Inf Retrieval (2008) 11:109–138
DOI 10.1007/s10791-007-9040-x

An analysis on document length retrieval trends in language modeling smoothing

David E. Losada · Leif Azzopardi

Received: 24 July 2007 / Accepted: 7 December 2007 / Published online: 20 December 2007

Abstract Document length is widely recognized as an important factor for adjusting retrieval systems. Many models tend to favor the retrieval of either short or long documents and, thus, a length-based correction needs to be applied for avoiding any length bias. In Language Modeling for Information Retrieval, smoothing methods are applied to move probability mass from document terms to unseen words, which is often dependant upon document length. In this article, we perform an in-depth study of this behavior, characterized by the document length retrieval trends, of three popular smoothing methods across a number of factors, and its impact on the length of documents retrieved and retrieval performance. First, we theoretically analyze the Jelinek–Mercer, Dirichlet prior and two-stage smoothing strategies and, then, conduct an empirical analysis. In our analysis we show how Dirichlet prior smoothing caters for document length more appropriately than Jelinek–Mercer smoothing which leads to its superior retrieval performance. In a follow up analysis, we posit that length-based priors can be used to offset any bias in the length retrieval trends stemming from the retrieval formula derived by the smoothing technique. We show that the performance of Jelinek–Mercer smoothing can be significantly improved by using such a prior, which provides a natural and simple alternative to decouple the query and document modeling roles of smoothing. With the analysis of retrieval behavior conducted in this article, it is possible to understand why the Dirichlet Prior smoothing performs better than the Jelinek–Mercer, and why the performance of the Jelinek–Mercer method is improved by including a length-based prior.

Keywords Language models · Smoothing · Document length

D. E. Losada (✉)
Departamento de Electrónica y Computación, Universidad de Santiago de Compostela, Santiago, Spain
e-mail: dlosada@dec.usc.es

L. Azzopardi
Department of Computing Science, University of Glasgow, Glasgow, Scotland
e-mail: leif@dcs.gla.ac.uk

1 Introduction

The problem of document length normalization (Singhal et al. 1996a; Robertson and Walker 1994) is ensuring that documents of particular lengths are not unduly favored over documents of other lengths by the retrieval model. The need to account for this problem is because (Singhal et al. 1996a): (1) Long documents tend to have more occurrences of different terms which means that long documents are more likely to match query terms, and; (2) As the length of the document increases, the number of times a particular term occurs in the document also increases, which in turn increases the matching score. Consequently, the term weights in a document need to be penalized in accordance to document length (and thus normalize the document). And accounting for document length effects within a retrieval algorithm tends to improve performance (Singhal et al. 1996a; Amati 2003; Chowdhury et al. 2002). Although these normalization issues have been extensively studied in the context of many IR models, such as the Vector-Space model with tf/idf weighting (Singhal et al. 1996a), the classic Probabilistic model (Robertson et al. 1995) and Divergence from Randomness models (Amati and van Rijsbergen 2002), the effect of document length has scarcely been discussed in the context of Language Modeling (LM).

In LM (Ponte and Croft 1998; Miller et al. 1999; Hiemstra 2000), smoothing methods are applied to move probability mass from document terms to unseen words when constructing a LM for a document. This provides an implicit length normalization component, where the amount of smoothing applied affects the distribution of the lengths in the retrieved set of documents. The smoothing method and parameter estimation will dictate whether longer or shorter documents are favored, or not. In Singhal et al. (1996a), the distribution of the length of documents is referred to as a length pattern, so the documents retrieved by a model produce a retrieval pattern. In the context of the Vector-Space Model, it was shown that by tailoring the retrieval pattern to the relevant pattern[1] improved performance. Not accounting for the length of documents could lead to a serious degradation to retrieval performance. Also, the benefits of document length normalization have been demonstrated in the context of other retrieval models, such as the classic Probabilistic Model (Robertson et al. 1995) or Divergence from Randomness models (Amati 2003). As a matter of fact, the evaluation of different term weighting schemes able to correct document length has been a prominent research topic as exemplified by the TREC experiments across the years (Harman 2005).

In this article, we analyze the length retrieval patterns of different LMs and how the retrieval performance is affected as the parameters are modified. To this aim, we compare and contrast two common smoothing methods used for estimating LMs, which is performed on a theoretical and empirical level. The two methods are Bayesian smoothing using Dirichlet priors (DP), which smoothes proportionally to the length of the document, and Jelinek–Mercer (JM) smoothing, which does not consider document length in the smoothing process. Thus, we explore how the difference in estimation affects the distribution of the length of documents retrieved (i.e. behavior of the retrieval function w.r.t. document length), and its impact on performance (effectiveness). While, it has been previously shown that DP tends to be better than JM in terms of effectiveness, an aim of this study is to answer the question, *why DP is better than JM* by examining their behaviorial differences. We contextualize this study by also examining the dual role of smoothing and the combination of DP and JM smoothing methods under the two-stage LM, and seek to understand (i) what role(s) the retrieval functions play, (ii) how the retrieval function

[1] The relevant pattern is defined by the set of relevant documents.

affects the behavior, (iii) how the behavior of the model affects performance, and (iv) how this affects document length normalization issues.[2]

The course of this study leads to a number of insights into the interactions between length retrieval patterns, smoothing and performance. In the process, we provide further insight into why DP smoothing is better than JM for ad hoc retrieval in terms of effectiveness (Zhai and Lafferty 2004), and that tailoring the retrieval model to retrieve documents with lengths similar to the relevant documents improves the overall performance (Singhal et al. 1996b). More importantly, we provide an explanation as to why this is the case, by examining the theory to understand the differences between the methods, the behavior to show the differences in terms of observable effects on length, and how this directly influences the system performance. We show that the retrieval patterns produced by DP smoothing more closely match the relevant patterns, while JM smoothing tends to retrieve documents that are much shorter than those in the relevant patterns. This behavior explains why DP smoothing outperforms JM smoothing in terms of performance.

We also show that the automatic parameter estimation in the two-stage smoothing method tends to produce length retrieval patterns that are close from the relevant pattern explaining the excellent retrieval performance of the two stage model. The two-stage models are motivated by the need to account for the dual role of smoothing, however, our theoretical analysis shows that a conflict between the roles exists. Consequently, we consider document priors as a way to independently handle the document length normalization issue (and decouple the roles within the retrieval function). If a document prior based on length is introduced then the tendency of JM smoothing to retrieve document that are too short can be compensated, which significantly improves the method's performance. On the other hand, DP smoothing with a document prior did not attract such improvements as the length based prior interfered with the implicit normalization within the DP smoothing method (and tended to retrieve document much longer than those in the relevant patterns).

This analysis leads to a better understanding of the mechanisms involved in the retrieval functions and how this translates in observable behavior and subsequent retrieval performance; and so the different characteristics of each smoothing method can be precisely understood, predicted and witnessed. The rest of the article is organized as follows: Sect. 2 introduces the Language Modeling approach for ad hoc retrieval and explains the smoothing techniques used for estimation. Section 3 describes how the estimates will affect the ranking of documents w.r.t. length in a theoretical manner. Then in Sect. 4, an empirical analysis is conducted to demonstrate the behavior characterized by retrieval length patterns of the three smoothing techniques and the impact upon performance. We report additional experiments using length-based priors in Sect. 5 and the main findings of this article are discussed in Sect. 6. Finally, we conclude the article with a summary of this work before outlining directions for future research.

2 Language modeling

The probability of a query as being generated by a probabilistic model based on a document (query likelihood) is one of the most standard approaches in LM for Information

[2] where we re-state the problem of document length normalization, as the objective to retrieve documents which are like relevant documents w.r.t. length. See (Azzopardi and Losada 2007) for an analysis of the original goal using Language Models.

Retrieval (Ponte and Croft 1998; Hiemstra 1998; Miller et al. 1999). This formulation results from computing the probability of a document given a query, $P(d|q)$, using Bayes' rule so that the ranking is proportional to the query likelihood:

$$P(d|q) = \frac{P(q|d) \cdot P(d)}{P(q)} \tag{1}$$

$$P(d|q) \stackrel{rank}{=} P(q|d) \tag{2}$$

where $P(q)$ is a constant which can be dropped since it does not affect the ranking. For now we shall assume that $P(d)$ is uniform across all documents. However, in Sect. 5 we will consider a non-uniform prior.

While there have been many different methods proposed for estimating $P(q|d)$, in this study we focus on the estimating the query likelihood using Unigram LMs (Hiemstra 1998; Zhai and Lafferty 2001). This is an standard approach commonly applied throughout the literature because it is simple, intuitive, efficient and effective. Under the query likelihood approach, document scoring is reduced to the following steps: (1) estimate a Unigram LM for each document, $p(.|d)$, and, (2) compute the probability of generating the query q from each document model (where each query term q_i is assumed to be sampled independently and identically from the document):

$$P(q|d) = \prod_{i=1}^{n} P(q_i|d) \tag{3}$$

Given the document's text there is a fruitful stream of research in the field of statistical natural language processing dedicated to smoothing techniques that distribute the probability mass between the words which are seen in the document and the terms that are not (Chen and Goodman 1998; Manning and Schütze 1999). This is very important in IR because it is very likely that a given user query mentions some non-document terms and, thus, words unseen in the document should be assigned a non-zero probability in the language model of the document (otherwise, a single non matching term would produce a query likelihood of zero, usually referred to as the Zero Probability Problem). As a consequence, a background collection, usually composed of a large number of texts, is used to define a fallback model that reflects the general use of the language and, therefore, is a good tool to smooth with.

The two predominate smoothing methods used are Jelinek Mercer (JM) smoothing and Bayes Smoothing with Dirichlet priors (DP). In (Zhai and Lafferty 2001; Zhai and Lafferty 2004), these smoothing methods were analyzed, along with Absolute Discounting for ad hoc retrieval on a number of test collections using queries of different lengths. For each collection tested, the same trends were found. It was shown that DP smoothing generally performs the best, though it performed better for short queries, than long queries. However, it was also shown that JM smoothing provides comparable performance to DP, but JM tends to perform better for longer queries than shorter queries. Both these methods were shown to be better than the other method tested; and in (Azzopardi 2005), they were shown to significantly outperform Laplace smoothing. Due to their simplicity and effectiveness, these two smoothing methods have been widely used in the literature and are representative of standard smoothing methods to construct Language Models. Additionally, in the next section, we introduce two-stage smoothing (Zhai and Lafferty 2002) , which is a powerful method based on combining both smoothing methods.

2.1 Bayesian smoothing with Dirichlet priors

This method results from applying Dirichlet priors within a Bayesian framework (Mackay and Peto 1995). Given a document d and a background collection C, the probability of a term in the LM of the document is computed as:

$$P(w|d) = \frac{tf(w,d) + \mu P(w|C)}{|d| + \mu} \tag{4}$$

where $tf(w, d)$ is the raw term frequency of w in d, $|d|$ is the total count of terms in the document (i.e. $|d| = \sum_w tf(w, d)$), and μ is a parameter for adjusting the amount of smoothing applied, where $\mu \geq 0$. $P(w|C)$ is the probability of the term w occurring in the collection C, and is usually a maximum likelihood estimator computed using the collection of documents (i.e. $P(w|C) = \sum_d tf(w, d)/\sum_{d'}|d'|$), which is a model that suffers less from sparseness. Putting together Eqs. 3 and 4, applying logarithms and re-arranging terms, the retrieval score can be reduced to a simple formula (Zhai and Lafferty 2001, 2004) as shown by Eq. 6:

$$\log P(q|d) \overset{rank}{=} \sum_{i:tf(q_i,d)>0} \log \frac{\frac{tf(q_i,d)+\mu P(q_i|C)}{|d|+\mu}}{\frac{\mu p(q_i|C)}{|d|+\mu}} + n \log \frac{\mu}{|d| + \mu} \tag{5}$$

$$\log P(q|d) \overset{rank}{=} \sum_{i:tf(q_i,d)>0} \log \left(1 + \frac{tf(q_i,d)}{\mu P(q_i|C)}\right) + n \log \frac{\mu}{|d| + \mu} \tag{6}$$

This shows that ranking documents using query likelihood and DP smoothing is essentially reduced to a regular sum across matching terms plus a document dependent constant.

2.2 Jelinek–Mercer smoothing

Jelinek–Mercer smoothing (Jelinek and Mercer 1980) is a traditional linear interpolation method. Although it was originally used to interpolate higher-order n-gram models with lower-order n-gram models (Jelinek and Mercer 1980), its application in IR has been mainly focused on mixing unigram maximum likelihood estimators with unigram background models (Hiemstra 1998; Zhai and Lafferty 2001). This can also be interpreted as a two-state Hidden Markov Model in which the state transitions are defined from the smoothing parameter (Miller et al. 1999). JM involves a linear interpolation between the maximum likelihood estimator given the document d and a fallback model:

$$P(w|d) = (1 - \lambda) \cdot P_{mle}(w|d) + \lambda \cdot P(w|C) \tag{7}$$

$$= (1 - \lambda) \cdot \frac{tf(w,d)}{|d|} + \lambda \cdot P(w|C) \tag{8}$$

The amount of smoothing applied is controlled by λ, which takes values in the interval [0,1].

Comparing Eqs. 4 and 8 one can clearly observe the distinct features of these smoothing methods. Like any other smoothing method, DP and JM produce a movement of probability mass from seen terms (or, in general, from seen events) to unseen material. But this

movement is done in very different fashions. While DP moves probability mass from the seen terms to the unseen terms in a document length-dependent way (short documents are smoothed more than long documents), JM smoothes all documents independently of length (λ is a constant for all documents).

The retrieval score of JM smoothing as the query log-likelihood is reduced to (Zhai and Lafferty 2001, 2004):

$$\log P(q|d) \overset{rank}{=} \sum_{i:tf(q_i,d)>0} \log \frac{(1-\lambda)\cdot \frac{tf(q_i,d)}{|d|} + \lambda \cdot P(q_i|C)}{\lambda \cdot P(q_i|C)} + n \cdot \log \lambda \tag{9}$$

$$\log P(q|d) \overset{rank}{=} \sum_{i:tf(q_i,d)>0} \log \left(1 + \frac{(1-\lambda)}{\lambda} \cdot \frac{tf(q_i,d)}{|d| \cdot P(q_i|C)}\right) \tag{10}$$

The second addend in Eq. 9, while proportional to the query length, is independent of any document feature and thus constant for all documents. Hence, it can be ignored for ranking purposes. Note that JM smoothing is a general case, where DP smoothing can be derived by setting λ equal to $\frac{\mu}{|d|+\mu}$. Intuitively, this shows that DP smoothing applies smoothing proportional to the length of the document, while JM smoothing applies a fixed amount of smoothing, regardless of document length. This small difference results in distinctly different behavior during retrieval which has a substantial impact on retrieval performance.

3 Theoretical analysis

From the ranking formulas of each smoothing method, let us consider the influence of each component on the ranking of documents with respect to the length of documents retrieved (the document length retrieval trends), in an analytical fashion.

DP smoothing (Eq. 6) yields a retrieval formula with a penalty for long documents (second addendum). The effect of this correction is higher for small μ values. As μ grows, the distinction for different lengths is less extreme. Although one could be tempted to infer that little smoothing implies less long documents retrieved, observe that little smoothing yields also more *coordination level ranking* (first addend).[3] That is, small μ values prevent that a document matching n query terms can get a higher sum over matching terms than a document matching $n + 1$ terms (because every single match is multiplied by $1/\mu$). Since long documents profit from coordination level strategies, there will be a point when longer documents are favored in lieu of shorter documents, as the influence from the second addendum is mitigated.

Further note that the length of the query also affects the ranking under DP smoothing because longer documents will incur a larger penalty from the second addendum (proportional to query length). Empirically, it has been seen that long queries tend to need more smoothing (higher μ values) whilst short queries usually work well with less smoothing (Zhai and Lafferty 2004).

Regarding JM smoothing, the final retrieval function is simply governed by a sum over matching terms (Eq. 10). Low smoothing values ($\lambda \approx 0$) lead to a coordination level-like retrieval function: every single match receives a high boost (multiplication by $(1-\lambda)/\lambda$).

[3] For a good discussion about coordination level ranking retrieval strategies we refer to (Hiemstra 2000).

Note that the term frequencies are normalized by the document's length and, therefore, a short document might have higher values of $\frac{tf(q_i,d)}{|d|\cdot P(q_i|C)}$ than a long document. On the other hand, a long document has more chance of matching more query terms. As λ approaches zero, we expect that this second effect prevails because every single match weights more (it is unlikely that a document matching $n-1$ query terms gets a retrieval score that is higher than the score obtained by a document with n matches). With higher smoothing values ($\lambda \gg 0$) there is a move away from coordination level ranking and increasingly more weight to the "idf effect" of the query terms (the relative importance of the query terms is accounted by $P(q_i|C)$). This results in shorter document being favored, and the length of query becomes less important as the "idf effect" weighting dominates the retrieval function.

We now discuss the roles of smoothing that have been identified in the literature, which led to the development of the two-stage smoothing method.

3.1 The dual role of smoothing

The discussion above connects directly with the *dual role of smoothing* witnessed by Zhai and Lafferty (2001, 2004) when studying different smoothing methods. According to Zhai and Lafferty (2001, 2004), smoothing plays two different roles: (i) *estimation role* which aims to improve the accuracy of the estimated documents' Language Models, and (ii) *query modeling role* which explains the common and non-informative words in a query. Although Zhai and Lafferty did not enter into further details (other than suggesting that DP suits the estimation role, and JM suits the query role), we try to contextualize these roles in terms of the retrieval functions derived.

With DP smoothing, the estimation of the documents' Language Models is done in a length-dependent way. This implements the intuition that shorter texts require more smoothing than long texts. In the retrieval formula derived (Eq. 6) this results in a sum across matching terms plus a document length correction. The different roles of the smoothing parameter μ are evident from the inspection of Eq. 6. It is a query modeling tool because as μ grows the relative discriminative power of matching terms (idf effect because of $p(q_i|C)$) receives more importance (i.e. we move away from coordination level). It is also a document modeling tool because it influences the document length correction in the second addendum. With DP smoothing, we therefore argue that the so called estimation role is simply a length retrieval corrector (addressing the issue of document length normalization).

The two roles of smoothing might be conflicting. For instance, a verbose query, with an assorted combination of common and informative terms, will require a high amount of smoothing. This parameter setting might be non-optimal from a document length retrieval point of view. This motivated the development of two-stage level smoothing, which tries to account for both roles separately (Zhai and Lafferty 2002).

In JM smoothing, λ acts only as a query modeling tool (there is no dual role). Unlike μ, λ is not involved in any document length correction. It simply balances the idf effect of the query terms. This smoothing strategy can be naturally interpreted as a discrete Hidden Markov Model (HMM) (Miller et al. 1999). The generation of a query is regarded as a Markov process dependent on the document the user has in mind. Intuitively, when a user formulates a query she/he chooses terms either from this *ideal* document or from the general vocabulary of the language (lexical tissue, etc.). This can be modeled as a 2-state HMM (one state associated to the document and one state to the language). The value of λ determines the relative weights of these two models. For a verbose query λ is expected to

be high (many transitions across the general model of the language) while for a keyword-like query most transitions occur through the document model and, therefore, λ is small. That is, verbose queries require more smoothing to explain the generation of terms with very different discriminative power (more idf effect is required) while keyword queries need less smoothing (the quality of the terms is more balanced). Although there is not an explicit document length correction in JM, there will still be a correlation between λ and the lengths of the documents retrieved. For instance, low λ's lead to coordination level-like retrieval, which tends to promote long documents).

3.2 Two-stage smoothing

Two stage smoothing is an evolved mechanism in which the document LM is firstly smoothed using Dirichlet and, next, the smoothed document LM is interpolated with a query background model (Zhai and Lafferty 2002). Although the background models used in these two stages might be different, it is common to assume that the collection model $p(w|C)$ is a reasonable approximation for both the collection LM and the query background model (Zhai and Lafferty 2002). This leads to the following estimate:

$$p(w|d) = (1-\lambda)\cdot\frac{tf(w,d)+\mu\cdot p(w|C)}{|d|+\mu}+\lambda\cdot p(w|C) \tag{11}$$

An important characteristic of this smoothing method is that the smoothing parameters are estimated in a completely automatic way: μ is estimated using the leave-one-out log likelihood method, which is fixed, and then λ is estimated using document mixture models with the EM algorithm on a query by query basis (i.e. λ is set according to the query). Putting together Eqs. 3 and 11, applying logarithms and re-arranging terms, the retrieval score can be reduced to:

$$\log p(q|d) \overset{rank}{=} \sum_{i:tf(q_i,d)>0} \log\left(1+\frac{(1-\lambda)\cdot tf(q_i,d)}{p(q_i|C)\cdot(\mu+|d|\cdot\lambda)}\right)+\sum_{q_i}\log\left(\frac{\mu+\lambda\cdot|d|}{\mu+|d|}\right) \tag{12}$$

Again, document scoring is essentially reduced to a sum across matching terms plus a document dependent constant. This constant needs to be computed online by traversing all query terms.

Note that Eq. 12 is equivalent to Eq. 6 when λ is equal to 0 and, conversely, Eq. 12 is equivalent to Eq. 10 when μ is equal to 0. This retrieval formula is influenced by both μ and λ. Having two different parameters one can be dedicated to optimize the document modeling role (μ) and the other (λ) is dedicated to query modeling purposes. However, despite being separate, the two parameters may interact with each other in such a way as to re-enforce or mitigate the effect of one another. For instance, increasing λ favors shorter documents, but increasing μ favors longer documents, having a conflicting effect; while decreasing λ instead favors longer documents, re-enforcing the effect from setting μ. This conflict between the two roles suggests that considering ways to decouple the two roles, such that the effects are controlled independently would be preferable.

3.3 Final remarks

Summing up, these smoothing methods apply different approaches to move probability mass from seen terms to unseen material. Bayesian smoothing with Dirichlet priors,

incorporates the intuition that we can trust that most of the relevant material was explicitly mentioned in a long verbose document. As a consequence, little probability mass is left for unseen words. On the contrary, short texts usually skip important terms and, thus, we need to be cautious and assign less probability mass to seen words. On the other hand, JM smoothing assigns probability mass to unseen terms in a non length-dependent manner, and the intuition above is not entertained. Consequently, the retrieval length patterns will differ between the smoothing methods and may have a significant impact on retrieval performance. Two-stage smoothing combines both smoothing techniques and promotes the automatic estimation of μ and λ.

The rest of this article now empirically examines the behavior of the smoothing methods on different collections, using different types of queries and over a range of parameters settings. Our aim is to investigate the retrieval length patterns of these methods and how they compare to length patterns of relevant documents to understand the impact of the length of documents retrieved on performance.

4 Empirical analysis

In our experiments we considered two different collections and sets of topics. The first experimental tests were run with the last available TREC adhoc track data (TREC-8) (Voorhees and Harman 1999), whose data collection consists of approximately 2 GB of text. The second pool of tests were performed on the TREC HARD Track 2005 (Allan 2005), which consists of the AQUAINT news collection and HARD Robust TREC Topics. For the TREC-8 experiments there are 50 topics available (#401 to #450) and the AQUAINT experiments supply another 50 topics (non consecutive topics from #303 to #689). In both cases, long and short queries were obtained from the TREC topics using either all topic subfields or only the title subfield. Table 1 depicts some basic statistics for the collections. The length figures refer to document sizes after pre-processing. We applied the Porter stemmer (Porter 1980) but we did not remove stopwords. As a matter of fact, the effects of stopword removal should be better achieved by exploiting language modeling techniques (Zhai and Lafferty 2004), which can naturally cope with words having very different patterns of usage within the language. In this way, the experiments reported here are not biased by any artificial choice of an stoplist of a given size. Regarding statistical significance, we applied the *t*-test to compare the performance of runs as suggested in (Sanderson and Zobel 2005), and concluded that the differences were significant when $p < 0.05$.

For JM smoothing we ran tests with λ values from 0.1 to 0.9 in steps of 0.1. For DP smoothing we tested the following μ values: 10, 100, 1,000, 2,000, 3,000, 4,000, 5,000 and 10,000. For two-stage smoothing we applied the standard estimation methods to set μ and λ automatically. All these experiments were run using the Lemur (2002) toolkit.

The performance results (mean average precision, MAP) are reported in Table 2. Each row is labeled with the name of the collection and the type of queries (e.g. T8L means

Table 1 Basic statistics of the collections

Collection	# Docs	Avg. doc length (# terms)	Median doc length (# terms)
TREC-8	528155	482	329
AQUAINT	1033461	437	289

Table 2 The retrieval performance in terms of mean average precision for TREC8 (top) and AQUAINT (bottom), with Dirichlet smoothing, Jelinek–Mercer smoothing and two-stage smoothing, for short queries (S) and long queries (L)

TREC8									
DP(μ)									
	10	100	1k	2k	3k	4k	5k	10k	Two-stage
T8S	0.225	0.237	**0.251**	0.247	0.243	0.240	0.238	0.228	**0.252**
T8L	0.136	0.169	0.246	**0.252**	0.250	0.247	0.245	0.234	**0.249**
JM(λ)									
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
T8S	0.237	0.237	**0.237**	0.237	0.235	0.233	0.231	0.226	0.214
T8L	0.171	0.197	0.212	0.224	0.234	0.245	0.250	**0.255**	0.249
AQUAINT									
DP(μ)									
	10	100	1k	2k	3k	4k	5k	10k	Two-stage
AQS	0.142	0.163	0.198	**0.200**	0.198	0.195	0.192	0.176	**0.190**
AQL	0.119	0.152	0.211	**0.215**	0.212	0.208	0.205	0.191	**0.209**
JM(λ)									
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
AQS	0.148	0.150	0.151	0.151	**0.152**	0.151	0.150	0.147	0.140
AQL	0.156	0.169	0.176	0.181	0.184	0.186	**0.187**	0.183	0.170

TREC-8 with long queries) the best MAP value for each collection, smoothing strategy and type of query is marked in bold. Figure 1 shows graphically how MAP changes in DP and JM as smoothing increases. These results and the shapes of the curves are similar to those found in (Zhai and Lafferty 2004).

With JM smoothing the performance was sensitive to the λ setting for long queries with the best performance obtained for higher values of λ. For short queries, the performance was relatively invariant to the parameter setting. In DP smoothing the best performance was obtained around $\mu = 1{,}000$–$2{,}000$, more or less smoothing resulted in poorer performance.

Comparing the best performance attainable by DP and JM we found that DP smoothing was statistically significantly better than JM smoothing in the AQUAINT collection (with both types of queries), whilst the differences found in the TREC-8 collection were not regarded as statistically significant. Regarding two-stage smoothing, its automatic estimation methods work quite consistently. In TREC-8, there was no statistically significant difference between the two-stage runs and the best DP/JM runs. In the AQUAINT collection, the best DP run outperformed significantly the two-stage run for short queries but the two-stage run was comparable to DP for long queries. For both types of queries, the two-stage runs in AQUAINT produced retrieval performance which was significantly better than the best JM's performance.

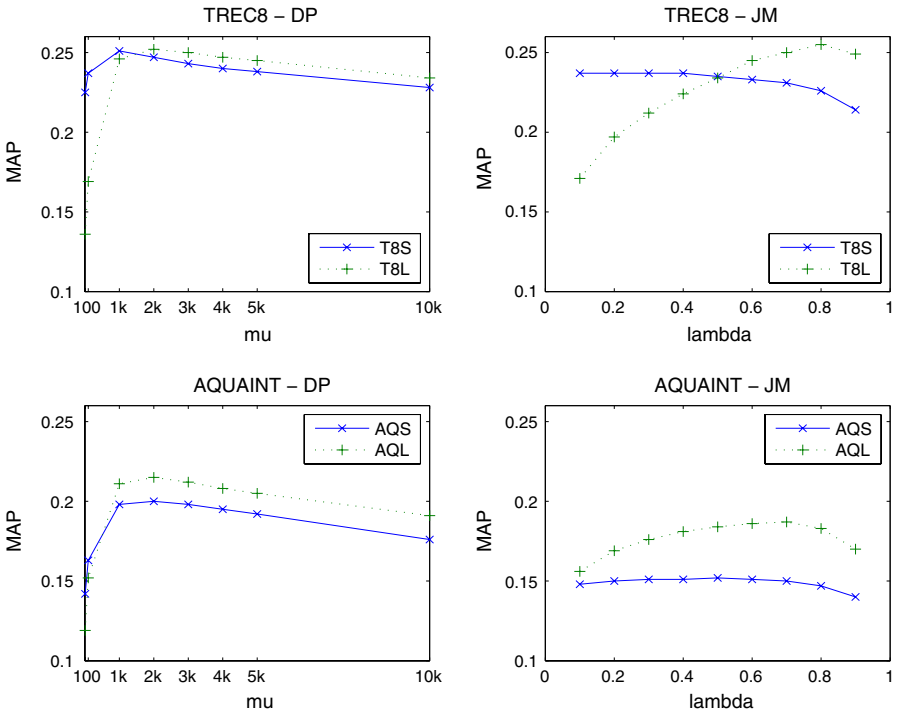

Fig. 1 The retrieval performance in terms of mean average precision for TREC8 (top) and AQUAINT (bottom), DP smoothing (left) and JM smoothing (right), for short queries (×) and long queries (+)

4.1 Document length retrieval trends

The main objective of this research is to understand the differences between these smoothing methods, not just in terms of performance, but in terms of retrieval functions and retrieval patterns.[4] By studying the interactions between smoothing levels, and query type on the length retrieval patterns we hope to gain deeper insights into how the retrieval functions operate and behave.

4.1.1 Methodology to analyze the length retrieval trends

To study the length retrieval trends, we adopt the methodology designed by Singhal et al. (1996b) when they proposed the pivoted length normalization method for the Vector-Space Model. Given a particular retrieval strategy, we can analyze the likelihood of relevance/retrieval for documents of all lengths by plotting these likelihoods against the document length. In this way, the relevance pattern and the retrieval pattern can be compared. To this aim, the document collection is first ordered by document length and documents are divided into equal-sizes chunks, which are called bins. Each bin is represented by the median length of the documents contained. To plot the relevant pattern, we simply take the

[4] For an extensive study comparing the performance of various smoothing methods including DP and JM, we refer the reader to the empirical study by Zhai and Lafferty (2004).

relevance judgments for the queries included in the experiment and count how many (query, relevant document) pairs have the document from each bin. Dividing these values by the total number of (query, relevant document) pairs, we obtain $P(D \in Bin_i | D \text{ is relevant})$. Similarly, a retrieval curve can be obtained by taking the top X retrieved documents for each query and doing the count across the (query, retrieved document) pairs. In (Singhal et al. 1996b) the top 1,000 retrieved documents were considered. In our experiments, we report results for the top 100, 500 and 1,000 because we want to analyze carefully the distribution of lengths in different parts of the rank. For instance, it might be the case that a particular smoothing strategy has a fair distribution of lengths in the top 1,000 but, on the other hand, it retrieved too many (or too few) long/short documents in the top 100.

Rather than bucketing, an alternative approach could have been adopted. In (Craswell et al. 2005), a novel methodology for studying query-independent features (such as document length) was proposed, which is based on kernel density estimation to obtain smoothed curves/estimates. Additionally, the likelihood measure is replaced by a log-odds function. While, this technique is valuable because it gives more information about the adjustments, the density estimates are less trustworthy when there are regions with few relevant examples. Consequently, we opted to follow Singhal's methodology.

The number of bins for our study is 529 for TREC8 and 1034 for AQUAINT and the number of (query, relevant document) pairs is 4,728 and 6,551, respectively. To have a clearer view of the plots, we followed the strategy taken in (Singhal et al. 1996b) to generate smoothed plots by representing a sequence of bins by a single point and connecting these points by a curve. The curves shown in the next pictures are computed for sequences of 20 bins.

In the Vector-Space Model (Singhal et al. 1996b), a retrieval function which retrieved documents such that the length of the documents returned closely matched the length of relevant documents resulted in excellent retrieval performance (in term of MAP). We are especially interested in studying these issues in the context of Language Modeling techniques and, particularly, checking the effects of smoothing on length retrieval trends. We therefore planned experiments for varying values of the smoothing parameters. In this way, we can observe how the retrieval patterns deviate from the relevant pattern as smoothing changes.

4.1.2 Experimental results (DP): length trends versus smoothing

The plots for Bayesian smoothing with Dirichlet priors are shown in Fig. 2. For simplicity, we only show the length trends for the AQUAINT collection. Actually, the experiments conducted against TREC8 produced equivalent trends. The figure depicts the length trends obtained with $\mu = 10$, 1,000, 2,000 and 10,000. In these graphs we observe some notable trends. First, DP smoothing tends to retrieve many short documents and few long documents for low smoothing values and this tendency is reversed as μ increases (high smoothing levels lead to retrieval of few short documents and more long documents). This happens regardless of the collection and type of queries (although the tendency is stronger with long queries). It is actually quite interesting to observe how the retrieval patterns swing around the relevance pattern as μ changes.

Given these plots, and coming back to Eq. 6, which summarizes the behavior of this smoothing strategy, we can conclude that the effect of the second addend (penalizing long documents for low μ values) prevails. Little smoothing means also more coordination level

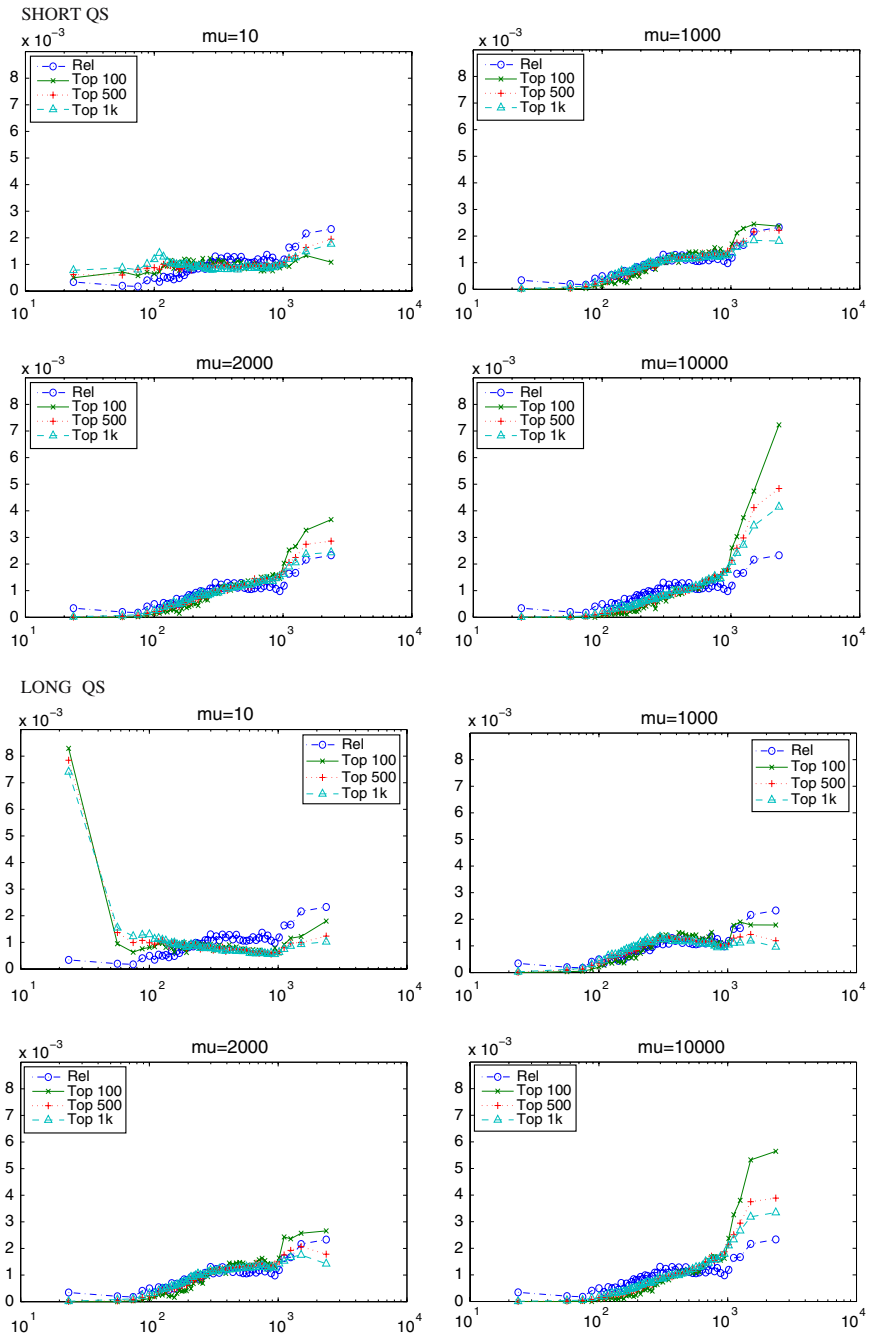

Fig. 2 Length retrieval/relevance trends for AQUAINT with Dirichlet smoothing, given short and long queries. The X axis displays the average of median bin length and the Y axis displays the average probability of retrieval/relevance. If the smoothing is too low, then the documents tend to be too short, and if there is too much smoothing, then the documents tend to be too long, with respect to the relevant pattern. When $\mu = 2{,}000$ the fit between the retrieval pattern and the relevant pattern are the closest

ranking in the sum (1st addend) but this effect is dismissed by $n \log \frac{\mu}{|d|+\mu}$. The plots show evidence to suggest that the same tendencies hold for short and long queries. The main difference we can observe between query types is that the preference for retrieving very short texts with low smoothing values is sharper for long queries. This can be explained because n is large and, thus, $n \log \frac{\mu}{|d|+\mu}$ gives more of a boost to short documents when μ is small. On the other hand, when the smoothing levels are high, the promotion of long documents seems to be more pronounced for short queries. Again, this is explained naturally looking at the second addend (n is short making the penalty for long texts smaller).

4.1.3 Experimental results (DP): optimal performance

Let us now pay attention to the smoothing levels yielding the best MAP performance. In AQUAINT, the DP method obtains the best performance with $\mu = 2{,}000$ (for both types of queries). If we observe the plots again, we can note that these smoothing values tend to be the ones in which the retrieval patterns are closer to the relevance pattern. That is, the maximum MAP performance tends to be found when the retrieval pattern retrieves documents at different lengths in a way which resembles the distribution of relevant documents in the collection. This shows graphically the common belief about the adequacy of Bayesian smoothing with Dirichlet priors for dealing with documents of different lengths.

4.1.4 Experimental results (JM): length trends versus smoothing

Figure 3 depicts the retrieval patterns for JM smoothing. Again, we only present the results achieved with the AQUAINT collection because there were no differences between the AQUAINT plots and the TREC8 plots. The figure includes plots for $\lambda = 0.1$, 0.4, 0.6 and 0.9. In both collections we observe patterns that are opposite to those found for DP; as the level of smoothing is increased JM smoothing tends to retrieve less long documents and more short texts. While for DP smoothing, more smoothing resulted in the retrieval of more long documents. This is not surprising because, as argued in Sect. 4.1, the query document similarity score derived from JM smoothing (Eq. 10) incorporates a factor $(1 - \lambda)/\lambda$ that multiplies the weights of the matching terms. Low λ's lead to a coordination level-like ranking where longer documents are ranked higher than short documents whereas high λ's promote the retrieval of shorter pieces of text.

Further, unlike DP smoothing, we observe for JM smoothing a distinct trend with short and long queries. With short queries the retrieval patterns deviate significantly from the relevance pattern for large document lengths. More specifically, the relevance plot indicates that more long documents should have been retrieved. This happens for all λ values (although the deviation increases as λ increases). With long queries and small λ's we retrieve too many long documents. As λ is increased it appears that the best fit between the retrieval pattern and relevance pattern is around $\lambda = 0.4$, but further smoothing (i.e. as λ approaches one) results in retrieving too few long texts.

4.1.5 Experimental results (JM): optimal performance

If we now analyze the smoothing level in which the best MAP performance is obtained and how the fit between the curves is at that level then we can observe a common trend for JM

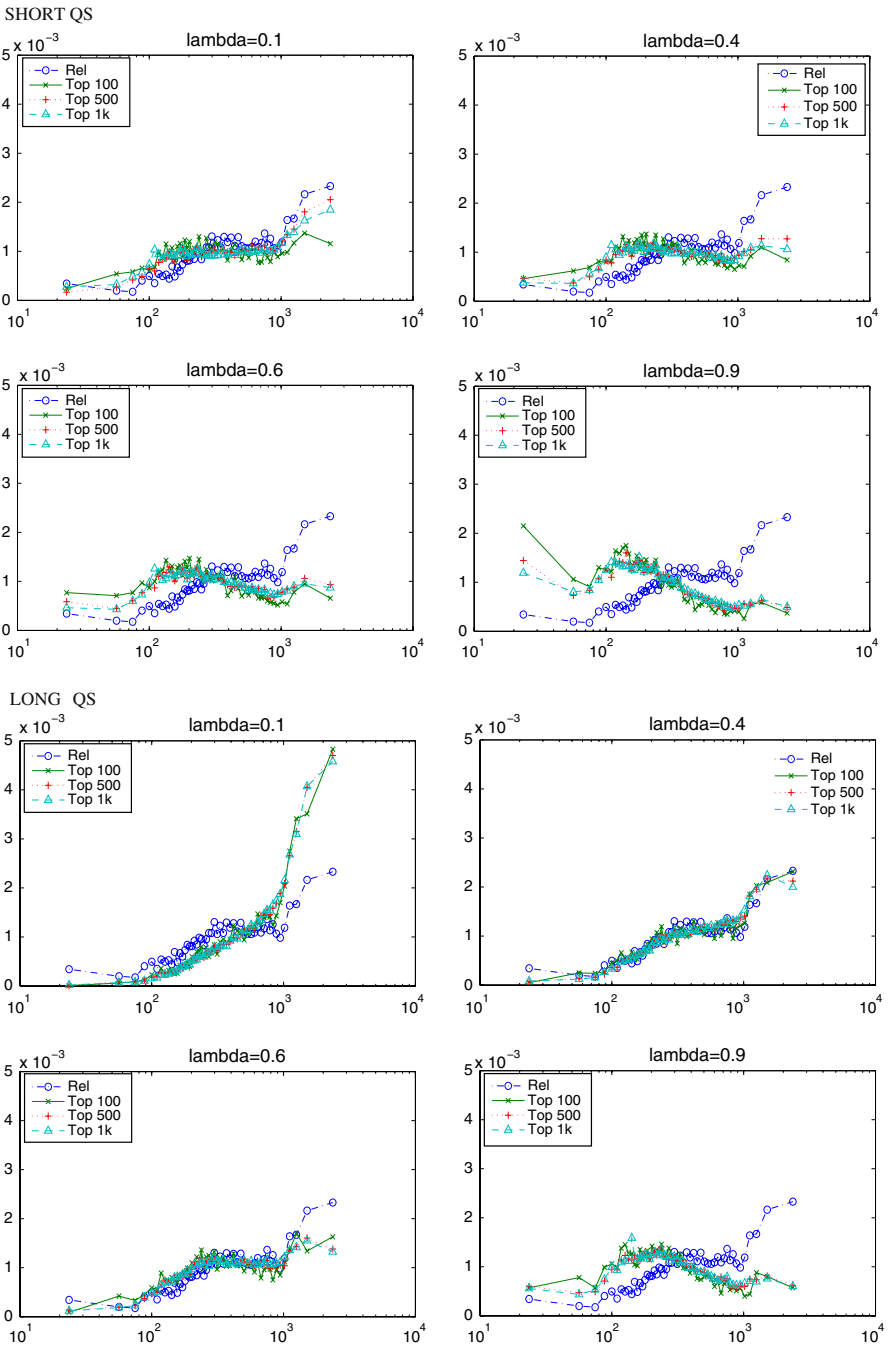

Fig. 3 Length retrieval/relevance trends for AQUAINT with Jelinek–Mercer smoothing, given short and long queries. The X axis displays the average of median bin length and the Y axis displays the average probability of retrieval/relevance

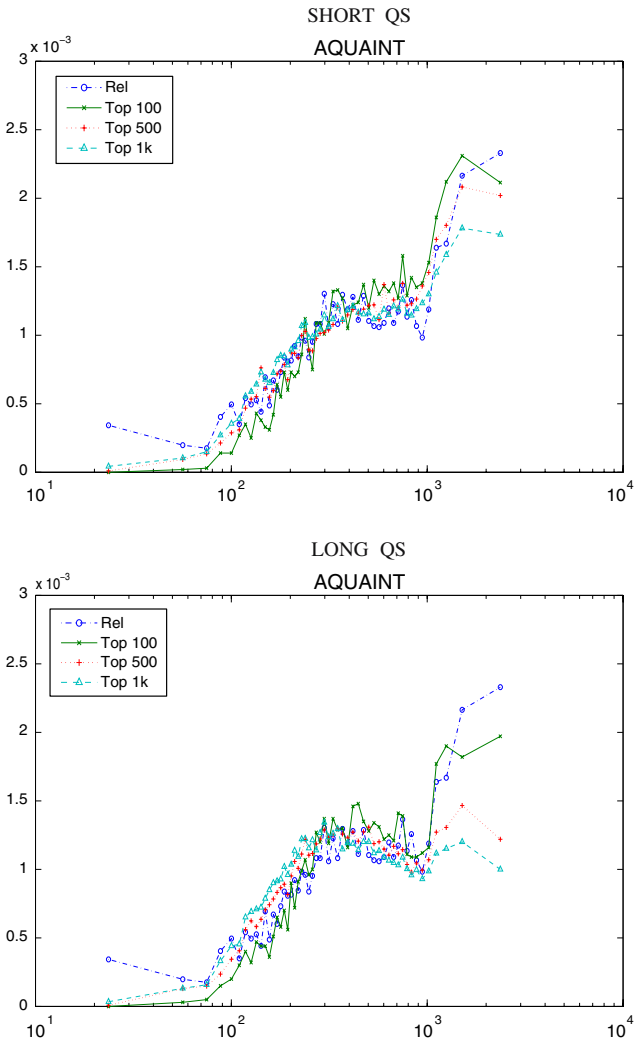

Fig. 4 Length retrieval/relevance trends for AQUAINT with two-stage smoothing, given short and long queries. The X axis displays the average of median bin length and the Y axis displays the average probability of retrieval/relevance

plots. The best performance does not coincide here with the best fitted plots. The best fits are always found at smoothing levels that are smaller than the ones required to achieve the highest performance. With short queries, the best MAP is achieved with $\lambda = 0.5$ (Table 2) whereas the best fitted plot is obtained with very low smoothing ($\lambda = 0.1$). The same happens for long queries (best MAP: $\lambda = 0.7$, best fitted plot: $\lambda = 0.4$). This explains the poor performance of JM smoothing w.r.t. DP smoothing. More specifically, these results illustrate that JM lacks a proper model of document length. As stated in Sect. 3.1, λ is simply a query modeling tool. This is evident here because the optimal λ values are strongly determined by the type of query. For instance, long queries required high smoothing to achieve optimal performance ($\lambda = 0.7$) but this smoothing level yielded a

poor retrieval model in terms of document length. That is, we need to set λ to a high value in order to explain the appearance of query terms with very different discriminative power but it does not account for the document length normalization problem.

4.1.6 Experimental results (two-stage smoothing)

The length retrieval trends for two-stage smoothing are presented in Fig. 4. The curves show that the automatic estimation applied by the two-stage model yields to length retrieval patterns with a good fit to the relevance pattern.

4.1.7 Fitness between retrieval and relevance patterns: L1 norm

To further analyze the correlation between retrieval performance and the fitness between the relevance and retrieval patterns, we computed the L1 norm between the relevance and retrieval patterns. This is a measure of Least Absolute Error between probability distributions:

$$L1 = \sum_{l} |P(Rel|l) - P(Ret|l)| \tag{13}$$

where $P(Rel|l)$ ($P(Ret|l)$) is the number of relevant (retrieved) documents whose length is l divided by the number of documents whose length is l in the collection. The smaller the L1 Norm the closer the two distributions are, with zero indicating that the distributions are identical. The larger the L1 Norm the further the two distributions are apart, with two being the furthest distance apart. Using the L1 Norm, while a relatively simple metric, is a sound means to determine the difference between distributions, and is a robust measure, despite the fact that no smoothing of the distributions is employed. We computed the L1 Norm values corresponding to retrieving the top 100, top 500 and top 1,000 documents but we report only the top 1,000 results because the other retrieval curves showed the same trends.

Figure 5 presents the results for all the collections, smoothing strategies and types of queries. One retrieval plot is shown for each smoothing method (DP or JM), collection and type of queries. The curve shows how the deviation between relevance and retrieval patterns changes as the smoothing parameter (μ or λ) increases. A vertical line indicates the smoothing level which produced the highest MAP given the smoothing strategy, collection and type of queries. For comparison purposes, the L1 Norm value obtained with two-stage smoothing is marked by a horizontal line.

The figure indicates very clearly that DP smoothing tends to achieve its best performance when the retrieval patterns get closer to the relevance pattern (i.e. the best MAP tends to be close to the minimum of the L1 graphs) and that there appears to be a correlation between the behavior and performance. Also, the graph explains the solid performance of the two-stage smoothing which tends to produce retrieval patterns which are also very close to the relevance pattern. However, there appears to be no such correspondence between performance and behavior under JM smoothing. JM smoothing is not accounting for the document length normalization issue adequately.

The fact that JM requires more smoothing to achieve the highest performance is naturally explained by the query modeling role of smoothing (Zhai and Lafferty 2004). As λ increases, we move away from coordination level ranking and the discriminative power of query words receives increasingly more weight. Longer queries are more dependant on idf

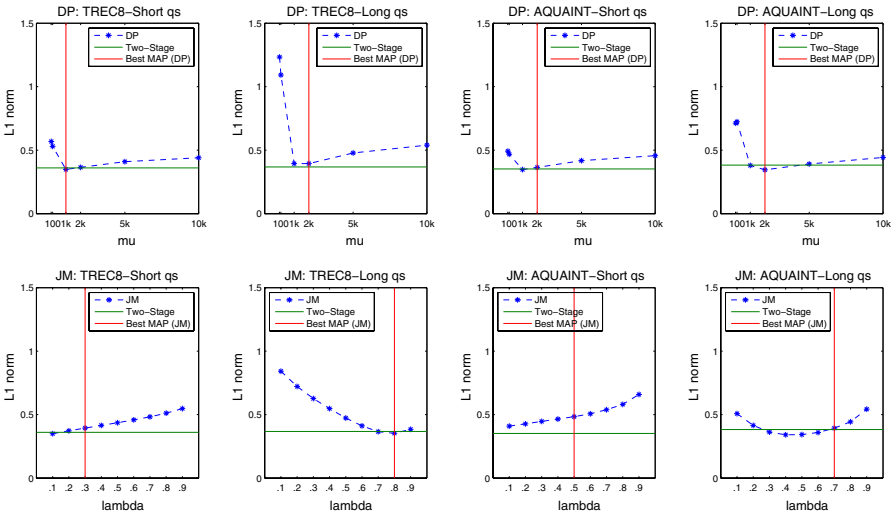

Fig. 5 L1 norm between relevance and retrieval length patterns at varying levels of smoothing: Results are shown for TREC8 (left) and AQUAINT (right), given DP smoothing (top) and JM smoothing (bottom), with short and long queries. A vertical line indicates the (DP/JM) smoothing level which achieves the highest performance. An horizontal line marks the L1 Norm value obtained with two-stage smoothing

than short queries because they tend to contain query terms with very different discriminative power. On the contrary, keyword-like queries are less dependant on idf because the quality of their terms is more homogeneous. This leads to a well-known result in JM smoothing: longer queries tend to require more smoothing than short queries. However, as shown in Fig. 5, the level of smoothing needed to yield optimal performance leads to length retrieval patterns deviating significantly from the relevance pattern. This indicates that the optimal smoothing level is strongly influenced by the query modeling role of smoothing, creating a conflict with the ideal document length retrieval trends. In contrast, DP smoothing conjugates naturally both roles, query modeling and document modeling, tending to achieve its highest performance when the retrieval and relevance length patterns are close. On the other hand, the automatic estimation process implemented by the two-stage model is nearly optimal, reporting deviation figures which tend to be equal to or less than the minimum deviation obtained across all DP/JM parameter settings.

An important conclusion from this analysis is that JM smoothing does not provide a good fit to the relevance length pattern and this explains the poorer performance of the model compared to DP and two-stage smoothing. Clearly, it is desirable to account for the dual roles of smoothing, and this leads to the questions: can the JM method be extended to cater for the length normalization problem? and can we achieve this by decoupling the two roles? In the next section, we explore how document priors can be used as a natural way to decouple the document length normalization issues from smoothing.

5 Document length based priors

In statistical language modeling for IR, document priors have been applied successfully in a number of tasks. For instance, in web search they have been found useful for entry page

search problems (Kraaij et al. 2002). In this context, different document priors can be defined using aspects such as the webpage length, url depth and link topology. These types of priors have been also applied to enhance informational and navigational web queries (Kamps 2005). The combination of a webpage's age and linkage for defining priors was also studied in (Hauff and Azzopardi 2005). In known-item search of e-mail messages, priors are also valuable and can be defined using aspects such as the the depth of the message in the thread (Ogilvie and Callan 2004).

The hypothesis that the probability of relevance is correlated with document length was empirically supported in (Kraaij and Westerveld, 2000).[5] The authors showed plots for several TREC adhoc and web collections and the correlation between relevance and length appears to be linear. Some length-based priors were subsequently defined to support web IR.

There is however little research on document priors in adhoc experiments. A notable exception can be found in (Hiemstra 2000), where a prior probability based on document length was defined and evaluated using three adhoc collections. This finding is important because it shows explicitly the connection between language modeling smoothing and tf/idf. However, this work focused on a single smoothing strategy (JM) and a single type of queries (short) and provided no comparison against other models. Also, the incorporation and analysis of document prior was not central to the objective in (Hiemstra 2000). Consequently, there is no analysis of the document retrieval trends derived and its interactions with smoothing.

However, document priors, which have demonstrated their utility in different IR problems, should be carefully studied for different smoothing strategies and types of queries to better understand their influence in the retrieval process. This is the objective pursued in this section. Document priors are natural components in the LM framework. Recall that, the probability of a document given a query is estimated as:

$$P(d|q) \stackrel{rank}{=} P(q|d) \cdot P(d) \tag{14}$$

When uniform priors are taken we end up with the popular query likelihood method, based on ranking documents using $P(q|d)$. Non-uniform priors lead to more evolved retrieval methods in which the effect of the query likelihood ($P(q|d)$) is combined with the prior effect. In our experiments we worked with the following document prior:

$$P(d) = \frac{|d|}{\sum_{d_k \in C} |d_k|} = \frac{\sum_w tf(w,d)}{\sum_{d_k \in C} \sum_w tf(w,d_k)} \tag{15}$$

This is the same prior utilized by Hiemstra in his experiments (Hiemstra 2000). It models an a priori preference for longer documents. As argued above, in many test collections there is evidence about the correlation between relevance and length (Kraaij and Westerveld 2000) (i.e. longer documents are more likely to be relevant). Since we also have shown that as the λ parameter increases, less long documents are retrieved by JM, then it seems appropriate that incorporating this prior would help offset this tendency. Of course, for other search tasks (e.g. a user searching for document abstracts) this document prior may be inappropriate. Here, since we are examining the task of ad hoc retrieval, we assume that the tendency is that the longer documents are more likely to be relevant.

[5] While this correlation has been empirical supported this does not fully validate the hypothesis. For instance, this correlation may be an artifact of the experimental design. Here we assume that this hypothesis holds.

In other cases, the prior distribution would need to be adjusted accordingly, to reflect the nature of the task at hand.[6]

The prior defined above helps to explain the distribution of different lengths across the collection. This leads to a model in which document length is modeled independently and, in this way, the smoothing strategy applied is not the only component of the scoring function responsible for shaping the final retrieval pattern. As shown in the previous sections, with uniform priors the optimal smoothing level is influenced by document length. This creates a tension between document length and other modeling roles of smoothing (such as idf). In contrast, we expect that non-uniform length-based priors allow smoothing to be focused on estimating document models (without document length as the main conditioning factor). This means that length-based document priors might be a natural way to decouple document modeling and query modeling, as an alternative to two-stage smoothing.

The non-uniform document prior sketched above was applied for JM smoothing and DP smoothing. The subsequent retrieval formulas are:

$$\log P(q|d) \overset{rank}{=} \sum_{i:tf(q_i,d)>0} \log\left(1+\frac{(1-\lambda)}{\lambda}\cdot\frac{tf(q_i,d)}{|d|\cdot P(q_i|C)}\right)+\log\frac{|d|}{\sum_{d_k\in C}|d_k|} \tag{16}$$

$$\log P(q|d) \overset{rank}{=} \sum_{i:tf(q_i,d)>0} \log\left(1+\frac{tf(q_i,d)}{\mu P(q_i|C)}\right)+n\log\frac{\mu}{|d|+\mu}+\log\frac{|d|}{\sum_{d_k\in C}|d_k|} \tag{17}$$

where, the retrieval function of each method has a document dependent addendum included in the sum. Note that this last addendum is document-dependent but query-independent. Using this retrieval function, we repeated the set of experiments in the previous section. The experimental results are reported in Table 3 and shown in Fig. 6. Using these priors we found that there was no statistically significant difference between the best performance attainable by each smoothing technique. This contrasts with the results achieved with uniform priors, where DP smoothing was better than JM smoothing.

For comparison purposes the graph in Fig. 6 shows also the results obtained with uniform priors. The runs with length-based priors are assigned labels whose last letter is a P (e.g. T8SP: TREC8 experiments with short queries and length-based prior).

To further evaluate the effects of the priors on each smoothing technique, Table 4 reports the performance attained by two-stage smoothing and the best performance attainable by DP (JM) with uniform priors against the best performance attainable by DP (JM) with non-uniform priors (marked with an star when the difference is statistically significant). The document priors test was detrimental to the performance of DP smoothing. Performance tends to fall dramatically when the length-based priors are applied. This is because the length of documents retrieved tended to be longer than relevant documents (due to the prior). However, we can see how the DP smoothing methods tries to compensate for this bias, and by adopting a lower μ, the effect can be mitigated. And this is when the best performance attainable with the length-based priors is obtained. Nonetheless, this is usually significantly worse than the best performance when uniform priors are used. Also, DP smoothing with uniform priors appears to be more stable w.r.t. the amount of smoothing (in Fig. 6, the shapes of the curves associated to uniform priors are smoother than the shapes of the curves associated to non-uniform priors). These results confirm that

[6] A whole study could be based on identifying the best document length prior. Here, our aim is to capture the intuitive of such a prior and to quantify its effect on the retrieval behavior.

Table 3 The retrieval performance in terms of mean average precision for TREC8 (top) and AQUAINT (bottom), with Dirichlet smoothing and Jelinek–Mercer smoothing, for short queries (S) and long queries (L), with non-uniform priors (P)

TREC8									
DP(μ)									
	10	100	1k	2k	3k	4k	5k	10k	
T8SP	0.250	0.**252**	0.225	0.209	0.198	0.190	0.184	0.164	
T8LP	0.146	0.180	**0.245**	**0.245**	0.240	0.235	0.233	0.216	
JM(λ)									
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
T8SP	0.247	0.247	0.247	**0.248**	0.248	0.247	0.245	0.242	0.227
T8LP	0.165	0.191	0.209	0.222	0.234	0.243	0.253	0.261	**0.263**
AQUAINT									
DP(μ)									
	10	100	1k	2k	3k	4k	5k	10k	
AQSP	0.171	**0.184**	0.176	0.160	0.149	0.141	0.134	0.109	
AQLP	0.130	0.163	**0.209**	0.204	0.196	0.190	0.184	0.164	
JM(λ)									
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
AQSP	0.176	0.178	0.181	0.182	0.185	0.187	0.188	**0.189**	0.185
AQLP	0.154	0.170	0.179	0.187	0.192	0.196	0.199	**0.200**	0.197

DP smoothing is a natural way to deal with documents of varying length without requiring any further adjustment. As a consequence, there is no reason to prefer the DP smoothing model with the length-based priors.

With JM smoothing the situation is the opposite because JM tends to retrieve documents that are shorter than relevant documents. Non-uniform priors are beneficial: the best performance attainable with these priors is always better than the best performance attainable with uniform priors and the difference is statistically significant in three out of the four cases. Comparing the curves corresponding to short and long queries in Fig. 6, it seems that the benefits from the non-uniform priors are more prominent with short queries. With long queries the plots are closer but, still, the runs with length-based priors are consistently better than the runs with uniform priors. This suggests that the most natural way to enhance the performance of JM models consists of applying a length-based prior to model the distribution of document lengths in the collection. In this way, the λ parameter is focused on query modelling and the document prior handles document length normalization issue.

Given that the prior introduces a preference for longer documents, it is reasonable that DP tends to get its highest performance for lower μ values. As argued in Sect. 4.1, with DP smoothing, the higher μ the more long documents retrieved. Since we have now an a priori preference for long texts, there is no need for high smoothing to retrieve longer documents.

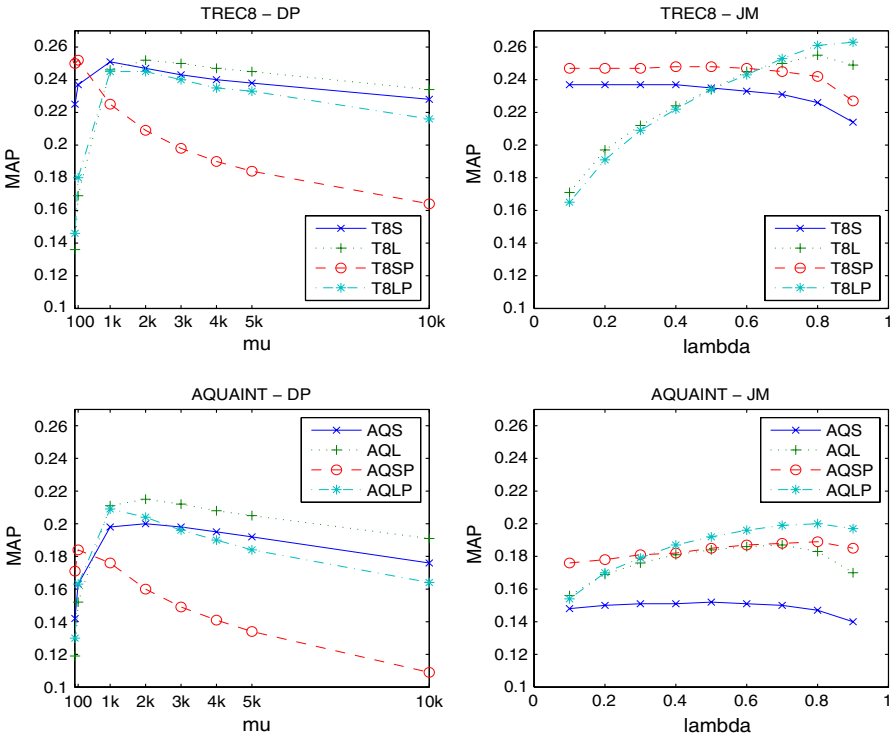

Fig. 6 The retrieval performance in terms of mean average precision for TREC8 (top) and AQUAINT (bottom), DP smoothing (left) and JM smoothing (right), for short queries and uniform priors (×), long queries and uniform priors (+), short queries and non-uniform priors (◯) and long queries and non-uniform priors (*)

Table 4 The retrieval performance in terms of mean average precision for TREC8 (top) and AQUAINT (bottom), with Dirichlet smoothing, Jelinek–Mercer smoothing and two-stage smoothing, for short queries (S) and long queries (L), with uniform and non-uniform priors

TREC8					
	DP		JM		
	Uniform priors	Non-uniform priors	Uniform priors	Non-uniform priors	Two-stage
T8S	0.251	0.252	0.237	0.248	0.252
T8L	0.252	0.245*	0.255	0.263*	0.249
AQUAINT					
	DP		JM		
	Uniform priors	Non-uniform priors	Uniform priors	Non-uniform priors	Two-stage
AQS	0.200	0.184*	0.152	0.189*	0.190
AQL	0.215	0.209*	0.187	0.200*	0.209

A non-uniform prior run is marked with an asterisk when the difference against the corresponding uniform prior run is statistically significant

Table 5 Statistical significance tests between the best runs of Dirichlet smoothing and Jelinek–Mercer smoothing for TREC8 (left) and AQUAINT (right), given short (S) and long queries (L)

TREC8			AQUAINT		
	DP Best run	JM Best run		DP Best run	JM Best run
T8S	0.252	0.248	AQS	0.200	0.189*
T8L	0.252	0.263	AQL	0.215	0.200*

The figures refer to mean average precision. A Jelinek–Mercer best run is marked with an asterisk when the difference against the corresponding Dirichlet best run is statistically significant

On the other hand, JM tends to achieve its best performance with greater λ values. Again, this is intuitive because high λ's yield less long documents retrieved and, thus, starting with a prior promoting long documents we need more smoothing to get a good balance across different lengths.

To compare the best results attainable by each smoothing method (either with uniform or with non-uniform priors) Table 5 reports the results of the statistical significance tests between the respective best runs. These tests show that there is not statistical significant difference between DP and JM for TREC8, while the difference between DP and JM is statistical significant for AQUAINT. These results also show that JM is promising with length-based priors (its performance is significantly improved with such priors and, for some collections, it works at least as well as DP). Nevertheless, it is still unclear whether or not JM can become an state-of-the-art smoothing method for document retrieval. To further investigate into this issue, a complete study on different length-based priors (including how to get prior estimates fitted to the collection data) needs to be conducted.

In order to analyze further these results, new plots showing how the retrieval (with non-uniform priors) and relevant patterns evolve against document length are presented in Figs. 7 and 8. Again, the graphs obtained for TREC8 are not shown because they are similar to the AQUAINT graphs reported here. In DP smoothing, the optimal μ values correspond to plots where the retrieval and relevance patterns are close. In Fig. 6 we could observe that DP's MAP values tend to fall for μ values greater than 1,000 (especially for short queries). This is nicely explained by Fig. 7 because, with $\mu \geq 1{,}000$, we get plots where the retrieval pattern deviates significantly from the relevance pattern. On the other hand, Fig. 8 explains clearly why JM improves its performance after the introduction of priors. The retrieval patterns show a much better fit to the relevance pattern (recall that, in Fig. 3, we had found significant deviations between the relevance and retrieval patterns for the best MAP run with uniform priors). This happens for all collections and types of queries but it is perhaps more evident for short queries (with uniform priors the fitness for the short queries was worse than the fitness for the long queries).

Again, we computed the L1 norm between the retrieval and relevance distributions and the results are shown in Fig. 9. The graph explains why JM gets important benefits from the length-based priors. JM now returns documents whose distribution of lengths is much closer from the distribution of lengths in relevant documents. With this smoothing strategy, there is now a correspondence between the best MAP and the lowest L1. This did not happen in the uniform prior experiments (Fig. 5). DP smoothing shows also here a good fit (because it compensates the a priori preference for long documents with smaller μs) and, therefore, the retrieval patters are still close from the relevance pattern. Anyway, DP smoothing had already shown good fits with uniform priors.

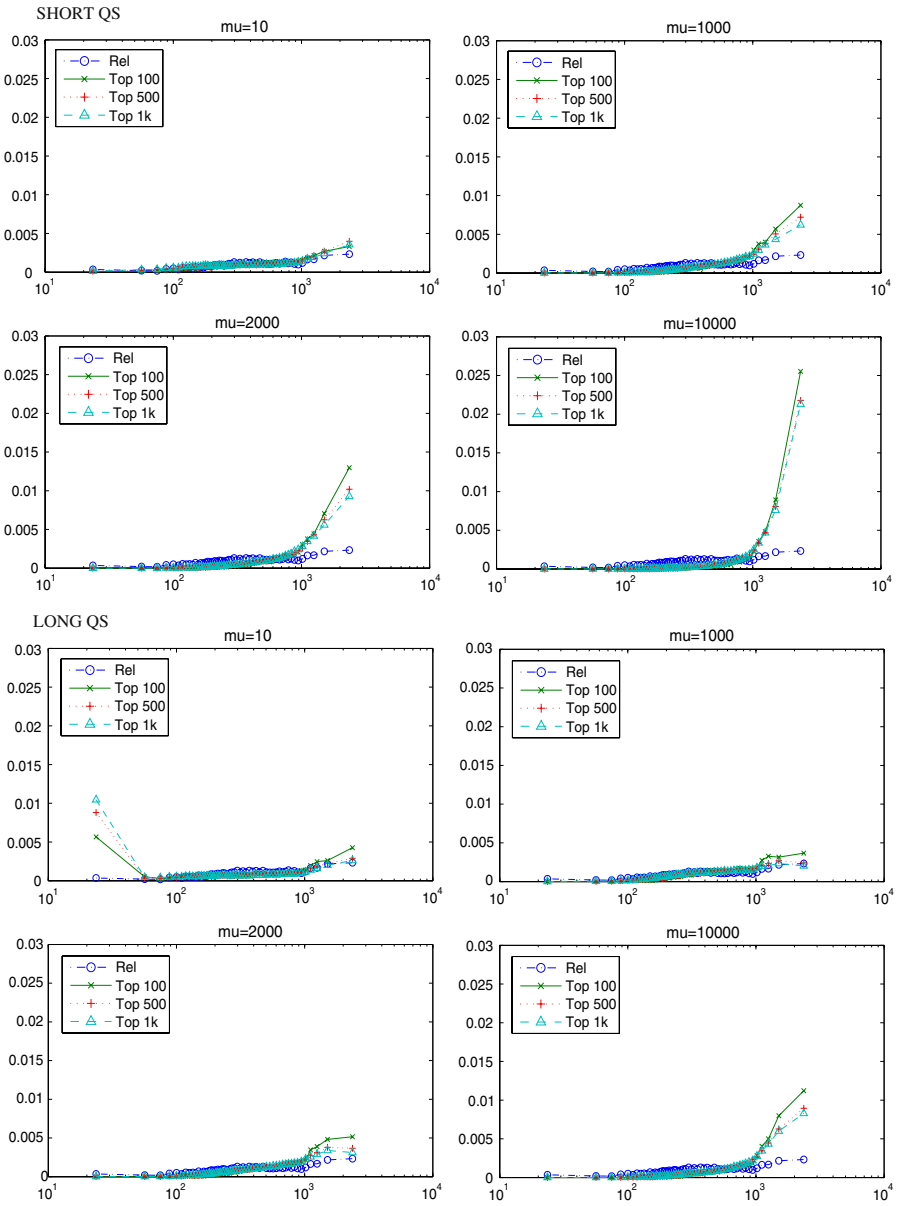

Fig. 7 Length retrieval/relevance trends for AQUAINT with Dirichlet smoothing and non-uniform document priors, and given short and long queries. The X axis displays the average of median bin length and the Y axis displays the average probability of retrieval/relevance

6 Discussion

We first focus our discussion on the results found with uniform priors, before discussing the results obtained using non-uniform priors. In particular, we retake the research questions raised in the introduction.

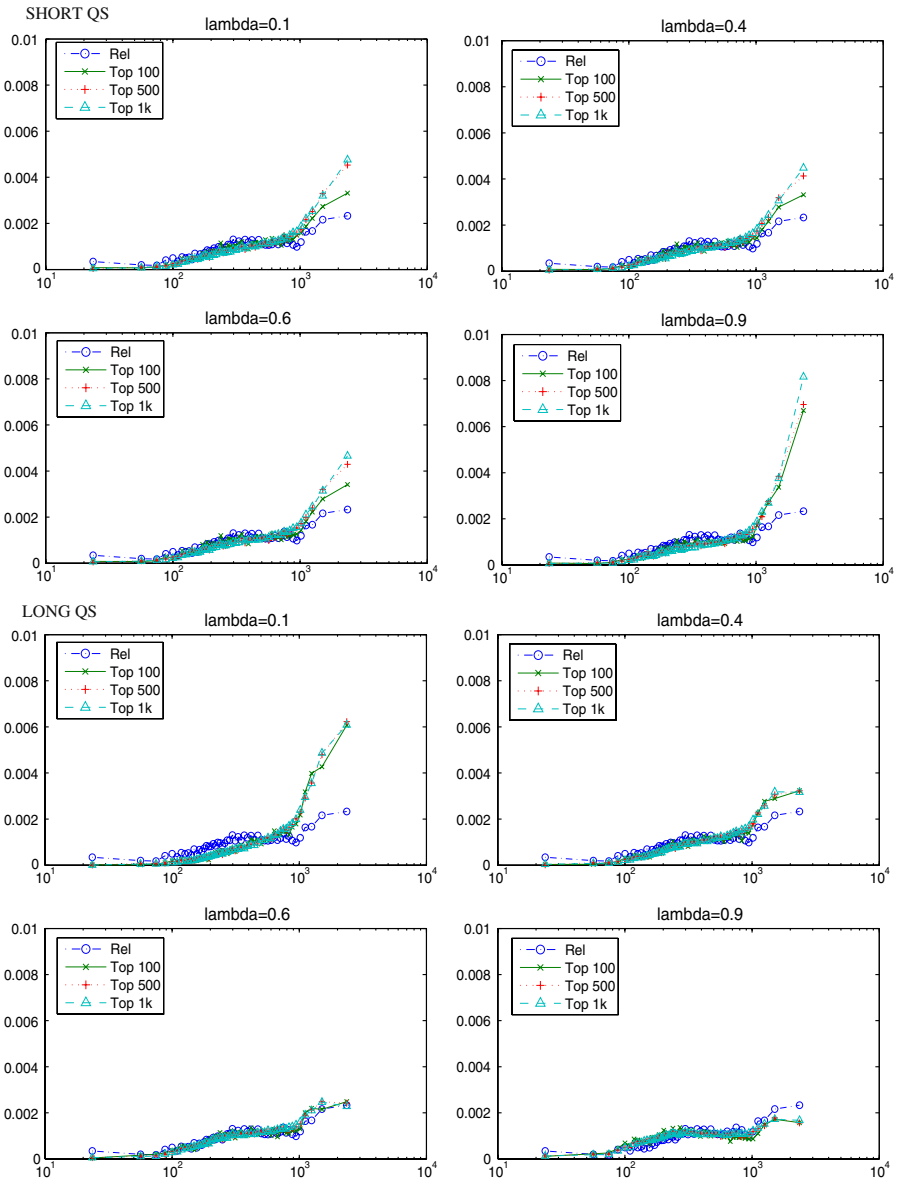

Fig. 8 Length retrieval/relevance trends for AQUAINT with Jelinek–Mercer smoothing and non-uniform document priors, and given short and long queries. The X axis displays the average of median bin length and the Y axis displays the average probability of retrieval/relevance

What roles do the retrieval functions play? The JM's retrieval function does not incorporate any document modeling role. The smoothing parameter only acts as a query modeling tool (adjusting the idf effect on query terms). On the contrary, DP's smoothing parameter acts as a query modeling tool and as document modeling tool. In particular, we argue that the document modeling role is restricted to be a document length corrector and accounts for the problems of document length normalization.

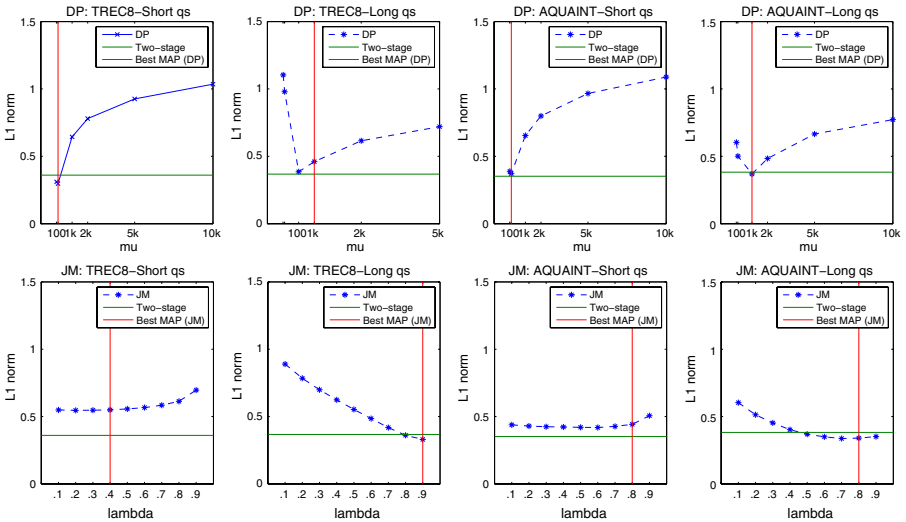

Fig. 9 L1 norm between relevance and retrieval length patterns (with non-uniform priors) at varying levels of smoothing: Results are shown for TREC8 (left) and AQUAINT (right), given DP smoothing (top) and JM smoothing (bottom), with short and long queries. A vertical line indicates the (DP/JM) smoothing level which achieves the highest performance. A horizontal line marks the L1 Norm value obtained with two-stage smoothing

In two-stage smoothing, each role is explicitly modeled by the different stages of smoothing. However, the two parameters used to control the influence of each role may either re-enforce or conflict with each other. More specifically, once you set μ, the estimation of λ not only influences the matching term weights (first sum in Eq. 12) but it also determines the document length correction (second sum in Eq. 12). This means that document and query modeling roles are not completely de-coupled.

How do the retrieval functions affect the behavior? (and, in particular, how this affects document length normalization issues). With JM, as the level of smoothing is increased less long documents (and more short texts) are retrieved. With DP, the tendency is the opposite. While the JM's tendency was expected (given the JM's retrieval function), the DP's tendency was not that clear, given the DP's retrieval formula (see Eq. 6). If μ is very low, then longer documents are favored, as μ is increased shorter documents are favored, but further smoothing leads to longer documents again being favored. While previous work has already pointed out that μ appears in different components of the retrieval formula and it was not evident whether or not an increment in μ would lead to the retrieval of shorter or longer documents (Zhai and Lafferty 2002, 2004), we have shown how the retrieval model behaves clarifying this case.

How does the behavior of the models affect their performance? Why is DP better than JM? For DP smoothing there appears to be a correlation between performance and the fitness of the retrieval and relevance length patterns. This smoothing strategy tends to produce the best retrieval performance when the document length retrieval pattern resembles the relevance pattern. DP's retrieval formula appears to naturally balance the query modeling and document modeling roles, because the smoothing performs a trade-off between the first and second addendum (which reflects the two roles). When the balance is struck, this corresponds to the best performance of the model (i.e. the retrieval pattern matches the relevance pattern). However, JM smoothing is not accounting for the document length normalization issue adequately; it is only accounting for the query role (i.e. the reliance only on the first

addendum which handles the query term variation). The best parameter setting for JM is strongly influenced by the type of query, leading to a retrieval pattern deviating significantly from the relevance pattern. This leads to poorer retrieval performance because the relevant documents tend to be longer than the documents retrieved; and the smoothing can not compensate adequately for this. With DP, the best parameter setting conjugates naturally query modeling and document modeling leading to a good fit between retrieval and relevance patterns and, consequently, good retrieval performance.

For the two-stage model, the automatic parameter estimation method selects parameters that result in retrieval patterns that closely fit the relevance pattern. This makes the two stage model very attractive because this results in performance which is close to or equal to the best performance found given DP and JM.

What is the influence of the length based prior? The experiments reported in this article demonstrate that JM smoothing's performance can be significantly improved with priors based on document length (over uniform priors). In both collections, JM performs significantly better with the non-uniform priors and, these priors almost make that JM becomes as competitive as DP). Intuitively, once the distribution of different lengths is adjusted by the prior probability, JM smoothing can focus on the query estimation role. This yields substantial benefits in terms of performance. In the two stage model, the first stage of smoothing implicitly performs this adjustment, but the prior-based method is an alternative approach.

Here we showed that non-uniform priors based on length act as as a fundamental tool to explain the distribution of different lengths in the collection. By relying on the prior to act as a document length corrector, then the query modeling role of smoothing (idf-like behavior) can be treated independently from document-specific features (document length). Since the distribution of lengths is modeled by the prior, smoothing can be focused on the query modeling role. In other words, the length-based priors allow for the removal of the tension between the query modeling and document modeling roles of smoothing. In (Zhai and Lafferty 2002) this problem was addressed by introducing a two-stage smoothing process. Our results suggests that non-uniform priors are also a natural way to handle this problem. An interesting line of future research motivated by this study would be an examination of different (length based) document priors and how they affect the behavior and performance of the model. Another interesting issue is to study methods to set automatically a given model's parameters in order to shape a particular retrieval pattern. This can lead to the development of novel estimation mechanisms focused on producing retrieval sets whose length distribution is close to the length distribution of the relevant documents (see (Azzopardi and Losada 2007) for an example).

7 Conclusions

In this article we have studied different Language Modeling smoothing strategies from a document length retrieval perspective and have shown through the theoretical and empirical analysis conducted important insights into the characteristics of smoothing and its implications in retrieval performance. The main conclusions of our study are:

- The document length retrieval pattern is also of major importance in Language Modeling for Information Retrieval. Large deviations between retrieval and relevance length patterns are often indicative of non-optimality of the retrieval method.
- Dirichlet Prior smoothing tends to retrieve many short documents (and few long documents) for low smoothing values and this tendency is reversed as the smoothing

parameter μ increases. From a theoretical perspective, the behavior of this smoothing technique against document length was previously unknown (the retrieval function associated to this smoothing strategy has a penalty for long documents that grows as μ decreases but, on the other hand, low μs favor long documents because there is more tendency to coordination level ranking in the sum across matching terms). The empirical analysis reported here demonstrates that low smoothing leads to the retrieval of more short documents, meaning that the penalty for long documents prevails. This clarifies the retrieval behavior of this smoothing technique against document length.

- Dirichlet Prior smoothing achieves its highest performance when the length retrieval pattern is closest to the length relevance pattern. This demonstrates that this smoothing techniques treats the appearance of different lengths in the collection appropriately.
- With JM smoothing, more long documents are retrieved as λ decreases and, conversely, more short documents are retrieved as λ grows. However, unlike Dirichlet Prior smoothing, the best performing JM runs still show a significant deviation between retrieval and relevance length patterns.
- The automatic estimation procedures inherent to two-stage smoothing yield retrieval patterns very close from the relevance pattern. This explains the excellent performance generally obtained with this smoothing strategy.
- The length of the retrieved documents by JM Smoothing tends to be shorter than relevant documents; however by addressing this with a length based document prior JM Smoothing can achieve significantly better performance (in some cases equivalent to Dirichlet smoothing's performance).

The utilization of non-uniform document priors based on document length appears as a natural alternative to two-stage smoothing to decouple the query and document modeling roles of smoothing. In particular, this type of priors proved useful to increase significantly the performance of Language Models based on JM smoothing. With Dirichlet Prior smoothing, these non-uniform priors are useless because this smoothing strategy with uniform priors retrieves properly documents of different length. On the contrary, JM with uniform priors does not retrieve fairly documents of different length because smoothing is focused on query modeling (explaining the appearance of terms with different discriminative power in the queries). The inclusion of length-based priors in JM's models is therefore a natural way to handle the document length retrieval problem.

This finding along with the fact that there is a strong correlation between performance and the fitness of the retrieval and relevance length patterns suggests a number of possible avenues for further research and analysis. For instance, an examination of different length-based priors and their influence on the behavior and performance of the models. Aspects such as how the parameters of the model can be estimated given the behavior of the model, how the collection pattern relates to the retrieval and relevant pattern and what data could be used to estimate the relevance pattern (past relevance judgments, assessments, click-through data, collection, etc.) are also interesting avenues to conduct further research in to which we have been pursuing (see (Azzopardi and Losada 2007)).

Acknowledgements The authors would like to thank Dr. Mark Baillie and the anonymous reviewers for their useful comments and suggestions which have been incorporated into this article. David E. Losada thanks the support obtained from projects TIN2005-08521-C02-01 (*Ministerio de Educación y Ciencia*), PGIDIT06PXIC206023PN and 07SIN005206PR (*Xunta de Galicia*). David E. Losada is funded on a "Ramón y Cajal" research fellowship, whose funds come from *Ministerio de Educación y Ciencia* and the FEDER program.

References

Allan, J. (2005). HARD track overview in TREC 2005 high accuracy retrieval from documents. In *Proceedings of the 14th Text Retrieval Conference (TREC 2005)*.

Amati, G. (2003). Divergence from randomness. Ph.D. thesis, Department of Computer Science, University of Glasgow.

Amati, G., & van Rijsbergen, C. (2002). Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems (TOIS), 20*(4), 357–389.

Azzopardi, L. (2005) Incorporating context into the language modeling for ad hoc information retrieval. Ph.D. thesis, University of Paisley, Glasgow, UK.

Azzopardi, L., & Losada, D. E. (2007). Fairly retrieving documents of all lengths. In *Proceedings of the First International Conference in Theory of Information Retrieval (ICTIR 2007)* (pp. 65–76).

Chen, S. F., & Goodman, J. (1998). An empirical study of smoothing techniques for language modeling. Technical report TR-10-98, Harvard University.

Chowdhury, A., McCabe, M. C., Grossman, D., & Frieder, O. (2002). Document normalization revisited. In *Proceedings of the 25th ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 381–382). New York, NY: ACM Press.

Craswell, N., Robertson, S., Zaragoza, H., & Taylor, M. (2005). Relevance weighting for query independent evidence. In *Proceedings of the 28th ACM Conference on Research and Development in Information Retrieval, SIGIR'05* (pp. 416–423). Salvador, Brazil.

Harman, D. (2005). *TREC: Experiment and Evaluation in Information Retrieval*, Chap. The TREC AdHoc Experiments, pp. 79–97. The MIT Press.

Hauff, C., & Azzopardi, L. (2005). Age dependent document priors in link structure analysis. In D. Losada & J. M. Fernandez-Luna (Eds.), *Proceedings of the 27th European Conference on Information Retrieval Research, ECIR'2005* (pp. 552–554). Santiago de Compostela, Spain: Springer Verlag, LNCS 3408.

Hiemstra, D. (1998). A linguistically motivated probabilistic model of information retrieval. In C. Nicolaou & C. Stephanidis (Eds.), *Lecture Notes in Computer Science: Research and Advanced Technology for Digital Libraries*, Vol. 1513 (pp. 569–584).

Hiemstra, D. (2000). A probabilistic justification for using tf x idf term weighting in information retrieval. *International Journal of Digital Libraries, 3*, 131–139.

Jelinek, F., & Mercer, R. (1980). Interpolated estimation of Markov source parameters from sparse data. In *Proceedings of the Workshop on Pattern Recognition in Practice*. Amsterdam.

Kamps, J. (2005). Web-centric language models. In *Proceedings of the ACM Conference on Information and Knowledge Management (CIKM)*.

Kraaij, W., & Westerveld, T. (2000). TNO/UT at TREC-9: How different are web documents. In *Proceedings of the TREC-9, the 9th Text Retrieval Conference*. Gaithersburg, USA.

Kraaij, W., Westerveld, T., & Hiemstra, D. (2002). The importance of prior probabilities for entry page search. In *Proceedings of the 25th ACM Conference on Research and Development in Information Retrieval, SIGIR'02* (pp. 27–34). Tampere, Finland.

Lemur. (2002). The Lemur toolkit. http://www.lemurproject.org

Mackay, D., & Peto, L. (1995). A hierarchical Dirichlet language model. *Natural Language Engineering, 1*(3), 1–19.

Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. The MIT Press.

Miller, D., Leek, T., & Schwartz, R. (1999). A hidden markov movel information retrieval system. In *Proceedings of the SIGIR-99, the 22nd ACM Conference on Research and Development in Information Retrieval* (pp. 214–221). Berkeley.

Ogilvie, P., & Callan, J. (2004). Experiments with language models for known-item finding of e-mail messages. In *Proceedings of the 14th Text Retrieval Conference, TREC-2004*.

Ponte, J., & Croft, W. B. (1998). A language modeling approach to information retrieval. In *Proceedings of the 21st ACM Conference on Research and Development in Information Retrieval, SIGIR'98* (pp. 275–281). Melbourne, Australia.

Porter, M. (1980). An algorithm for suffix stripping. *Program, 14*(3), 130–137.

Robertson, S., & Walker, S. (1994). Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the SIGIR-94, the 17th ACM Conference on Research and Development in Information Retrieval* (pp. 232–241). Dublin, Ireland.

Robertson, S., Walker, S., Jones, S., Hancock Beaulieu, M., & Gatford, M. (1995). Okapi at TREC-3. In Harman, D. (Ed.), *Proceedings of the TREC-3, the 3rd Text Retrieval Conference* (pp. 109–127). NIST.

Sanderson, M., & Zobel, J. (2005). Information retrieval system evaluation: Effort, sensitivity, and reliability. In *Proceedings of the 28th ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 162–169).

Singhal, A., Buckley, C., & Mitra, M. (1996a). Pivoted document length normalization. In *Proceedings of the 19th ACM SIGIR conference on Research and Development in Information Retrieval* (pp. 21–29).

Singhal, A., Buckley, C., & Mitra, M. (1996b). Pivoted document length normalization. In *Proceedings of the SIGIR-96, the 19th ACM Conference on Research and Development in Information Retrieval* (pp. 21–29). Zurich, Switzerland.

Voorhees, E., & Harman, D. (1999). Overview of the eight text retrieval conference. In *Proceedings of the TREC-8, the 8th text retrieval conference*.

Zhai, C., & Lafferty, J. (2001). A study of smoothing methods for language models applied to adhoc information retrieval. In *Proceedings of the 24th ACM Conference on Research and Development in Information Retrieval, SIGIR'01* (pp. 334–342). New Orleans, USA.

Zhai, C., & Lafferty, J. (2002). Two-stage language models for information retrieval. In *Proceedings of the 25th ACM Conference on Research and Development in Information Retrieval, SIGIR'02* (pp. 49–56). Tampere, Finland.

Zhai, C., & Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems, 22*(2), 179–214.