4 Experiments

In this section, we detail the experiments we conducted to test our hypothesis that Dirichlet prior's performance advantage comes more from a penalization of short documents' scores than from its potentially better estimation of document models.

We designed two experiments to test our hypothesis. Our approach in the first experiment is to remove the need for a retrieval algorithm to factor in the probability of a document being relevant given the document's length, $P(Rel|Len)$, by making the test collections have a uniform $P(Rel|Len)$. In our second experiment, rather than remove the need to deal with a non-uniform $P(Rel|Len)$, we directly supply this information as part of the retrieval model. We compare each experiment's results to the performance of each smoothing method when there are no changes to the collection or document smoothing.

In each experiment we perform a sweep of the smoothing parameters and find the best performance as measured by mean average precision for Jelinek-Mercer and Dirichlet prior smoothing. The parameters for Dirichlet prior and Jelinek-Mercer smoothing were determined by evaluating the mean average precision for a set of parameter values. For Dirichlet prior, $m$ was tried with values of $\{25, 50, 100, 150, 200, 250, 300, 350, 400, 500, 600, 800, 1000, 1250, 1500, 1750, 2000, 2500, 3000, 5000\}$. For Jelinek-Mercer, $\lambda$ was tried with values of 0.01, 0.05 to 0.95 inclusive in 0.05 increments, and 0.99.

## 4.1 Uniform $P(Rel|Len)$

In this experiment, we transform the TREC collections we used into collections with a uniform probability of relevance given document length, $P(Rel|Len)$, without changing the underlying collection statistics. The result is that a retrieval method has no need to a priori prefer documents of a certain length. If Dirichlet prior's advantage comes from better estimation, it should still produce superior results compared to Jelinek-Mercer.

We obtain a uniform $P(Rel|Len)$ by first binning the documents by length. Starting from a length of zero, we add all the documents of a given length to a bin until there are at least 10,000 documents in the bin. We then create a new bin and repeat the process. If the number of documents in the last bin is less than 10,000, we move those documents to previous bin to avoid having a bin with less than 10,000 documents.

Figure 1 shows the fraction of relevant documents in each bin for the three test collections. In each case, we see that the probability of relevance given document length, $P(Rel|Len)$, is greater for longer documents.

We next compute the probability of relevance, $P(Rel)$, for each test collection as the number of relevant documents in the collection divided by the total number of documents in the collection. We modify each bin by randomly removing or duplicating non-relevant documents to obtain a probability of relevance for each bin equal to the collection's $P(Rel)$. The result of this is that we now have a test collection with a uniform $P(Rel|Len)$.

Removal and duplication of non-relevant documents is symbolic and no ac-

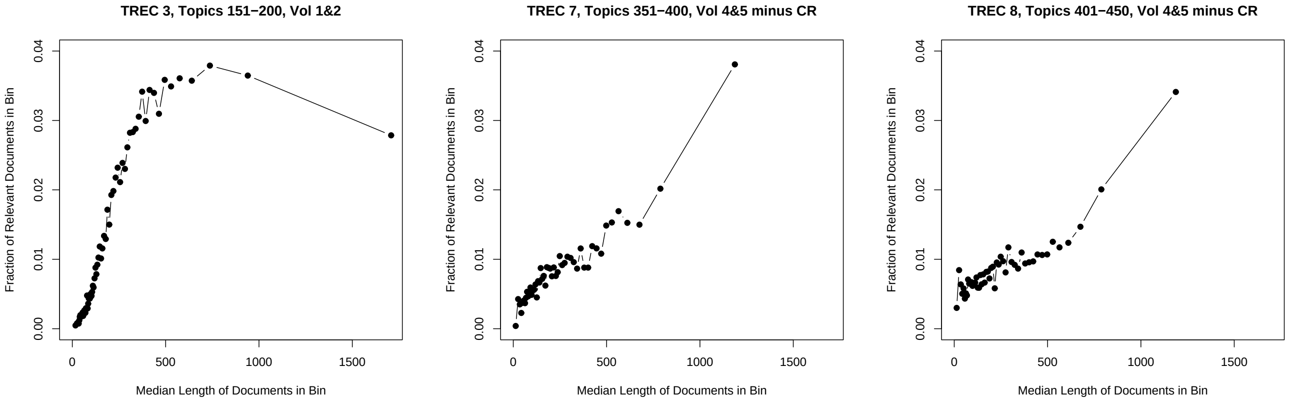

Figure 1: The probability of relevance for documents binned by length for TRECs 3, 7, and 8. Bins have a minimum size of 10,000 documents. TREC 7 and 8 use the same underlying collection but have different sets of relevant documents. For very long documents, TREC 3 also shows a significant increase in probability of relevance, which would have been evident if we had used a smaller bin size.

tual change to the collection occurs. We perform a regular retrieval and then post process the results by removing "deleted" non-relevant documents and repeating "duplicated" documents. As a result, the ordering of the remaining documents is the same as for a regular retrieval.

## 4.2 Document Prior

Rather than remove the need to a priori prefer documents based on length, we can supply the known prior probability of relevance given a document's length as the document prior, $P(D)$, in Equation 5. If Dirichlet prior's performance advantage comes from better estimation and not a bias towards longer or shorter documents, then it should be helped by the addition of this document prior.

To compute the document prior, we use the same binning process as the uniform $P(Rel|Len)$ experiment and for documents in a given bin we use the fraction of relevant documents in that bin as the prior. Thus, Figure 1 shows the document priors.

Our goal is to test our hypothesis that Dirichlet prior's performance advantage comes from a bias toward longer documents. We are not proposing a document prior as a means to improve retrieval performance. If we were testing the document prior as a method to improve retrieval performance, then we would use the document priors from one set of topics as a training set and test on another set of topics. Instead, we purposely use the document priors calculated for a set of topics with that set of topics. With the given document priors, any performance advantage that a smoothing method shows must come from better estimation of document models or other unknown features of the smoothing method.

## 4.3 Possible Experimental Outcomes

For each experiment, the performance of a smoothing method can either degrade, have no change, or improve. For the first experiment where we make a uniform probability of relevance given document length, we can conclude the following about a smoothing method for each of the outcomes:

- Degrade: The smoothing method has a document length bias that aids its performance. When the advantageous non-uniform $P(Rel|Len)$ is replaced with a uniform $P(Rel|Len)$, the smoothing method is unable to adjust. By being "unable to adjust," we mean that there is not a parameter setting that allows it to perform as well.
- No change: Either the smoothing method has a bias but can adjust to the uniform $P(Rel|Len)$ or the method has no bias.
- Improve: The method had a negative document length bias that was degrading its performance. By "negative bias" we mean the method prefers short document when it should prefer long documents or vice versa.

For the second experiment where we provide the known $P(Rel|Len)$ as a document prior:

- Degrade: The smoothing method has an existing bias that helped. When the document prior is added, this positive bias is compounded and the method is unable to adjust.
- No change: The method may have a bias but adjusts or has no bias and is for some reason unable to utilize the document prior.
- Improve: The method lacked a bias or had a negative document length bias.

Our hypothesis will be falsified if the performance of Dirichlet prior improves in either experiment. If Dirichlet prior's performance degrades in either experiment we will have support for our hypothesis. If Dirichlet prior shows no change in an experiment, that outcome neither supports nor falsifies our hypothesis.

## 5 Results and Discussion

Table 1 shows the results for the experiments. Figures 2, 3, and 4 show the performance of the smoothing methods as their smoothing parameter is changed.

As Table 1 shows, when we perform "regular" retrieval with a uniform document prior, we see that Dirichlet prior either beats or ties Jelinek-Mercer. While in all cases, Dirichlet prior obtains a higher mean average precision (MAP), for three of the six cases the smoothing methods produce statistically significant results. Statistical significance is measured by a two-sided, paired randomization test with 100K samples (Smucker et al. 2007). These result confirm the findings of Zhai and Lafferty (2001).

For the experiment where we created a uniform $P(Rel|Len)$, we see that the performance of Jelinek-Mercer shows no change compared to the regular retrieval. On the other hand, Dirichlet prior shows a marked degradation in performance. This experiment supports our hypothesis that Dirichlet prior's performance advantage comes more from a document length bias than from its potentially better estimation. If Dirichlet prior's estimation was the source of its power, then its performance should still be superior to Jelinek-Mercer even when the $P(Rel|Len)$ is uniform.

For the second experiment where we supplied the known $P(Rel|Len)$ as a document prior, we see that Jelinek-Mercer's performance is improved to match that of Dirichlet prior's performance without the document prior. Dirichlet prior's performance shows little to no degradation for title queries but does show some degradation for description queries in comparison to its performance without the prior. While not as strong of an effect as the first experiment, we again see support for our hypothesis. Indeed, the performance gain of Jelinek-Mercer so closely matches that of Dirichlet prior without a document prior that we can possibly think of Dirichlet prior as Jelinek-Mercer smoothing plus a document prior that penalizes shorter documents compared to longer documents.

Taken together these experiments give us evidence that Jelinek-Mercer lacks a document length bias while Dirichlet prior has a bias. The experiments also

Regular Retrieval

| Query | Topics | JM MAP | DP MAP | p-value | JM $\lambda$ | DP $m$ |
|---|---|---|---|---|---|---|
| title | trec 3 | 0.216 | **0.256** | 0.000 | 0.30 | 800 |
| title | trec 7 | 0.168 | **0.189** | 0.005 | 0.55 | 1250 |
| title | trec 8 | **0.236** | **0.249** | 0.130 | 0.25 | 350 |
| desc. | trec 3 | 0.182 | **0.212** | 0.000 | 0.80 | 1750 |
| desc. | trec 7 | **0.171** | **0.183** | 0.259 | 0.90 | 5000 |
| desc. | trec 8 | **0.224** | **0.228** | 0.778 | 0.85 | 2500 |

Uniform $P(Rel|Len)$

| Query | Topics | JM MAP | DP MAP | p-value | JM $\lambda$ | DP $m$ |
|---|---|---|---|---|---|---|
| title | trec 3 | **0.211** | **0.204** | 0.147 | 0.75 | 350 |
| title | trec 7 | **0.172** | **0.163** | 0.214 | 0.85 | 150 |
| title | trec 8 | **0.235** | **0.232** | 0.640 | 0.50 | 150 |
| desc. | trec 3 | **0.184** | 0.147 | 0.001 | 0.95 | 1250 |
| desc. | trec 7 | **0.177** | 0.146 | 0.007 | 0.95 | 1000 |
| desc. | trec 8 | **0.224** | 0.192 | 0.004 | 0.90 | 600 |

With Document Prior

| Query | Topics | JM MAP | DP MAP | p-value | JM $\lambda$ | DP $m$ |
|---|---|---|---|---|---|---|
| title | trec 3 | **0.253** | **0.251** | 0.650 | 0.70 | 350 |
| title | trec 7 | **0.184** | **0.184** | 0.959 | 0.80 | 300 |
| title | trec 8 | **0.244** | **0.249** | 0.313 | 0.40 | 150 |
| desc. | trec 3 | **0.225** | 0.204 | 0.023 | 0.95 | 1250 |
| desc. | trec 7 | **0.188** | **0.171** | 0.070 | 0.90 | 1500 |
| desc. | trec 8 | **0.239** | 0.215 | 0.007 | 0.85 | 1000 |

Table 1: These tables show the non-interpolated mean average precision (MAP) scores for Jelinek-Mercer (JM) and Dirichlet prior (DP) smoothing. The top table is a regular retrieval without any changes to the test collections and uses a uniform document prior, $P(D)$. The middle table shows the results for the experiment where we modified the test collections to obtain a uniform probability of relevance given a document's length, $P(Rel|Len)$. The bottom table shows the results when we set the document prior equal to the known non-uniform $P(Rel|Len)$. Each table also reports, for each pair of results, the p-value as computed by the paired, two-sided randomization test (Smucker et al. 2007). P-values less than 0.05 should be considered statistically significant differences. The best score of a pair of run is shown in bold and both scores are bold if there is not a statistically significant difference.

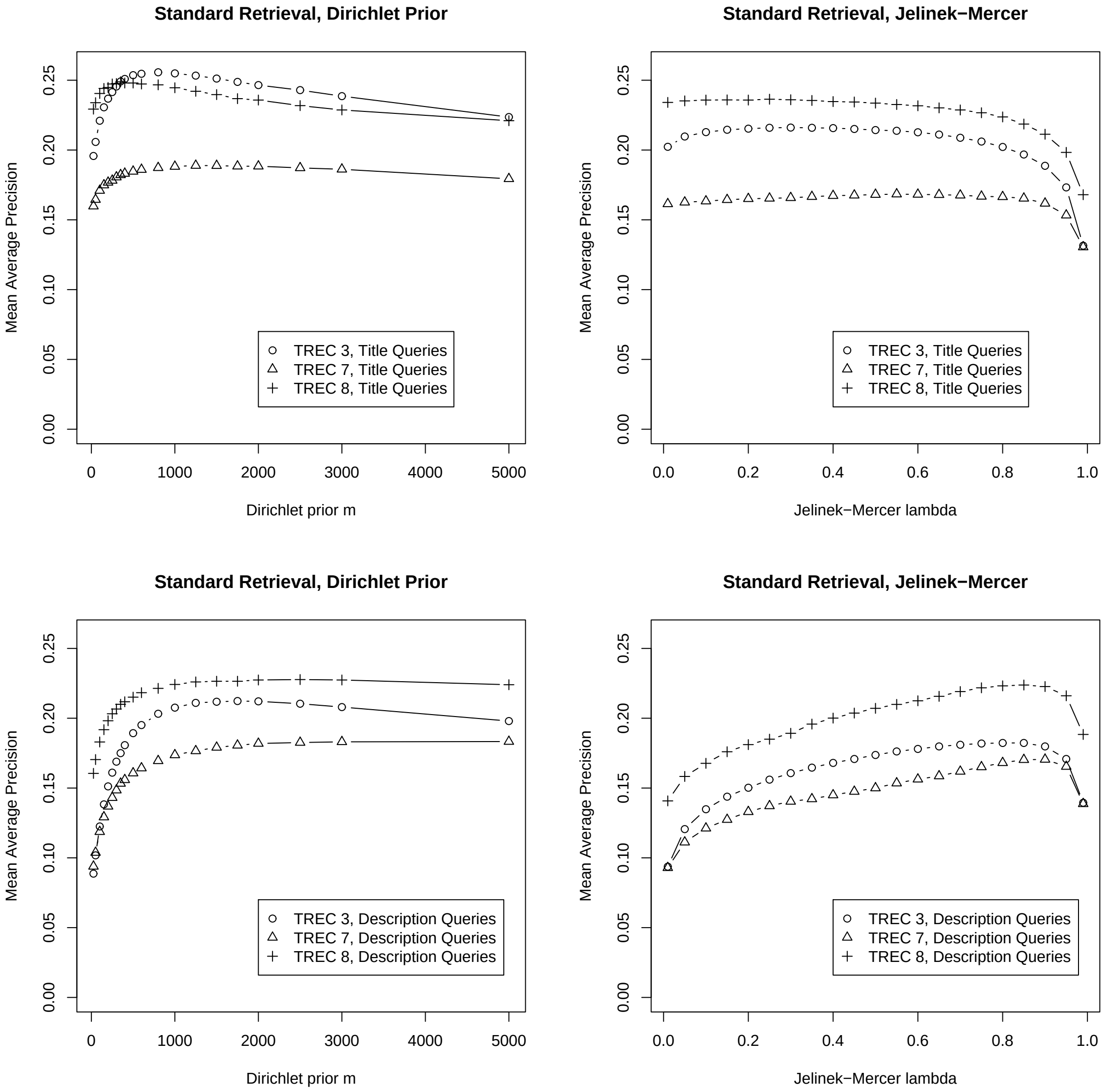

Figure 2: The mean average precision for Dirichlet prior and Jelinek-Mercer smoothing as their respective smoothing parameters are varied. Shown on the top are title queries and on the bottom are description queries.

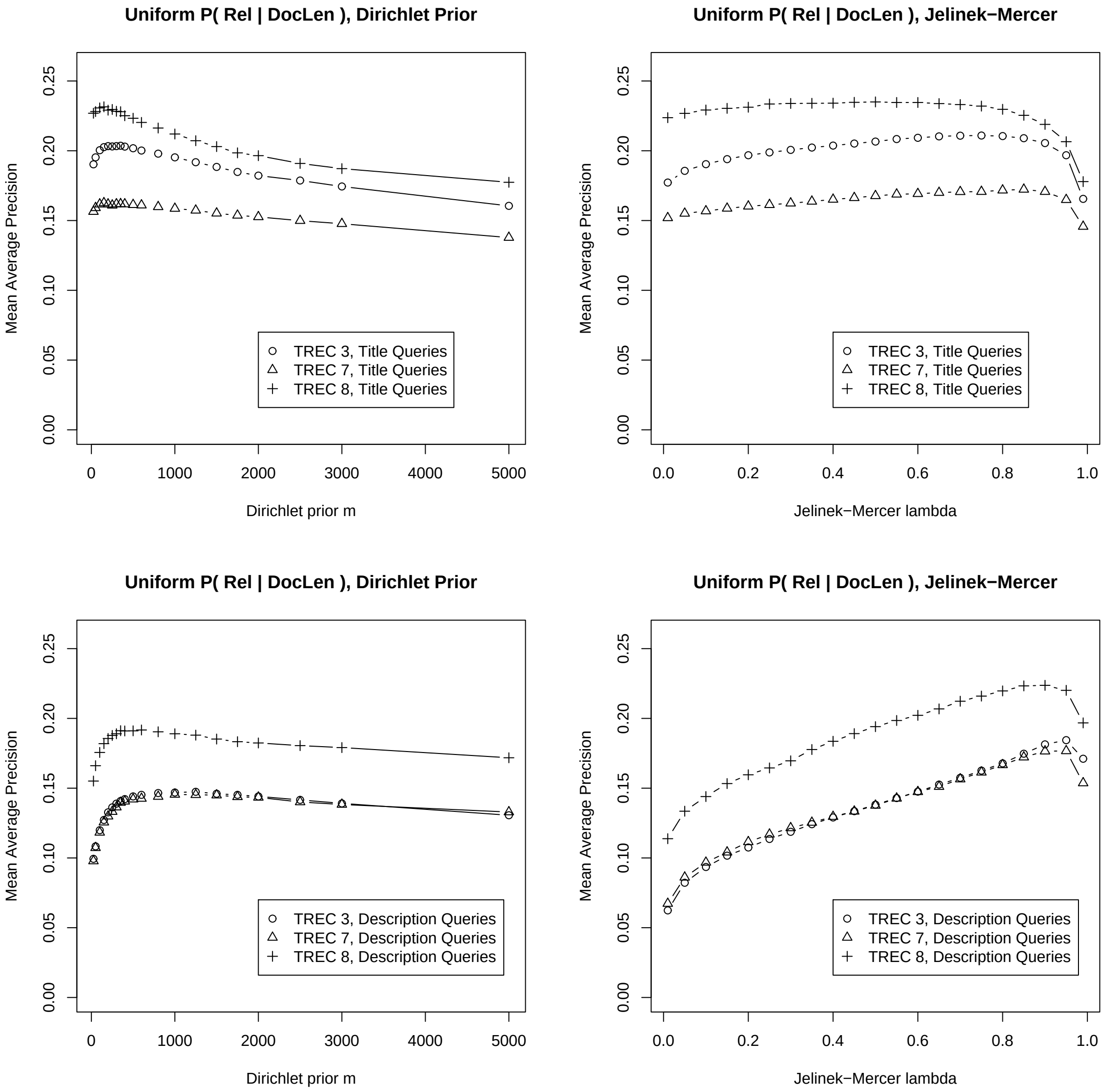

Figure 3: For this experiment, we modified the document collections to obtain a uniform probability of relevance given length, $P(Rel|Len)$. Shown is the mean average precision for Dirichlet prior and Jelinek-Mercer smoothing as their respective smoothing parameters are varied. Shown on the top are title queries and on the bottom are description queries.

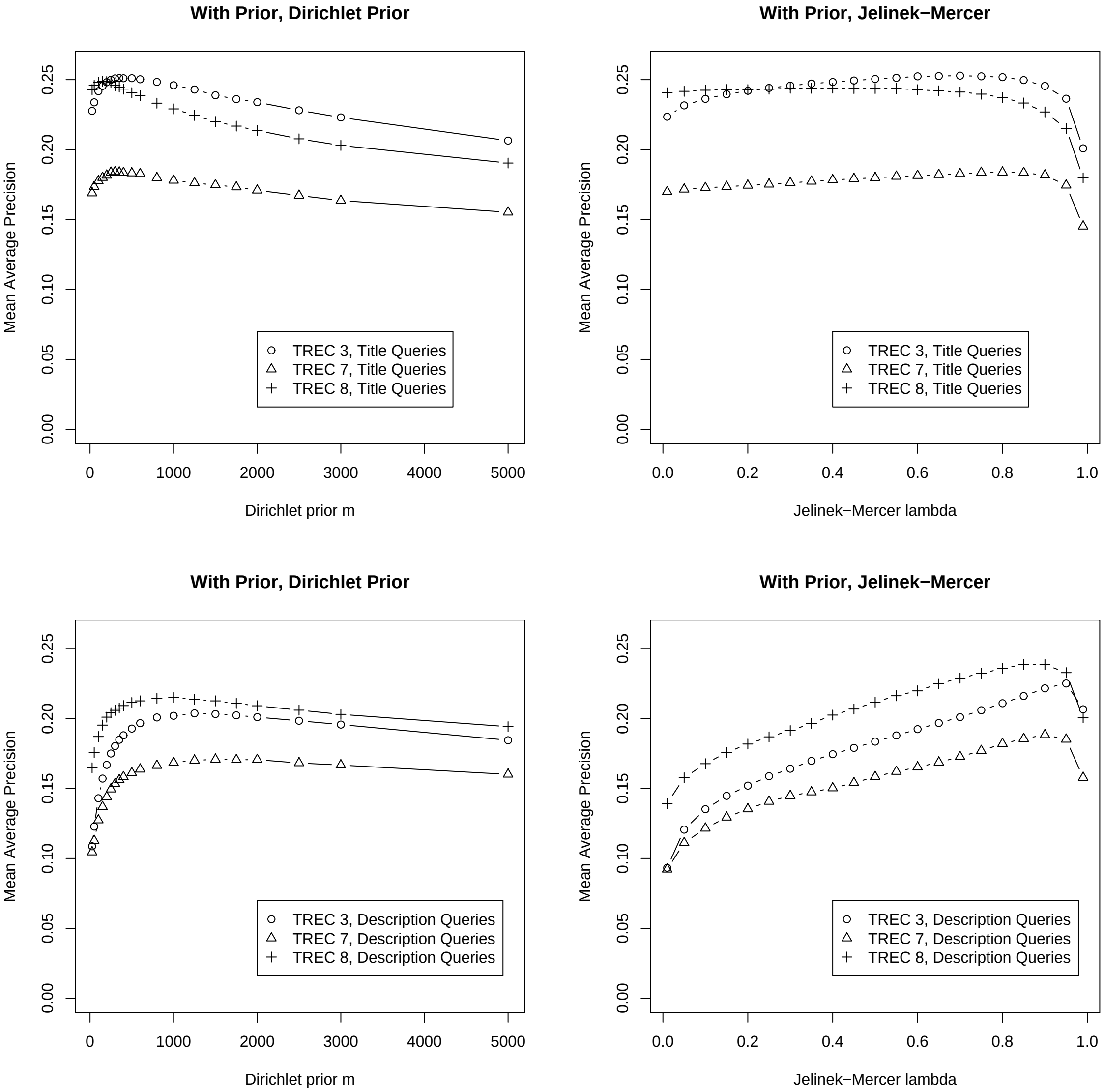

Figure 4: For this experiment, we supplied the known probability of relevance given length, $P(Rel|Len)$ as the document prior, $P(D)$. Shown is the mean average precision for Dirichlet prior and Jelinek-Mercer smoothing as their respective smoothing parameters are varied. Shown on the top are title queries and on the bottom are description queries.

show that Dirichlet prior's document length bias is the reason for its performance advantage rather than Dirichlet prior's potential ability to estimate better document models.

This finding mirrors the experience of Buckley (2005) with the SMART retrieval system. Buckley (2005) writes that adding pivoted document-length normalization (Singhal et al. 1996) to SMART "fundamentally changed every approach we had been taking." Buckley discovered that many techniques that had shown promise actually had no benefit beyond their side effect to prefer longer documents. Besides finding that some techniques had gained their advantage by preferring longer documents, Buckley also found that other techniques performed differently when freed from the role of document-length normalization.

For each experiment, the amount of smoothing used by Jelinek-Mercer remained the same or increased. Dirichlet prior consistently used less smoothing under the experimental conditions. It is interesting that as Jelinek-Mercer is freed from the need to handle a non-uniform $P(Rel|Len)$ present in the test collections, Jelinek-Mercer uses a considerable amount of smoothing to obtain its performance. Both methods used more smoothing for description queries than for the title queries.

The behavior of the description queries for TREC 3 and TREC 7 as shown in Figure 3 are strikingly similar. While we are not sure of the reason for the similarity, one possible reason is that both TREC 3 and 7 description queries contain many non-topical words. For example the queries often begin with phrases like "The document will provide information" or "Identify documents discussing." Many of these non-topical words are not on our stopword list and remain in the queries. TREC 8 description queries tend to be formulated with less "extra" material and may say "What is" or other such similar phrases that are cleaned up by the stopword list.

## 5.1 Dirichlet Prior's Length Bias

It is non-obvious that Dirichlet prior smoothing has a document length bias. Zhai and Lafferty (2001) manipulate Equation 4 to produce the following equivalent formulation:

$$\log P(Q|D) = Q(w) \sum_{w \in Q \wedge w \in D} \log \frac{P(w|M_D)}{\lambda P(w|C)} + |Q| \log \lambda + CQL \qquad (12)$$

where $P(w|M_D)$ is the smoothed probability of word $w$ given the document model $M_D$, $\lambda$ is the linear interpolated smoothing parameter, and $CQL = Q(w)\sum_{w \in Q} \log P(w|C)$. When the smoothing method is Dirichlet prior, $\lambda = 1 - \frac{|D|}{|D|+m}$ as in Equation 11. Recall that for linear interpolated smoothing, $P(w|M_D) = (1-\lambda)P(w|D) + \lambda P(w|C)$ as in Equation 10.

Zhai and Lafferty (2001) use Equation 12 to show that smoothing introduces an IDF-like effect because the first term contains the collection probability, $P(w|C)$, in the denominator. The last term is constant for all documents and has no effect on ranking. Zhai and Lafferty (2001) say we can think of the second term $|Q| \log \lambda$ as "playing the role of document length normalization." For Dirichlet prior smoothing, longer documents will have a smaller $\lambda$ and as Zhai and Lafferty (2001) write, "thus a long document incurs a greater penalty than a short one because of this term."

If Dirichlet prior penalizes *longer* documents, then how can we say that Dirichlet prior's performance advantage comes from penalizing shorter documents? The equations in IR may sometimes be simple, but considerable complexity comes from the data contained in the counts of the words in each document. We can't determine the length bias of Dirichlet prior smoothing without examining the entire scoring equation and also evaluating it in terms of actual document collections.

For example, assume a document contains each query term and for each query term $w \in Q$ we let $P(w|D) = P(w|C)$, then no matter what value $\lambda$ is given between 0 and 1, we know that $P(w|M_D) = P(w|D) = P(w|C)$. In this case, linear interpolated smoothing will have no effect on the resulting query likelihood score. For Dirichlet prior smoothing the document length will make no difference in the score of the document since the document length only determines $\lambda$.

If the probabilities of the words in the document, $P(w|D)$ are greater than the collection probabilities, $P(w|C)$, then the more a document is smoothed, the lower its score will go. Conversely, if $P(w|D) < P(w|C)$, then the more smoothing that occurs the higher the document's score will go.

Of course, many documents do not contain all the query terms and for some terms $P(w|D)$ may be less than $P(w|C)$ and for others they may be equal or greater than $P(w|C)$. We can see any score bias by subtracting from the Dirichlet prior document score the Jelinek-Mercer score. Figure 5 shows for TREC 3 title queries of two and three terms, the difference on a per document basis between the query log likelihood score produced by Dirichlet prior smoothing

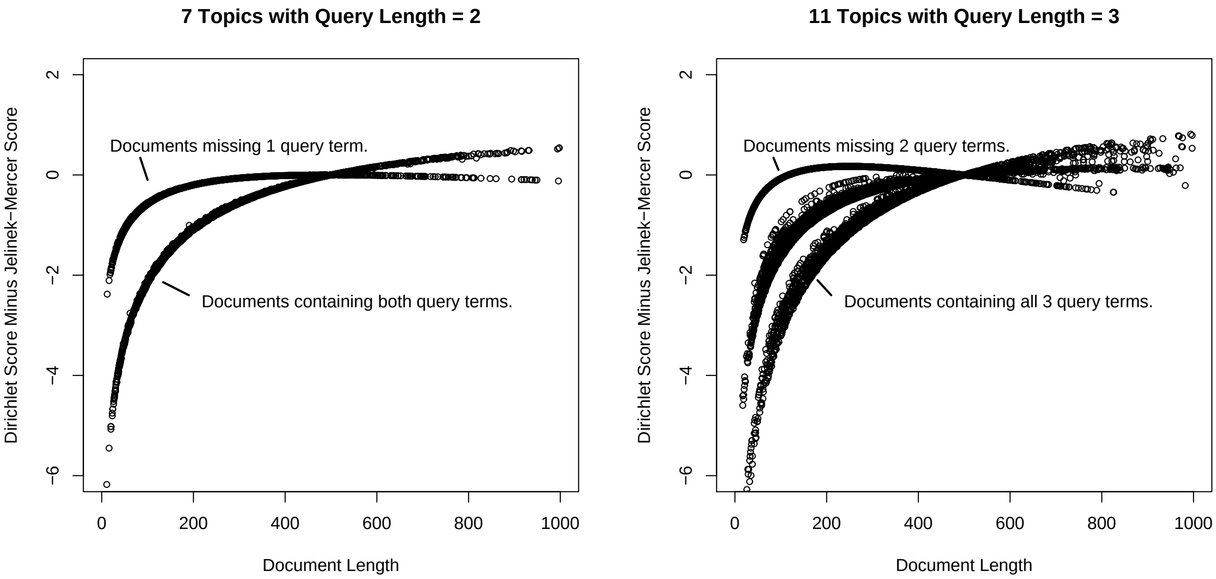

Figure 5: The difference on a per document basis between the query log likelihood score produced by Dirichlet prior smoothing and Jelinek-Mercer for the top 1000 documents for TREC 3 title queries of two and three terms. Dirichlet prior's $m = 500$ and Jelinek-Mercer $\lambda = 0.5$.

and Jelinek-Mercer for the top 1000 documents.

For this figure, Dirichlet prior's $m = 500$ and Jelinek-Mercer $\lambda = 0.5$. Dirichlet prior's effective $\lambda$ at 500 equals 0.5 and this explains why at a document length of 500, the document scores for both methods are equal.

As the document length becomes shorter, Dirichlet prior increases the amount of smoothing. We can clearly see that in general the more smoothing that Dirichlet prior applies to documents, the lower their score becomes. Documents that are missing some of the query terms are helped somewhat by the additional smoothing but very short documents are still penalized. As the query length increases, for documents that are missing query terms, Dirichlet will increasingly prefer documents that are neither very short or long. Given that longer documents are more likely to be relevant, this could be one of the issues that limits Dirichlet prior's performance with long queries.

While Dirichlet's bias depends on the number of query terms, the number of query terms missing from a document, and the collection probability of the terms, we can see from Figure 5 that Dirichlet prior has a length bias that can be thought of as behaving like a document prior. This document prior penalizes short documents. In effect, Dirichlet prior can be approximated by using a document prior in conjunction with Jelinek-Mercer smoothing.

## 5.2 Smoothing Longer Documents Less

Outside of the advantage of preferring longer documents, does it makes sense to smooth longer documents less? Linear interpolated smoothing (and thus Dirichlet prior) is a discounting smoothing method. Discounting methods reduce the probability of the words seen and reallocate the probability mass to words not seen in the document. The mass assigned to the unseen words is called the *zero probability mass.* Neither Jelinek-Mercer nor Dirichlet prior smoothing specify the amount of discounting explicitly but instead an increase in the value of their smoothing parameters results in more discounting. Good-Turing is another form of discounted smoothing. Good-Turing explicitly uses the zero probability mass, $P_0$, and estimates it for a document $D$ to be:

$$P_0 = \frac{\mathrm{N}_1(D)}{|D|}$$

where $\mathrm{N}_1(D)$ is the number of words that occur exactly once in the document $D$ (Sampson 2001). We will not use or discuss Good-Turing smoothing beyond using its estimation of the zero probability mass. Gale and Sampson (1995) provide a good explanation of Good-Turing smoothing.

The $\lambda$ parameter for linear interpolated smoothing can be determined directly from the Good-Turing estimate of the zero probability mass. To do this, we take the sum of the seen probabilities and set the sum equal to $1 - P_0$ and solve for the smoothing parameter. For linear interpolated smoothing, the $P_0$ derived $\lambda$ is:

$$\begin{aligned} \sum_{w \in D} ((1-\lambda)P(w|D) + \lambda P(w|C)) &= 1 - P_0 \\ \lambda &= \frac{P_0}{1 - \sum_{w \in D} P(w|C)} \end{aligned} \tag{13}$$

A similar derivation can be done for the Dirichlet prior parameter $m$, but this merely produces an identical smoothing method. Using Equation 13, we can determine the amount to smooth each separate document based on the Good-

Turing estimate of its zero probability mass.

Figure 6 shows the $P_0$ derived $\lambda$ values for a random set of two thousand documents from the 1.6 million documents comprising TREC disks 1-5 minus the CR collection on discs 4 and 5. The curve marked "Average" is the average of the 1.6 million documents' $P_0$ derived $\lambda$'s after binning the documents by length. To produce a smoother average curve, each bin has a minimum of 1000 documents and at least 2 document lengths. Also shown is the equivalent $\lambda$ value for the Dirichlet prior parameter $m = 1000$. Dirichlet prior follows the general trend of the $P_0$ derived $\lambda$ values; longer documents receive less smoothing than shorter documents. Jelinek-Mercer (JM) smoothing, on the other hand, smooths long and short documents equally and is seen as a horizontal line in the figure for $\lambda = 0.8$. It thus could be argued that Dirichlet prior is correct in smoothing longer documents less if we believe in the Good-Turing estimate of the zero probability mass, $P_0$. In comparison, JM smoothing appears to use too little smoothing for very short documents and smooths long documents too much.

Good-Turing calls for much less smoothing than is needed for good document retrieval. We next discuss why so much smoothing is used by both Dirichlet prior and Jelinek-Mercer.

## 5.3 IDF Behavior of Smoothing

If it is correct to use as little smoothing as suggested by the Good-Turing estimate of the zero probability mass, then why do Dirichlet prior and Jelinek-Mercer use so much more smoothing than appears to be needed for good estimation? Both Hiemstra and de Vries (2000) and Zhai and Lafferty (2001) have shown that smoothing the document model with the collection model can be viewed as introducing an inverse document frequency (IDF) like behavior to the query likelihood retrieval model. Zhai and Lafferty (2001) and our experimental results show that longer, verbose queries require more document smoothing than shorter queries. As we will illustrate with an example, high levels of smoothing increase the importance of rare terms relative to common terms. In other words, the IDF-like behavior shown to exist by Zhai and Lafferty is accentuated with high levels of smoothing.

When a document is scored using query likelihood as in Equation 4, each term in the query contributes to the document's score. When ranking documents, it is their scores relative to each other that matters. If a query consisted

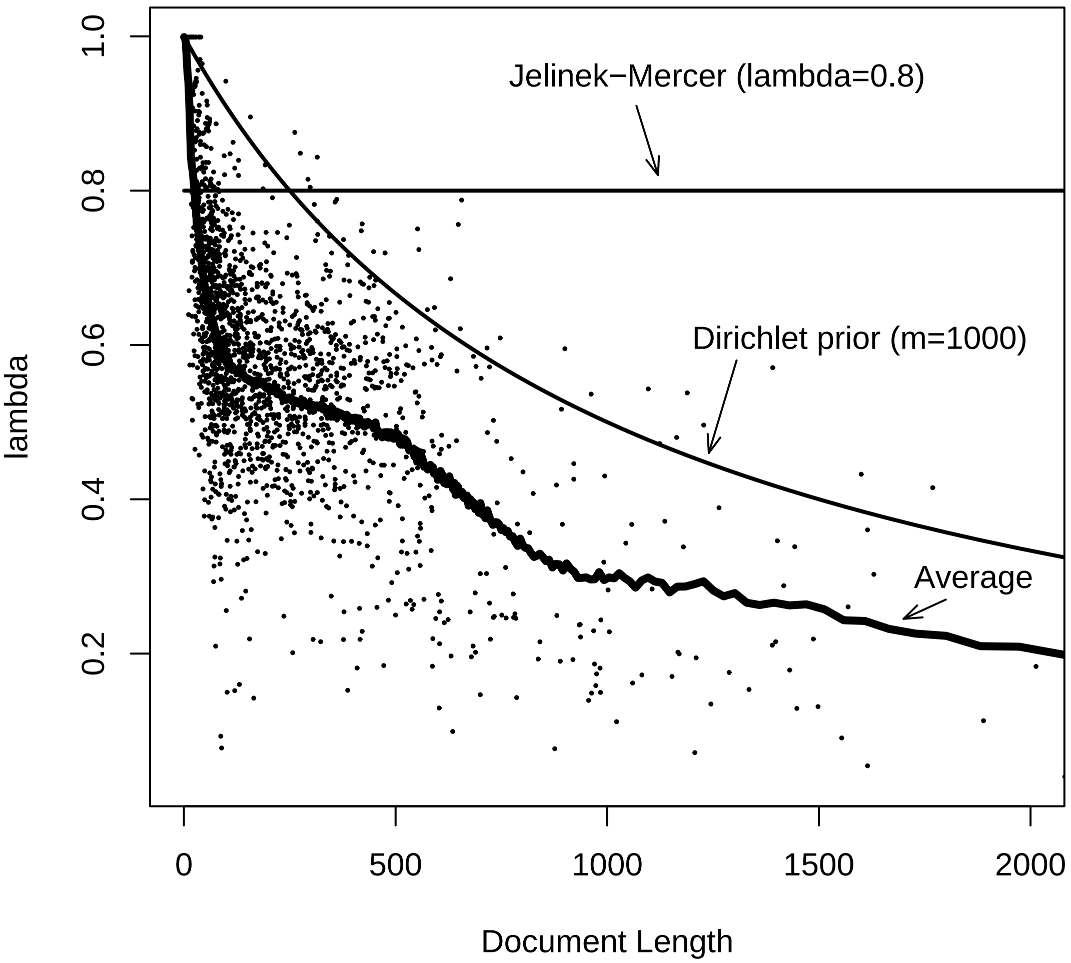

Figure 6: This figure shows the $P_0$ derived $\lambda$ for two thousand randomly selected documents from the 1.6 million documents comprising TREC discs 1-5 minus the CR collection on discs 4 and 5. The curve marked "Average" is the average of the 1.6 million documents' $P_0$ derived $\lambda$'s after binning the documents by length. Also shown is the equivalent $\lambda$ value for the Dirichlet prior smoothing method with $m$ set to 1000. Jelinek-Mercer smoothing is plotted with $\lambda = 0.8$.

of two words $w_1$ and $w_2$, the ratio of a document $A$ to a document $B$ tells us to what extent either one is more likely to have generated the query:

$$\frac{P(w_1|M_A)P(w_2|M_A)}{P(w_1|M_B)P(w_2|M_B)}$$

where $M_A$ and $M_B$ are the smoothed models of documents $A$ and $B$. This ratio is simply a product of the ratios for each word. The ratio for word $w_1$ is:

$$\frac{(1-\lambda)P(w_1|A)+\lambda P(w_1|C)}{(1-\lambda)P(w_1|B)+\lambda P(w_1|C)}$$

Let $P(w_1|A) = P(w_2|A) = 0.003$ and $P(w_1|B) = P(w_2|B) = 0.001$. Document $A$ is the superior document. When $\lambda = 0$ the ratio of $A$'s score to $B$'s is 9:1 and each word contributes equally to $A$'s higher score. If we increase $\lambda$, the individual word ratios will change from 3:1 to ratios nearer to 1:1 until $\lambda = 1$ and the 1:1 ratio is obtained. The way the individual ratios change though depends on their respective collection probabilities.

Let us further assume that $w_1$ is a rare term and $w_2$ is a common term. To determine what makes a term rare or common, we can look at the actual collection probabilities for words found in the description queries. The words used in description queries are skewed to rare informative words, but many common and less-informative words are also used. For topics 351-450, the minimum collection probability of a query term is $7.3 \times 10^{-8}$ and the maximum is $3.1 \times 10^{-3}$. The median probability is $2.6 \times 10^{-4}$. Let us assume that the first quartile is a good representative of a rare term and the third quartile represents a common word. We thus let $P(w_1|C) = 6.0 \times 10^{-5}$ (rare) and $P(w_2|C) = 4.6 \times 10^{-4}$ (common).

Figure 7 shows the scenario just described. As $\lambda$ increases from 0 to 1, the 3:1 ratio for each word changes at different rates. The rare word $w_1$ has a document probability that is large relative to its collection probability and thus requires significantly more smoothing to affect the ratio between documents $A$ and $B$. The common word being closer to its collection probability moves faster to a 1:1 ratio as smoothing is increased. For this example, the result is that at $\lambda = 0.86$ the effective power of the rare word over the common word is maximized.

Informative words are characterized as occurring in bursts and being unevenly distributed in the collection while non-informative words are more evenly distributed (Church 2000). A common heuristic to identify informative words is the inverse document frequency (IDF). In the language modeling approach with

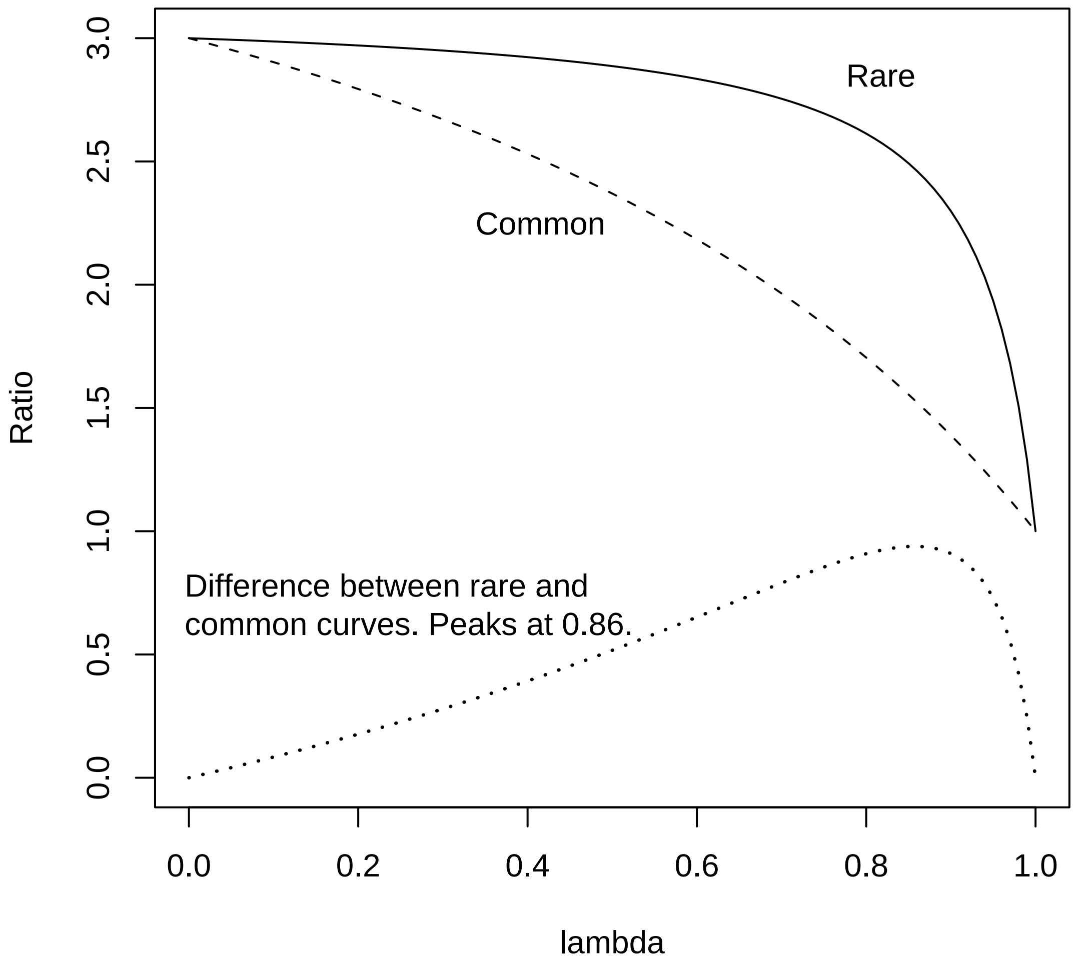

Figure 7: This figure shows an example illustrating that large amounts of linear interpolated smoothing increase the IDF effect of smoothing documents with the collection. As the linear interpolated smoothing parameter $\lambda$ is increased from 0 to 1, the relative impact of the common term decreases at a faster rate than the rare term. Also plotted is the difference between the two curves. In this example, at $\lambda = 0.86$ the importance of the rare term compared to the common term is maximized.

documents smoothed with the collection, IDF is replaced by the inverse collection probability, which functions similarly. Informative, rare words will tend to have large document probabilities relative to the collection probability. Thus for informative words their influence on a document's ranking is little changed until large amounts of smoothing are applied. Common words will likely have document probabilities already near the collection probabilities. Thus common words lose their influence on a document's ranking much faster than rare words as the amount of smoothing increases.

The power of rare words will tend to be amplified with high levels of smoothing. This is the likely explanation of why Dirichlet prior and Jelinek-Mercer smoothing succeed with so much smoothing even when for estimation purposes they should be using less smoothing. This is a surprising notion given that increased smoothing should be used to correct poor model estimates. Instead we find that smoothing more, but not too much, increases the weight given to rarer words in a query. In other work, we've found that document retrieval performance can increase if we decouple smoothing's IDF and estimation roles (Smucker and Allan 2006).

We can generalize our example. We will still only consider two words, $w_1$ and $w_2$, but we will make the ratio a variable $R$ such that one document has $R$ times the word probabilities of the other document. We want to maximize the difference between the two term ratios:

$$f(\lambda) = \frac{R(1-\lambda)P(w_1|D) + \lambda P(w_1|C)}{(1-\lambda)P(w_1|D) + \lambda P(w_1|C)} - \frac{R(1-\lambda)P(w_2|D) + \lambda P(w_2|C)}{(1-\lambda)P(w_2|D) + \lambda P(w_2|C)} \quad (14)$$

Taking the derivative of $f(\lambda)$ and solving for its roots, we find that the $\lambda$ that maximizes the difference is:

$$\lambda = \frac{P(w_1|D)P(w_2|D) - \sqrt{P(w_1|D)P(w_2|D)P(w_1|C)P(w_2|C)}}{P(w_1|D)P(w_2|D) - P(w_1|C)P(w_2|C)} \quad (15)$$

To get a feel for the behavior of Equation 15, we can look at the probabilities of query terms in relevant documents and in the collection. Table 2 shows the average probabilities for the query terms and test collections we studied. The document probabilities are on average over 10 times greater than the collection probabilities. Figure 8 shows the behavior of Equation 15 with $P(w_1|D) = P(w_2|D) = 0.007$ and $P(w_1|C) = 0.0006$ as $P(w_2|C)$ is varied from 0 to 0.01. In general, it appears that the large amounts of smoothing used by Jelinek-Mercer can be explained by the need to distinguish between rare and common

| Query | Collection | Topics | $P(w\|D)$ | $P(w\|C)$ |
|---|---|---|---|---|
| title | 1&2 | trec 3 | 0.008 | 0.0006 |
| title | 4&5-CR | trec 7 | 0.008 | 0.0004 |
| title | 4&5-CR | trec 8 | 0.011 | 0.0004 |
| desc. | 1&2 | trec 3 | 0.004 | 0.0007 |
| desc. | 4&5-CR | trec 7 | 0.004 | 0.0007 |
| desc. | 4&5-CR | trec 8 | 0.006 | 0.0006 |
| | | Average | 0.007 | 0.0006 |

Table 2: The average word probabilities for query terms in the relevant documents, $P(w|D)$, and for the collections, $P(w|C)$.

terms.

## 5.4 Two Stage Smoothing

Given that our entire discussion is about Jelinek-Mercer and Dirichlet prior smoothing, it is only natural to address the two stage smoothing method created by Zhai and Lafferty (2002) that combines Jelinek-Mercer and Dirichlet Prior smoothing. One of the goals of two stage smoothing is to obtain a smoothing method that performs well at the roles Zhai and Lafferty (2002) have described as the estimation and query modeling roles.

Given that we've shown in this paper that Dirichlet prior's performance advantage comes more from a penalization of shorter documents than from its potentially better estimation of document models, two stage smoothing is better understood as a smoothing method designed to penalize short documents by smoothing them more than long documents while also using enough smoothing for all document lengths to provide adequate discrimination between rare and common terms.

Rather than linearly combine the MLE document model with the MLE collection model, two stage smoothing substitutes the MLE document model with the smoothed Dirichlet prior model:

$$P(w|M_D) = (1-\alpha)\frac{D(w) + mP(w|C)}{|D| + m} + \alpha P(w|C) \tag{16}$$

where $m$ is the Dirichlet prior smoothing parameter and $\alpha$ can vary between 0 and 1.

Two stage smoothing includes innovative work on automatically determining

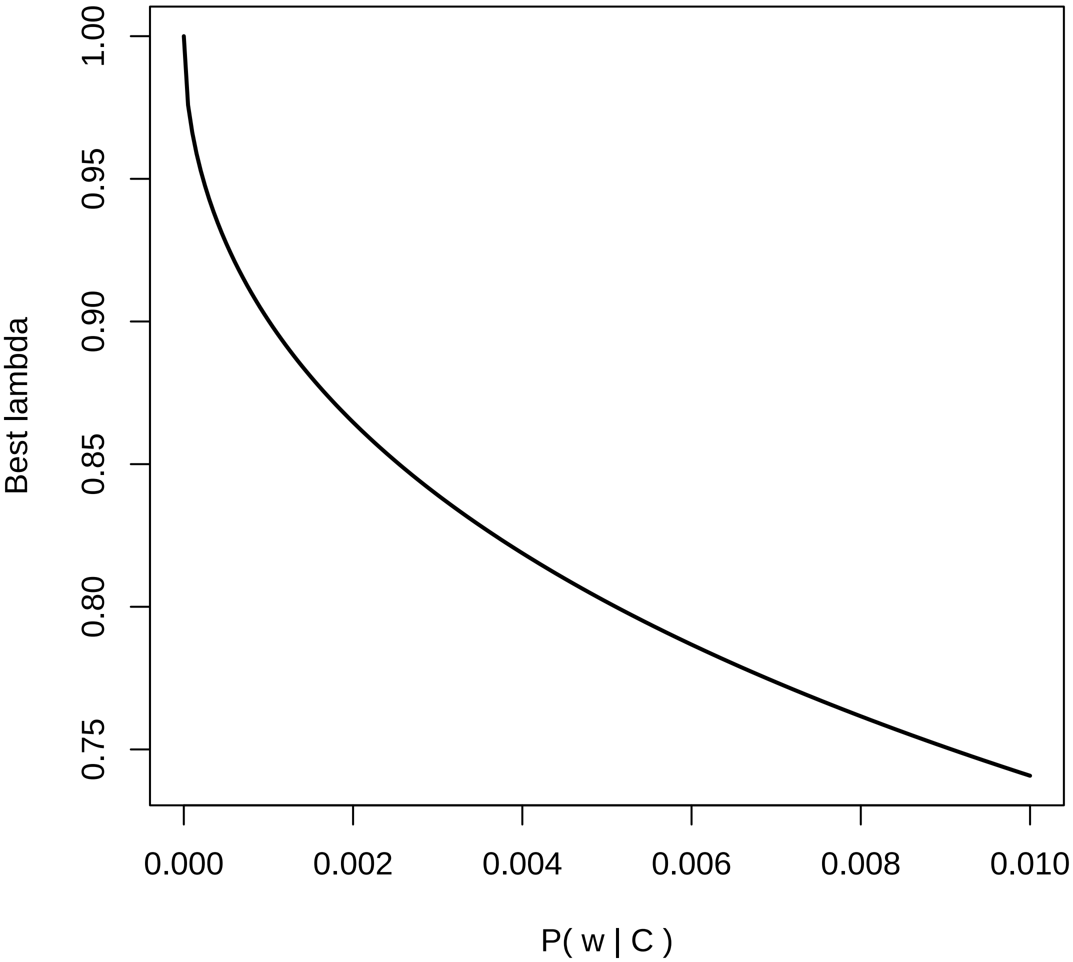

Figure 8: The response of Equation 15 as $P(w_2|C)$ is varied from 0 to 0.1 with with $P(w_1|D) = P(w_2|D) = 0.007$ and $P(w_1|C) = 0.0006$. While this figure is a correct representation Equation 15, for these variable values, when $P(w_2|C) = 0.0006$, then $f(\lambda) = 0$ and any setting of $\lambda$ maximizes Equation 14.

the values for the parameters $\alpha$ and $m$, but the smoothing is still linear interpolated smoothing. A linear transformation of a linear transformation produces another linear transformation. Thus, we can write Equation 16 in the form of linear interpolated smoothing by setting $\lambda$ in Equation 10 as follows:

$$\lambda = 1 - \frac{|D| - \alpha|D|}{|D| + m} \tag{17}$$

Figure 9 shows an example of the behavior of $\lambda$ in Equation 17 with $\alpha = 0.5$ and $m = 300$. Two stage smoothing pushes the tail of the Dirichlet prior curve upwards. Two-stage smoothing sets a new minimum amount of smoothing such that $\lambda$ varies between 1 and $\alpha$ instead of 1 and 0. A small $m$ and a large enough $\alpha$ should allow two stage smoothing to penalize short documents by smoothing them more than long documents while also using enough smoothing for all document lengths to maximize performance.

# 6 Conclusion

Dirichlet prior smoothing should produce better estimated document models than Jelinek-Mercer smoothing, for the longer a document is, the less Dirichlet prior smooths it. When Zhai and Lafferty (2001, 2002) discovered that Dirichlet prior outperformed Jelinek-Mercer, they suggested that estimation was the reason for Dirichlet prior's performance advantage. In contrast, we found that Dirichlet prior's advantage comes more from a penalization of shorter documents than from its estimation. In other words, the primary advantage gained from smoothing longer documents less is that Dirichlet prior has a document length bias that results in better retrieval performance on TREC collections. From an estimation standpoint, it does make sense to smooth longer document less, but better estimation is not the cause of Dirichlet prior's performance advantage. Smoothing's theoretical role in the language modeling approach to information retrieval has been one of better estimation. Instead we have found that one of its more important practical roles, as played by Dirichlet prior smoothing, has also been to penalize the scores of shorter documents. This is the same role as played by pivoted document-length normalization in tf-idf retrieval techniques (Singhal et al. 1996). Language modeling retrieval should preferentially weight documents more likely to be relevant with a scoring component such as a document prior. Separately modeling the probability of relevance given doc-

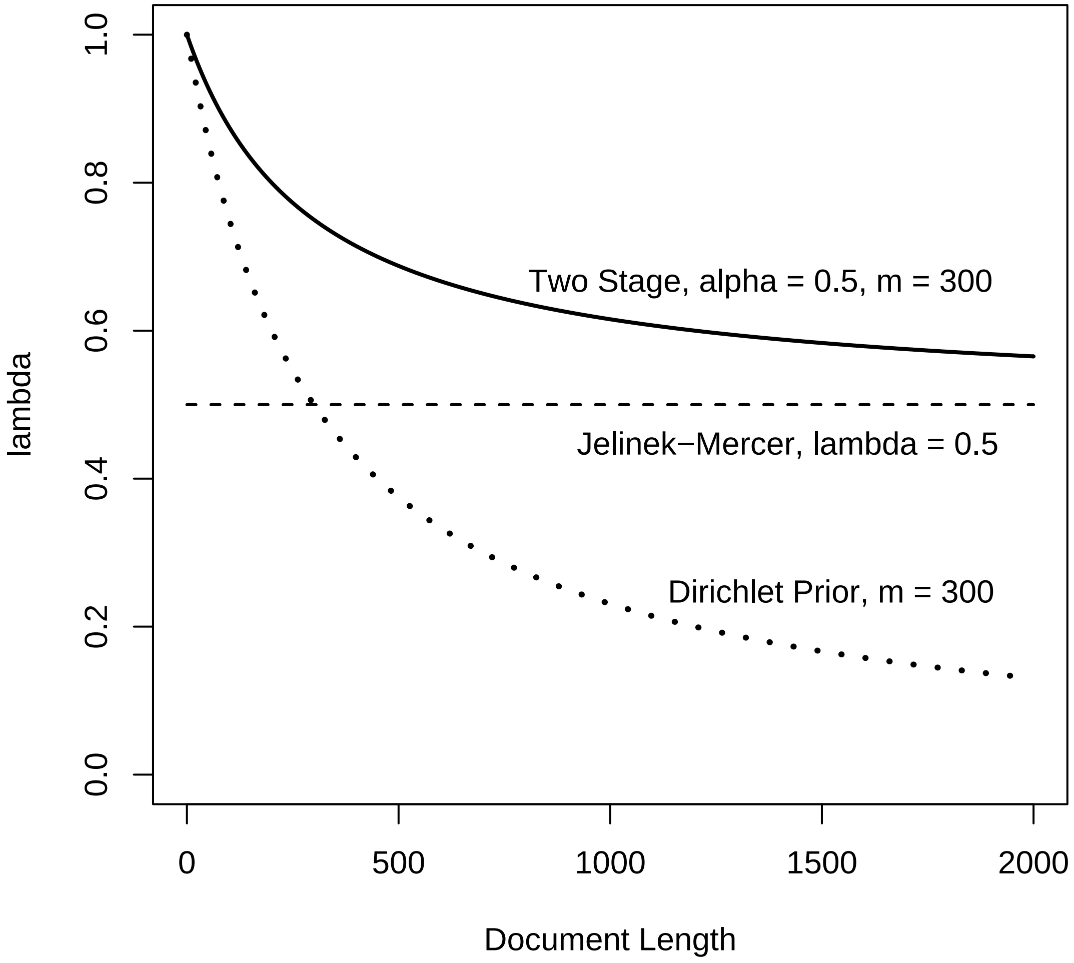

Figure 9: An example of the effective $\lambda$ of two stage, Jelinek-Mercer, and Dirichlet prior smoothing. All three are forms of linear interpolated smoothing that linearly combine the MLE document model with the MLE collection model: $P(w|M_D) = (1-\lambda)P(w|D) + \lambda P(w|C)$.

ument length will remove a confounding component from document smoothing and should allow the remaining roles of smoothing to be better understood and optimized.

### Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval, the Defense Advanced Research Projects Agency (DARPA) under contract number HR0011-06-C-0023, and by SPAWARSYSCEN-SD grant number N66001-02-1-8903. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

## References

Berger A and Lafferty J (1999) Information retrieval as statistical translation. In: SIGIR '99: Proceedings of the 22th annual international ACM SIGIR conference on research and development in information retrieval. ACM Press, pp. 222–229.

Buckley C (2005) The SMART project at TREC. In: Voorhees EM and Harman DK, eds., TREC, MIT Press, chap. 13. pp. 301–320.

Chen SF and Goodman J (1998) An empirical study of smoothing techniques for language modeling. Tech. Rep. TR-10-98, Center for Research in Computing Technology, Harvard University.

Church KW (2000) Empirical estimates of adaptation: the chance of two Noriegas is closer to $p/2$ than $p^2$. In: Proceedings of the 17th Conference on Computational Linguistics. Association for Computational Linguistics, pp. 180–186.

Gale WA and Sampson G (1995) Good-Turing frequency estimation without tears. Journal of Quantitative Linguistics, 2(3):217–237. Reprinted (Sampson 2001, chap. 7).

Hiemstra D and de Vries A (2000) Relating the new language models of information retrieval to the traditional retrieval models. CTIT Technical Report TR-CTIT-00-09, Centre for Telematics and Information Technology, University of Twente, Enschede. ISSN 1381-3625.

Hiemstra D and Kraaij W (1998) Twenty-One at TREC-7: Ad-hoc and cross-language track. In: The Seventh Text REtrieval Conference (TREC-7). Department of Commerce, National Institute of Standards and Technology, pp. 227–238.

Johnson WE (1932) Probability: deductive and inductive problems. Mind, 41(164):409–423.

Krovetz R (1993) Viewing morphology as an inference process. In: SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on research and development in information retrieval. ACM Press, pp. 191–202.

Lemur (2003) Lemur Toolkit for Language Modeling and IR. `http://www.lemurproject.org/`.

Miller DRH, Leek T and Schwartz RM (1998) BBN at TREC7: Using hidden markov models for information retrieval. In: The Seventh Text REtrieval Conference (TREC-7). Department of Commerce, National Institute of Standards and Technology, pp. 133–142.

Mitchell TM (1997) Machine Learning. McGraw-Hill.

Narayanan A (1991) Algorithm as 266: Maximum likelihood estimation of the parameters of the Dirichlet distribution. Applied Statistics, 40(2):365–374.

Ponte JM and Croft WB (1998) A language modeling approach to information retrieval. In: SIGIR '98: Proceedings of the 21th annual international ACM SIGIR conference on research and development in information retrieval. ACM Press, pp. 275–281.

Sampson G (2001) Empirical Linguistics. Continuum.

Singhal A, Buckley C and Mitra M (1996) Pivoted document length normalization. In: SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on research and development in information retrieval. ACM Press, pp. 21–29.

Sjölander K, Karplus K, Brown M, Hughey R, Krogh A, Mian IS and Haussler D (1996) Dirichlet mixtures: a method for improved detection of weak but significant protein sequence homology. Computer Applications in the Biosciences, 12:327–345.

Smucker MD and Allan J (2005) An investigation of Dirichlet prior smoothing's performance advantage. Tech. Rep. IR-391, Center for Intelligent Information Retrieval (CIIR), Department of Computer Science, University of Massachusetts Amherst.

Smucker MD and Allan J (2006) Lightening the load of document smoothing for better language modeling retrieval. In: SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval. ACM Press, pp. 699–700.

Smucker MD, Allan J and Carterette B (2007) A comparison of statistical significance tests for information retrieval evaluation. In: CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management. ACM, New York, NY, USA, pp. 623–632.

Song F and Croft WB (1999) A general language model for information retrieval. In: CIKM '99: Proceedings of the eighth international conference on information and knowledge management. ACM Press, pp. 316–321.

Zhai C and Lafferty J (2001) A study of smoothing methods for language models applied to ad hoc information retrieval. In: SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval. ACM Press, pp. 334–342.

Zhai C and Lafferty J (2002) Two-stage language models for information retrieval. In: SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval. ACM Press, pp. 49–56.

# Notes

[1] Center for Intelligent Information Retrieval, Department of Computer Science, University of Massachusetts, Amherst, MA 01003. Email: smucker@cs.umass.edu and allan@cs.umass.edu