Assessing Multi-variate Bernoulli models for Information Retrieval

DAVID E. LOSADA
Grupo de Sistemas Inteligentes, Departamento de Electrónica y Computación
Universidad de Santiago de Compostela, Spain
dlosada@dec.usc.es
and
LEIF AZZOPARDI
Department of Computing Science
University of Glasgow, Scotland
leif@dcs.gla.ac.uk

Although the seminal proposal to introduce Language Modeling in Information Retrieval was based on a multi-variate Bernoulli model, the predominant modeling approach is now centered on Multinomial models. Language modeling for retrieval based on multi-variate Bernoulli distributions is seen to be inefficient and believed to be less effective than the Multinomial model. In this paper, we examine the multi-variate Bernoulli model with respect to its successor and examine its role in future retrieval systems. In the context of Bayesian Learning both modeling approaches are described, contrasted and compared theoretically and computationally. We show that the query likelihood following a multi-variate Bernoulli distribution introduces interesting retrieval features which may be useful for specific retrieval tasks such as sentence retrieval. Then, we address the efficiency aspect and show that algorithms can be designed to perform retrieval efficiently for multi-variate Bernoulli models, before performing an empirical comparison to study the behaviorial aspects of the models. A series of comparisons is then conducted on a number of test collections and retrieval tasks to determine the empirical and practical differences between the different models. Our results indicate that for sentence retrieval the multi-variate Bernoulli model can significantly outperform the Multinomial model. However, for the other tasks the Multinomial model provides consistently better performance (and in most cases significantly so). An analysis of the various retrieval characteristics reveals that the multi-variate Bernoulli model tends to promote long documents whose non-query terms are informative. While this is detrimental to the task of document retrieval (documents tend to contain considerable non-query content), it is valuable for other tasks such as sentence retrieval, where the retrieved elements are very short and focused.

Categories and Subject Descriptors: H.3 [**Information Systems**]: Information Storage and Retrieval; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Retrieval models*

General Terms: Theory, Experimentation
Additional Key Words and Phrases: Information Retrieval, Language Models, Multinomial, Multi-variate Bernoulli

1. INTRODUCTION

Statistical language models provide a principled way to quantitatively capture the uncertainty associated with the use of natural language. These formal statistical artifacts power a wide range of applications, involving language technology tasks in one way or another. For instance, discourse generation, automatic speech recognition and statistical machine translation have all applied Language Models (LM) to support their processes [Rabiner 1989; Manning and Schutze 2000; Rosenfeld 2000]. In the field of Information Retrieval (IR), LMs have facilitated the development of new principled approaches to retrieval which exploit powerful statistical estimation methods to improve retrieval performance. The use of LMs for IR is extremely valuable because it has opened the door to a stream of well-founded studies built upon Statistics and Probability Theory [deGroot 1988]. Since the late nineties, LMs have become a prominent research topic in IR, yielding successful results at an experimental level which has led to their widespread adoption.

The seminal application of LM for IR was proposed by Ponte and Croft [Ponte and Croft 1998] and Ponte [Ponte 1998]. The generation of queries is treated as a random process where a document language model is inferred for each document in the collection. Given a query, the collection is ranked by estimating the probability of generating that query according to each of these document language models. This is often referred to as the query likelihood approach. In their application, the query likelihood was modeled as a multivariate Bernoulli (MB) distribution. However, a review of the literature since then reveals that the mainstream approach to estimating the query likelihood uses a distinctly different distribution, the Multinomial (MN). This was introduced and developed concurrently by Miller *et al.* [Miller et al. 1999] and Hiemstra [Hiemstra 2000]. In this context, many unigram LMs, which are simply MN word distributions, and higher order n-gram models, which are conditional MN, have been proposed to support several IR tasks. For more extensive details about the Multinomial approach and its development in IR, we refer the reader to the books [Croft 2000; Croft and Lafferty 2003] and many theses [Hiemstra 2001; Zhai 2002; Kraaij 2004; Lavrenko 2004; Azzopardi 2005] dedicated to the subject. Whilst, the development of the MB has been very limited, with only a few recent publications [Metzler et al. 2004; Losada 2005; Azzopardi and Losada 2006].

The adoption of the MN model over the MB model seems to have been for two main reasons, besides the effectiveness of the MN model over other retrieval methods: simplicity and efficiency. Conceptually, the MN model is simpler to understand, easier to implement and appears more efficient. Intuitively, the MN model can be related to the problem of sampling balls from urns (which is described in most statistics textbooks). And the standard estimation procedures used for the Multinomial model provide principled estimates relying only on one parameter, which can be tuned to deliver state of the art performance. The simplicity of the MN model translated into a computationally efficient algorithm equivalent to tf/idf [Zhai and Lafferty 2001b; Hiemstra 2000]. On the other hand, the MB model requires a computation over non-query terms which introduces more computational effort and lacks some of the intuitiveness of the MN sampling. Further, the initial formulation under the MB model was overly complicated and difficult to follow and re-produce [Ponte and Croft 1998]. And so, the MB model has been usually regarded as a very inefficient and complicated method which is not applicable or feasible in a practical setting. These reasons meant that the MB model was superseded by the streamlined and elegant Multinomial model. As a consequence of these implicit and explicit decisions, little is actually known

about the retrieval effectiveness of the MB model in contrast to the well studied MN.

The recent work investigating the effectiveness of MB models has prompted our interest in revisiting the model, not only in terms of efficiency and effectiveness, but also how the models differ theoretically and behaviorally. While the MB and MN models both apply the query likelihood to rank documents, the MB approximation of the query likelihood differs in a very distinctive manner. The MB query likelihood includes a product across every non-query term whose implications for retrieval are largely unknown. In particular, no one has derived the retrieval function associated to the MB models. Unlike the MN models, whose implicit retrieval function was derived in [Zhai and Lafferty 2001b; 2004], the effects of the MB distribution with respect to important aspects such as inverse document frequency (idf) and document length remain unclear. In our work, we provide the retrieval function derived from the MB model and show the role of idf and document length in the context of these models. Following this theoretical derivation, it becomes obvious that MB models can also be implemented efficiently.

The current evidence in the literature presents a mixed view of the MB models' retrieval effectiveness. The original approximation of the MB query likelihood by Ponte and Croft [Ponte and Croft 1998] was shown to be significantly better than benchmark IR models (i.e. BM25 and tf/idf). In a recent study by Amati [Amati 2006], he experimented with Ponte and Croft's approach and an alternative Bernoulli-based LM, finding them comparable to the Divergence from Randomness approaches [Amati and van Rijsbergen 2002]. Nevertheless, neither study compared the MB models against the standard MN models. However, in work by McCallum and Nigam [Mccallum and Nigam 1998], they compared the performance of multi-variate Bernoulli and Multinomial models on a classification task using several different test collections. They provided evidence on the adequacy of a multi-variate Bernoulli view when vocabulary sizes were small, but as the vocabulary size increased the Multinomial model performed significantly better. In Metzler et al. [Metzler et al. 2004], they used a Bayesian estimation of the MB model which lead to two different formulations of the MB model (one which accounted for term frequency, and one that did not). They found that the former MB model was comparable to the MN model, but the latter performed poorly against the Multinomial model. However, findings in the context of sentence retrieval by Losada [Losada 2005], provides evidence to suggest that the basic MB model is comparable with MN. The characteristics of sentences appear to be more suited to a formulation based on the MB distribution. First, the lack of a non-binary term frequency component in the standard MB model seems less important for this task because sentences are short pieces of text. Second, the likelihood used in MB models, which takes into account the non-query terms, might be especially useful for isolating relevant material at a sentence level. As we will show in section 3, the MB models not only consider the matching terms but also takes into account how discriminative the rest of the document/sentence terms are. This means that the MB models rank documents and sentences using content *beyond* just the query words. For document retrieval this might problematic because verbose documents covering many topics tend to be promoted in the ranking. However, for sentence retrieval this could be a good feature because sentences are very specific, and, therefore, the risk of promoting irrelevant material is reduced.

A thorough investigation into behavioral aspects of the different language models and how this translates in differing performance is warranted, in order to capture, understand and appreciate the differences between MN and MB. The goal of this work is to study in

depth the characteristics of both retrieval approaches based on the multi-variate Bernoulli and the Multinomial models. This is performed on three levels; theoretically and formally, computationally and behaviorally. To achieve this we first provide a review of the contextualization of the MB model under the context of Bayesian Learning. Then, we address the derivation of the retrieval functions associated to each model. This helps significantly to gain insight into the understanding of these models and, also clarifies the computational efficiency issues involved. Finally, we examine the behavior of the MB approaches by reporting a number of experiments for several IR problems (retrieval with different granularities: sentences, summaries and full documents) showing how it varies dramatically given these different tasks. We show when the MB models perform better and worse than MN models, and explain the specific features of the MB approach which affect its performance. Finally, we show that the MB models are especially suitable when retrieval elements are short and focused, as is the case of sentence retrieval.

The rest of the paper is organized as follows. Section 2 outlines the basic foundations of Language Modeling as applied to Information Retrieval, with emphasis on Bayesian Learning under the two different query likelihood approaches based on the MN distribution and the MB distribution. The retrieval functions derived from the models and the implications in computational complexity are presented in Section 3. An empirical comparison between the two approaches is performed to assess their retrieval effectiveness over a number of retrieval tasks. Experiments are conducted on sentence retrieval in Section 4.1, document retrieval in Section 4.2 and summary retrieval in Section 4.3. Section 5 provides our discussion on the utility of the multi-variate Bernoulli approach, given the current and prior research. Our conclusions are then presented in the final section.

2. LANGUAGE MODELING

Since the introduction of Language Modeling for Information Retrieval [Ponte and Croft 1998], different approaches applying statistical language models for supporting the retrieval process have been proposed. The initial approach was the query likelihood approach, which is based on 1) estimating a statistical language model for every document and 2) computing the probability of generating the query according to each document model. The first step is a classical statistical learning problem [Bernardo and Smith 1994]. We have to estimate a probability distribution (θ_D) from the document's text D, which is treated as a language sample. This problem has been recurrently studied in the context of Bayesian statistics which provides a powerful framework for estimation because it removes the reliance of explicitly tuning parameters [Zaragoza et al. 2003]. Under the Bayesian framework the estimation of document language models is formally justified and so avoids any kind of ad hoc or heuristic estimation procedures. So, given the document D, we can compute the probability of different probability distributions by applying Bayes' rule:

$$P(\theta_D|D) = \frac{P(D|\theta_D)P(\theta_D)}{P(D)} \tag{1}$$

$P(\theta_D)$ encodes our prior belief about the adequacy of the distribution θ_D, $P(D|\theta_D)$ is the likelihood of the data D under distribution θ_D and $P(\theta_D|D)$ is the posterior distribution. $P(D)$ is the probability of generating the document and it is independent on θ_D.

This is a Bayesian learning problem where the hypotheses are all possible probability distributions and the data is the document itself. Bayesian learning analyzes the data available and makes predictions about the unknown model parameters [deGroot 1988]. Bayesian techniques are very well suited to this sort of learning problem where the data samples are small and we need to take account of the resulting parameter uncertainty. This is very valuable in the current application scenario, where there is a single (and fixed-sized) data sample (i.e. the document). This contrasts to other IR tasks, where we might have more data to drive the estimation of a single LM. For instance, estimating a LM for a whole collection (e.g. for resource selection in distributed retrieval [Callan and Connell 2001]), estimating a LM of a small set of relevant documents (e.g. to estimate a model of relevance in the context of a feedback cycle [Zhai and Lafferty 2001a]), etc. In such cases, the hypothesis space would still be infinite but the final estimates are expected to be more reliable. Note that the application of LM in IR assumes that language is generated as a random process modeled by an unknown probability distribution [Ponte and Croft 1998]. Of course, it is not the case that texts (and queries) are generated randomly, but it is the case that retrieval systems are not endowed with knowledge of the generation process. In this respect, it is natural to apply a Bayesian approach that captures uncertainty in a principled way.

In order to obtain an estimate $\widehat{\theta_D}$, different approximations have been studied in the literature. A very common approach is to select the single most probable distribution $\widehat{\theta_D}$ that maximizes $P(\theta_D|D)$ (i.e. the mode of the posterior distribution). This is called *maximum a posteriori* distribution[1]. Moreover, if we assume that all distributions are equally probable (i.e. we choose a uniform prior distribution for $P(\theta_D)$) then we get the maximum likelihood (ML) estimator, whose problems for IR have been extensively reported in the literature (i.e. the Zero Probability Problem [Ponte and Croft 1998]).

A different estimation method uses the expectation of the posterior distribution ($E(\theta_D|D)$) as the estimated document model. This estimate is more conservative because it is a move away from the most probable distribution as it *averages out* over all possible posterior distributions. This approach ensures that the expected loss computed from the least quadratic error function is minimum[2] [deGroot 1988].

The two methods summarized in the previous paragraphs apply different techniques for obtaining a *single point estimate* which is subsequently used for computing the query likelihood. Another approach results from the application of the predictive distribution [Zaragoza et al. 2003]. This method takes more uncertainty into account because the probability of the query is averaged out under the model over all possible distributions, weighted by the posterior probability $P(\theta_D|D)$. With appropriate choice of the prior distribution $P(\theta_D)$ this integral can be computed analytically leading to a tractable retrieval formula.

In general, depending on the characteristics of the document in question (e.g. document length), the posterior distribution $P(\theta_D|D)$ can be narrow or broad. In the case of a long document, the posterior $P(\theta_D|D)$ might be peaked around some value $\widehat{\theta_D}$ and thus a narrower distribution characterizes the lower uncertainty in the parameter values. Conversely, when there is not abundant data, the posterior distribution is broader, accounting for the

[1] We can just drop the factor $P(D)$ and take the distribution that maximizes $P(D|\theta_D)P(\theta_D)$.

[2] Roughly speaking, the *average error* w.r.t an hypothetical ideal distribution is minimized.

uncertainty in the parameter values.

However, single point estimates are standard and commonly employed in LM for IR. For instance, as it will be shown in the next section, the well-known Dirichlet smoothing, which has been shown to be a very effective method for retrieval [Zhai and Lafferty 2001b], can be obtained from the single point estimates in a straightforward manner[3]. In this paper, we will be focused on LMs that employ a *maximum a posteriori* estimate. Note that similar estimates can be also derived from the expectation-based method with appropriate adjustments in the parameters of the model.

The choice of probability distributions is central to Bayesian inference. As argued above, in real applications we must resort to approximate or simplified methods to obtain tractable formulas. In the following section we detail how the Multinomial and multivariate Bernoulli distributions can be explained in the context of the Bayesian framework, which defines the two retrieval models (MN and MB) and how they approximate the query likelihood.

2.1 Multinomial Model

A key modeling decision is the specific form of the likelihood $P(D|\theta_D)$. Consider the representation of texts as vectors of term counts:

$$D = (tf(1, D), tf(2, D), \cdots, tf(|V|, D)) \tag{2}$$

where V is the vocabulary, $|V|$ denotes the size of the vocabulary and $tf(i, D)$ is the count of term w_i in the document D. The total term count of the document will be referred to as n_D ($n_D = \sum_{i=1}^{|V|} tf(i, D)$). These vectors can be regarded as possible outcomes of a random experiment modeled by a Multinomial distribution with parameters n_D and $\{\theta_i\}_{i=1}^{|V|}$, with $\theta_i = P(w_i|\theta)$:

$$P(D|\theta_D) = \frac{n_D!}{tf(1, D)! \cdot tf(2, D)! \cdots tf(|V|, D)!} \cdot \prod_{i=1}^{|V|} \theta_i^{tf(i,D)} \tag{3}$$

The probabilities θ_i are the probabilities of emission of the different terms in the vocabulary. Note that, this model assumes that terms are generated independently. Since a document is regarded as a sample from a Multinomial distribution, the model implicitly assumes that the document results from n_D independent trials, where each trial is associated to a word position. So, the possible outcomes of every individual experiment are the words in the vocabulary ($\sum_{|V|}^{1} \theta_i = 1$). Further, note that the combinatorial coefficient represents the number of possible sequences of texts associated to a given vector. Statistical language models do not include this factor because the space of events is different (sequences of terms in a given order rather than unordered bags of terms).

Given the likelihood $P(D|\theta_D)$, different choices of the prior distribution $P(\theta_D)$ may be selected. Nevertheless, for certain choices of the prior, the posterior distribution $P(\theta_D|D)$ has the same algebraic form as the prior. Such a choice is called *conjugate prior* and is an algebraic convenience because it simplifies enormously the Bayesian analysis. For

[3]The reader interested in a comparison between single point estimation and predictive smoothing can review the work by Zaragoza *et al.* [Zaragoza et al. 2003].

instance, when $P(D|\theta_D)$ parameterizes a Multinomial and $P(\theta_D)$ is Dirichlet (with parameters α_i, i.e. $P(\theta_D) \sim Dir(\alpha_i)$), which is the conjugate prior for the Multinomial distribution, the posterior distribution $P(\theta_D|D)$ is also a Dirichlet distribution:

$$P(\theta_D|D) \propto \frac{\Gamma(n_D + \sum_{i=1}^{|V|} \alpha_i)}{\prod_{i=1}^{|V|} \Gamma(tf(i,D) + \alpha_i)} \prod_{i=1}^{|V|} (\theta_i)^{tf(i,D)+\alpha_i - 1} \tag{4}$$

where Γ is the Gamma function. The values α_i are hyper-parameters, which can be interpreted as additional data or pseudo-counts associated to each term w_i. The Dirichlet distribution computes a probability associated to each choice of the Multinomial parameters, $\{\theta_i\}_{i=1}^{|V|}$, taking into account the set of hyper-parameters α_i, which encodes *prior observation counts* for the terms. The posterior distribution is also Dirichlet with parameters $\alpha_i + tf(i,D)$ (i.e. $P(\theta_D|D) \sim Dir(\alpha_i + tf(i,D))$). Applying now the *maximum a posteriori* approach on the previous equation:

$$\widehat{\theta_D} = \arg\max_{\theta_D} \frac{\Gamma(n_D + \sum_{i=1}^{|V|} \alpha_i)}{\prod_{i=1}^{|V|} \Gamma(tf(i,D) + \alpha_i)} \prod_{i=1}^{|V|} (\theta_i)^{tf(i,D)+\alpha_i - 1} \tag{5}$$

It can be proved that the solution to this equation results in the following estimates for the Multinomial parameters:

$$\widehat{\theta_i} = P(w_i|\widehat{\theta_D}) = \frac{tf(i,D) + \alpha_i - 1}{n_D + \sum_{i=1}^{|V|} \alpha_i - |V|} \tag{6}$$

When the hyper-parameters α_i are set to the same value then we are giving equal *a priori* counts to all terms in the vocabulary. In particular, setting $\alpha_i = 1$ results in the maximum likelihood estimator and $\alpha_i = 2$ results in Laplace smoothing [Laplace 1995]. The formula can also handle Lidstone's smoothing [Lidstone 1920] by setting all α_i values equal to $\lambda + 1$, where λ is the positive value added by Lidstone's law[4]. On the other hand, setting $\alpha_i = \mu P(w_i|C) + 1$ gives the Dirichlet smoothing, in which we apply prior counts for the words across the vocabulary depending on its distribution on some background model (e.g. the model estimated based on the collection statistics, in this case $P(w_i|C)$). This is intuitive because, before viewing specific text, the only information we have to drive the estimation is the distribution of the words in the background model .

The maximum a posteriori approach takes the mode of the posterior distribution, $\widehat{\theta_D} = \arg\max_{\theta_D} P(\theta_D|D)$, but, as argued in the previous section, there are a number of alternative ways to get to a final single point estimate. We can use the expectation of the posterior distribution as an estimate, $\widehat{\theta_D} = E[P(\theta_D|D)]$. The posterior distribution is a Dirichlet distribution with parameters $\alpha_i + tf(i,D)$ and, therefore, its expectation is equal to $\frac{\alpha_i + tf(i,D)}{\sum_i \alpha_i + tf(i,D)}$[5]). This leads to the following estimate:

[4] Lidstone's law of succession, which can be seen as a generalization of Laplace's approach, adds a real positive value λ to the empirical count associated to each term.

[5] The expectation of a Dirichlet distribution with parameters α_i is equal to $\frac{\alpha_i}{\sum_i \alpha_i}$.

$$\widehat{\theta_i} = P(w_i|\widehat{\theta_D}) = \frac{tf(i,D) + \alpha_i}{n_D + \sum_{i=1}^{|V|} \alpha_i} \tag{7}$$

where, again, popular smoothing strategies can be implemented with the appropriate selection of the α_i values[6]. Although we will focus here on the estimates derived from *maximum a posteriori* (equation 6), it is easy to show estimates can be derived by choosing the hyper-parameters appropriately.

Once the document's model $\widehat{\theta_D}$ has been estimated, we can compute the probability that the query is generated from the estimated document model (query likelihood). Queries are also represented as vector of indexed term counts:

$$Q = (tf(1,Q), tf(2,Q), \cdots, tf(|V|,Q)) \tag{8}$$

The total term count of the query will be referred to as n_Q ($n_Q = \sum_{i=1}^{|V|} tf(i,Q)$). This is modeled by a Multinomial distribution with parameters n_Q and $\{\widehat{\theta_i}\}_{i=1}^{|V|}$.

$$P(Q|\widehat{\theta_D}) = \frac{n_Q!}{tf(1,Q)! \cdot tf(2,Q)! \cdots tf(|V|,Q)!} \cdot \prod_{i=1}^{|V|} (\widehat{\theta_i})^{tf(i,Q)} \tag{9}$$

The first factor depends only on the query and, therefore, it can be dropped for document ranking purposes (i.e. Eq. 10 is the basis of the MN retrieval model). This means that this formulation yields the same ranking of documents as the one produced by computing the probability of generating a given sequence of query terms according to a unigram Language Model (word order is disregarded in unigram models).

Note that, following this Bayesian estimation process, documents and queries are assumed to be generated by distributions with the same form [Zaragoza et al. 2003]. This is not the case in other LM approaches such as the approach by Ponte and Croft [Ponte and Croft 1998] or the two-stage smoothing proposed by Zhai and Lafferty [Zhai and Lafferty 2002]. In this paper we are only concerned with LM models based on this Bayesian derivation, and do not consider other probability distributions which are derived in other LM approaches (such as [Hiemstra 2000]).

$$P(Q|\widehat{\theta_D}) \propto \prod_{w_i \in Q} p(w_i|\widehat{\theta_i})^{tf(i,Q)} \tag{10}$$

The retrieval function derived by Eq. 10 was shown to be equivalent to a sum of logarithms of tf/idf-like weights of matching terms plus a document dependent factor [Zhai and Lafferty 2001b][7]. The models based on the Kullback-Leibler divergence (KLD), which compute the divergence between two probability distributions, cover the MN query likelihood approach as a special case [Zhai 2002] when the query model is the empirical distribution of the query. However, other more general KLD-based retrieval functions cannot be derived from this Bayesian analysis.

The motivation of scoring documents by query likelihood, was derived by Ponte and Croft [Ponte and Croft 1998], who suggested that the probability of a document given

[6] In particular, setting $\alpha_i = \mu P(w_i|C)$ in eq. 7 yields again the popular Dirichlet smoothing.

[7] The specific retrieval function derived is shown in section 3.

the query, would be indicative of the document's relevance (i.e. the query infers the document)[8]. Through the application of Bayes Theorem, this can be approximated through the query likelihood as follows:

$$P(D|Q) = \frac{P(Q|D) \cdot P(D)}{P(Q)} \tag{11}$$

$$P(D|Q) \approx P(Q|D) \cdot P(D) \tag{12}$$

$$P(D|Q) \approx P(Q|D) \tag{13}$$

Here, the prior probability of a query, $P(Q)$ is constant and independent of any document feature and so it can be discarded for ranking purposes. The prior probability of a document, $P(D)$ captures our prior belief that the document is relevant to any query. If we assume that $P(D)$ is to be constant (i.e. uniform) then we can score documents by the query likelihood, alone. While this is the case for most existing work [Zhai and Lafferty 2001b], some papers have applied non-uniform priors to model factors such as document length, link information, url length or webpage age [Miller et al. 1999; Hiemstra 2000; Kraaij et al. 2002; Kamps 2005; Hauff and Azzopardi 2005; Losada and Azzopardi 2008]. In this work we are only concerned with the standard query likelihood approach and the effects of choosing different probability distributions to model this likelihood.

2.2 Multi-variate Bernoulli Model

Instead of assuming that the texts are bags of words, the MB approach makes a different assumption. The text generation process is viewed as a random process in which $|V|$ independent Bernoulli trials are run. In this case, the space of events is very different from the Multinomial case. It is composed of binary vectors of size |V|[9]. Consequently, the number of times a word occurs in a text is not captured.

$$D = (\delta_{1,D}, \delta_{2,D}, \cdots, \delta_{|V|,D}) \tag{14}$$

where $\delta_{i,D}$ is 1 if $w_i \in D$ and 0 otherwise. This vector represents the outcome of multiple Bernoulli processes. Each individual Bernoulli process is governed by its probability of success (i.e. the probability of selecting the term to be included in the generated text), which will be referred to as $\theta_i = P(w_i|\theta)$, with $\sum_{i=1}^{|V|} \theta_i = 1$. Observe that the sampling process is significantly different to the MN case. Rather than sampling at each word position in the document, the MB model assumes that there is a trial associated to each word in the vocabulary. It models a hypothetical process in which the author of the document reviews every term of the vocabulary to determine whether or not the term should be included into the document's text. Observe that the possible outcome of every individual trial is binary, meaning that multiple occurrences of a given term in the document are ignored. This binary notion of presence of terms in texts and the explicit consideration of the missing terms (the model takes into account the document terms but also the non-occurring terms) are the two main distinctive features of the MB model with respect to the MN model.

[8] Note that Ponte and Croft [Ponte and Croft 1998] make the assumption that the query likelihood is correlated with the relevance of a document, see Azzopardi [Azzopardi 2005](pg.44) or Azzopardi and Roelleke [Azzopardi and Roelleke 2007] for a detailed discussion.

[9] The Binary Independence Model in the context of classical Probabilistic Models of IR stands on an analagous space of events.

The form of the likelihood under the MB approach is:

$$\begin{aligned} P(D|\theta) &= \prod_{w_i \in D} P(w_i|\theta) \prod_{w_i \notin D} (1 - P(w_i|\theta)) \\ &= \prod_{i=1}^{|V|} P(w_i|\theta)^{\delta_{i,D}} (1 - P(w_i|\theta))^{1-\delta_{i,D}} \end{aligned} \tag{15}$$

A multi-variate Bernoulli formulation for the query generation process was taken by Ponte and Croft in their original application of LM for IR [Ponte and Croft 1998]. However their estimation of the document models was based on a geometric distribution and was somewhat artificial. In the context of Bayesian learning, the interpretation of text generation as a multi-variate Bernoulli process leads to a coherent framework in which queries and documents are treated in a uniform way. The conjugate prior for the multi-variate Bernoulli distribution is the multiple-Beta distribution:

$$P(\theta) = \prod_{i=1}^{|V|} \frac{\Gamma(\alpha_i + \beta_i)}{\Gamma(\alpha_i)\Gamma(\beta_i)} \theta_i^{\alpha_i - 1} (1 - \theta_i)^{\beta_i - 1} \tag{16}$$

where α_i and β_i are parameters associated to the Beta distribution. These parameters determine the shape of the probability density function. When the two parameters are equal, the distribution is symmetrical. For example, when both α_i and β_i are equal to one, the distribution becomes uniform. If α_i is less than β_i, the distribution is skewed to the left. And if α_i is greater than β_i, the distribution is skewed to the right. Applying the *maximum a posteriori* distribution:

$$\begin{aligned} \widehat{\theta_D} &= \arg\max_{\theta_D} P(D|\theta_D) P(\theta_D) \\ &= \arg\max_{\theta_D} \prod_{i=1}^{|V|} (\theta_i)^{\delta_{i,D}} (1 - \theta_i)^{1-\delta_{i,D}} \prod_{i=1}^{|V|} \frac{\Gamma(\alpha_i + \beta_i)}{\Gamma(\alpha_i)\Gamma(\beta_i)} \theta_i^{\alpha_i - 1} (1 - \theta_i)^{\beta_i - 1} \\ &= \arg\max_{\theta_D} \prod_{i=1}^{|V|} \frac{\Gamma(\alpha_i + \beta_i)}{\Gamma(\alpha_i)\Gamma(\beta_i)} (\theta_i)^{\delta_{i,D} + \alpha_i - 1} (1 - \theta_i)^{\beta_i - \delta_{i,D}} \end{aligned}$$

The solution to this equation results in the following estimates for the Bernoulli parameters [Metzler et al. 2004]:

$$\widehat{\theta_i} = P(w_i|\widehat{\theta_D}) = \frac{\delta_{i,D} + \alpha_i - 1}{\alpha_i + \beta_i - 1} \tag{17}$$

when $\alpha_i = \beta_i = 1$ this results in the maximum likelihood estimator. Smoothed probabilities can be obtained from some background model, e.g. $\alpha_i = \mu P(w_i|C) + 1, \beta_i = \frac{1}{P(w_i|C)} + \mu(1 - P(w_i|C)) - 1$. These specific values are obtained after setting the expectation value of θ_i to be equal to the probability of the term w_i in the background model. Since the prior distribution is multiple Beta, its expectation is equal to $\frac{\alpha_i}{\alpha_i + \beta_i}$. The values above ensure that the expectation of the prior is equal to $p(w_i|C)$. This is intuitive because, before seeing any specific text, it is desirable that the expectation of the prior reflects the

distribution of the terms in the background collection. Similarly to the MN case, we set $\alpha_i = \mu P(w_i|C) + 1$ (this makes that the estimated LM smoothing dependant on the background model) and we derive the β_i value that ensures $\frac{\alpha_i}{\alpha_i+\beta_i}$ is equal to $P(w_i|C)$. This derivation can be found in Appendix A. This is indeed quite similar to Dirichlet smoothing in the MN distribution. Of course, many other choices of α_i and β_i are possible. From the estimated $\widehat{\theta_D}$, the query likelihood of the MB model is computed as:

$$P(Q|\widehat{\theta_D}) = \prod_{w_i \in Q} P(w_i|\widehat{\theta_D}) \prod_{w_i \notin Q} (1 - P(w_i|\widehat{\theta_D})) \tag{18}$$

2.3 Extended multi-variate Bernoulli Model

One of the obvious limitations of the MB model is that it ignores multiple occurrences of terms in texts since the event space is composed of $|V|$-sized binary vectors. The number of times that a word is mentioned in a piece of text has been shown to provide a good indication of the importance of the term to characterize the document's contents, as proved by the recurrent success obtained by incorporating the term frequency component in most IR systems and tasks. Metzler *et al.* [Metzler et al. 2004] have designed a variant of the MB model to cater for non-binary term frequency information. The MB approach proposed in [Metzler et al. 2004], was named *Model B*, and while some of its assumptions are questionable we include the model in our study for completeness[10]. To avoid confusion, the standard MB model presented in the previous section will be referred to as MB model whereas the *extended* MB model will be referred to as MBB.

Instead of associating every Bernoulli trial with a document and a term, the MBB model results from assuming that multiple Bernoulli trials (as many as vocabulary terms) are run *at every position in the document*. That is, every Bernoulli trial is associated to a document, a term and a position in the document. Documents are therefore modeled as a collection of samples from a multi-variate distribution, where each sample is a binary vector (whose size is the size of the vocabulary) containing a single element set to 1 (corresponding to the word appearing in that location):

$$D = (pos_{1,D}, pos_{2,D}, \cdots, pos_{n_D,D}) \tag{19}$$

$$pos_{j,D} = (\delta_{1,j,D}, \delta_{2,j,D}, \cdots, \delta_{|V|,j,D}), j = 1, \cdots, n_D \tag{20}$$

where $\delta_{i,j,D}$ is set to 1 when the term w_i appears in the j-th position of document D. The $pos_{j,D}$ vectors result from multiple Bernoulli processes governed by the probabilities $\theta_i = P(w_i|\theta)$, with $\sum_{i=1}^{|V|} \theta_i = 1$

This means that, for each term, we have n_D binary samples reflecting its appearances throughout the text. In the MB model, the estimation of $p(w_i|\widehat{\theta_D})$ (i.e. the probability of generating the term from the document's model which, roughly speaking, is an indication of how good the term is in describing the document) is determined by the presence/absence of the term in the document. On the contrary, the MBB model handles n_D binary samples for each term, reflecting whether or not the term is present at each document's location. The more times a term appears, the more likely the document is *about* that term.

[10] In a recent work by Amati [Amati 2006], another interesting Bernoulli-based formulation was discussed. This approach, based on Dirichlet priors (instead of multiple-beta priors) can also handle non-binary frequencies.

One could argue that the MBB model is somehow artificial because the natural generative modeling of text, with multiple occurrences of terms, is a Multinomial distribution and to apply a multi-variate Bernoulli distribution is not justified. Indeed, there are some components in the MBB model which, from a probabilistic point of view, could be a matter of concern. Since a multiple Bernoulli process is run at every position in the document (i.e. |V| Bernoulli trials), in theory, any |V|-sized binary vector can result from this random process. Nevertheless, the samples used to estimate the model are not unconstrained binary vectors but have a single element set to 1 (because, in a given text location, a single term appears). This modeling leads a constrained or construed probabilistic stand point. Nonetheless, we still feel that a review on the MBB model is in order here because 1) it exemplifies the limitations of the multi-variate distribution for modeling text generation and can even act as an incentive to promote the development and study of more formal Bernoulli-based models, and 2) the experiments reported in [Metzler et al. 2004] show significant improvements w.r.t. the MB model and, thus, it is interesting to study empirically the effect of term frequency in the context of a multiple Bernoulli likelihood. The MBB likelihood takes the following form:

$$P(D|\theta) \;=\; \prod_{i=1}^{|V|} P(w_i|\theta)^{tf(i,D)}(1-P(w_i|\theta))^{n_D-tf(i,D)} \tag{21}$$

Renaming again $P(w_i|\theta)$ as θ_i, the conjugate prior (multiple-Beta) can be expressed as:

$$P(\theta) = \prod_{i=1}^{|V|} \frac{\Gamma(\alpha_i+\beta_i)}{\Gamma(\alpha_i)\Gamma(\beta_i)}\theta_i^{\alpha_i-1}(1-\theta_i)^{\beta_i-1} \tag{22}$$

Finally, the application of the *maximum a posteriori* distribution leads to:

$$\begin{aligned}
\widehat{\theta_D} \;&=\; \arg\max_{\theta_D} P(D|\theta_D)P(\theta_D) \\
&=\; \arg\max_{\theta_D} \prod_{i=1}^{|V|}(\theta_i)^{tf(i,D)}(1-\theta_i)^{n_D-tf(i,D)} \prod_{i=1}^{|V|} \frac{\Gamma(\alpha_i+\beta_i)}{\Gamma(\alpha_i)\Gamma(\beta_i)}\theta_i^{\alpha_i-1}(1-\theta_i)^{\beta_i-1} \\
&=\; \arg\max_{\theta_D} \prod_{i=1}^{|V|} \frac{\Gamma(\alpha_i+\beta_i)}{\Gamma(\alpha_i)\Gamma(\beta_i)}(\theta_i)^{tf(i,D)+\alpha_i-1}(1-\theta_i)^{n_D+\beta_i-tf(i,D)-1}
\end{aligned}$$

and the maximum yields the following estimates for the parameters:

$$\widehat{\theta_i} = P(w_i|\widehat{\theta_D}) = \frac{tf(i,D)+\alpha_i-1}{n_D+\alpha_i+\beta_i-2} \tag{23}$$

Again, $\alpha_i = \beta_i = 1$ results in the maximum likelihood estimator and smoothed probabilities can be obtained by setting the expectation value of θ_i to be equal to the probability of the term w_i in a background model. Once the model is estimated, the MBB query likelihood is:

$$P(Q|\widehat{\theta_D}) \;=\; \prod_{i=1}^{|V|} P(w_i|\widehat{\theta_D})^{tf(i,Q)}(1-P(w_i|\widehat{\theta_D}))^{n_Q - tf(i,Q)} \tag{24}$$

where $tf(i,Q)$ is the frequency of the term in the query. The model therefore is able to capture and utilize term frequency information[11].

2.4 Additional remarks

In the previous subsections, we have provided an overview of the three different Language Models in the context of Bayesian Learning. We have shown how the different approaches lead to different formulations of the query likelihood. Now we turn our attention to the implications of the differences and consider how the these interpretations affect the retrieval performance. The MB and MBB models present a very distinctive feature, the product across non query terms, whose effect on retrieval performance has not been studied in depth. In our experiments, we try to shed light on its effect on retrieval performance. The consideration of different multi-variate Bernoulli models in our study will also permit to analyze how retrieval performance is affected when the probabilistic model of text handles either binary or non-binary term frequencies. We therefore expect that the comparison of these multi-variate Bernoulli models with the standard Multinomial model will help to clarify the role of the MB models in the current map of LM approaches for IR and, possibly, identify IR tasks that can benefit from a using multi-variate Bernoulli distribution to estimate the query likelihood.

In this work we are only concerned with smoothing strategies derived from eqs 6, 17 and 23. Other smoothing methods, such as Jelinek-Mercer, backoff [Zhai and Lafferty 2001b] or two-stage smoothing [Zhai and Lafferty 2002], cannot be studied under this framework because they are based on different assumptions. For instance, Jelinek-Mercer involves a mixture of two generative models that cannot be derived from Bayesian learning [Zaragoza et al. 2003]. Two-stage smoothing was developed to combine the Bayesian Dirichlet smoothing and Jelinek-Mercer and it can be seen as a first approximation to the full Bayesian treatment of the linear interpolation smoothing method [Zaragoza et al. 2003]. To summarize, the Bayesian derivations of the MN and MB(B) models we have presented provide the framework to contrast and compare the differences between the different strategies in a principled way.

3. RETRIEVAL FUNCTIONS DERIVED FROM THE MODELS

It is interesting to analyze the retrieval functions derived from the MB models when applied with the estimations presented in section 2. The purpose of this derivation is two-fold. First, this will help to gain insight into the MB approaches, explaining the connection with popular IR components such as inverse document frequency and document length. Second, we address the computational issues involved in estimating the MB model and show that an efficient algorithm can be designed to perform retrieval efficiently. Next we outline the retrieval function for the MN before deriving the retrieval functions for the MB models.

[11] In section 3 we present the retrieval function derived from this model showing that the term frequency factor is handled in a log-linear way.

3.1 MN Model Retrieval Function

First of all it is important to observe that the retrieval function derived from the Multinomial model presented in section 2.1 has been shown to be equivalent to [Zhai and Lafferty 2001b; 2004]:

$$log\ p(Q|\theta_D) \approx \sum_{w_i \in Q \cap D} log(1 + \frac{tf(w_i, D)}{\mu p(w_i|C)}) + n_q \cdot log\frac{\mu}{n_D + \mu} \tag{25}$$

where μ is a free parameter. The MN retrieval function is therefore reduced to a sum of weights for each term common between the query and the document plus a document length dependent constant. The weights of the matching terms are clearly connected to the popular tf/idf matching strategies (the lower $p(w_i|C)$, the more significant the term w_i). This was referred to as the Inverse Collection Frequency (icf), and since the method multipled the icf with tf, together it was referred to as tf.icf.[12] A complete study on the interactions between smoothing and document length for this retrieval formula can be found in [Losada and Azzopardi 2008]. Regarding the MB models, we now derive equivalent formulations for these models.

3.2 MB Model Retrieval Function

Applying logarithms on the MB likelihood (eq 18) obtains:

$$\log P(Q|\widehat{\theta_D}) \ = \sum_{w_i \in Q} \log P(w_i|\widehat{\theta_D}) + \sum_{w_i \notin Q} \log(1 - P(w_i|\widehat{\theta_D})) \tag{26}$$

Following Zhai and Lafferty's terminology [Zhai and Lafferty 2001b; 2004] the probability of a seen term in a document D will be referred to as $P_s(w_i|\widehat{\theta_D})$ and the probability of an unseen term will be denoted by $P_u(w_i|\widehat{\theta_D})$. In the MB model these probabilities are defined as (eq. 17):

$$P_s(w_i|\widehat{\theta_D}) \ = \ \frac{\alpha_i}{\alpha_i + \beta_i - 1} \tag{27}$$

$$P_u(w_i|\widehat{\theta_D}) \ = \ \frac{\alpha_i - 1}{\alpha_i + \beta_i - 1} \tag{28}$$

where $\alpha_i = \mu P(w_i|C) + 1$ and $\beta_i = \frac{1}{P(w_i|C)} + \mu(1 - P(w_i|C)) - 1$. Re-writing eq.26, we obtain:

$$\begin{aligned} &\log P(Q|\widehat{\theta_D}) = \\ &\sum_{w_i \in Q \cap D} \log P_s(w_i|\widehat{\theta_D}) + \sum_{w_i \in Q, w_i \notin D} \log P_u(w_i|\widehat{\theta_D}) \\ &+ \sum_{w_i \notin Q, w_i \in D} \log(1 - P_s(w_i|\widehat{\theta_D})) + \sum_{w_i \notin Q, w_i \notin D} \log(1 - P_u(w_i|\widehat{\theta_D})) \end{aligned} \tag{29}$$

[12] See [Zhai and Lafferty 2001b; 2004] for more details, and [Hiemstra 2000] for a probabilistic justification of tf.idf.

By adding zero to Eq. 29 by first adding and then subtracting $\sum_{w_i \in Q \cap D} \log P_u(w_i|\widehat{\theta_D})$, and then re-arranging terms, we obtain:

$$\log P(Q|\widehat{\theta_D}) = \tag{30}$$
$$\sum_{w_i \in Q \cap D} \log \frac{P_s(w_i|\widehat{\theta_D})}{P_u(w_i|\widehat{\theta_D})} + \sum_{w_i \in Q} \log P_u(w_i|\widehat{\theta_D})$$
$$+ \sum_{w_i \notin Q, w_i \in D} \log(1 - P_s(w_i|\widehat{\theta_D})) + \sum_{w_i \notin Q, w_i \notin D} \log(1 - P_u(w_i|\widehat{\theta_D}))$$

Similarly, adding and subtracting $\sum_{w_i \in Q \cap D} \log(1 - P_s(w_i|\widehat{\theta_D}))$, we now obtain:

$$\log P(Q|\widehat{\theta_D}) = \tag{31}$$
$$\sum_{w_i \in Q \cap D} \log \frac{P_s(w_i|\widehat{\theta_D})}{P_u(w_i|\widehat{\theta_D}) \cdot (1 - P_s(w_i|\widehat{\theta_D}))} + \sum_{w_i \in Q} \log P_u(w_i|\widehat{\theta_D})$$
$$+ \sum_{w_i \in D} \log(1 - P_s(w_i|\widehat{\theta_D})) + \sum_{w_i \notin Q, w_i \notin D} \log(1 - P_u(w_i|\widehat{\theta_D}))$$

And finally, adding and subtracting $\sum_{w_i \in D} \log(1 - P_u(w_i|\widehat{\theta_D}))$, $\sum_{w_i \in Q} \log(1 - P_u(w_i|\widehat{\theta_D}))$ and $\sum_{w_i \in Q \cap D} \log(1 - P_u(w_i|\widehat{\theta_D}))$, we obtain the following expression:

$$\log P(Q|\widehat{\theta_D}) = \tag{32}$$
$$\sum_{w_i \in Q \cap D} \log \frac{P_s(w_i|\widehat{\theta_D}) \cdot (1 - P_u(w_i|\widehat{\theta_D}))}{P_u(w_i|\widehat{\theta_D}) \cdot (1 - P_s(w_i|\widehat{\theta_D}))} + \sum_{w_i \in Q} \log \frac{P_u(w_i|\widehat{\theta_D})}{(1 - P_u(w_i|\widehat{\theta_D}))}$$
$$+ \sum_{w_i \in D} \log \frac{(1 - P_s(w_i|\widehat{\theta_D}))}{(1 - P_u(w_i|\widehat{\theta_D}))} + \sum_{w_i \in V} \log(1 - P_u(w_i|\widehat{\theta_D}))$$

As shown in [Azzopardi and Losada 2006], this formula can be interpreted as follows. The latter addends (involving the sum across document terms and the sum across vocabulary terms) account for the probability of a hypothetical *empty* query being generated from the document language model. This computation is performed before query time (i.e. the starting assumption is that all terms will be non-query terms and, hence, the product across non-query terms can be computed in advance). Then, at query time, the pre-computed document score is adjusted according to the terms that appear in the query. In this respect, given the query terms, the second addend removes the initial assumption that they were non-query terms. Finally, the first addend accounts for the document-query matching terms, adjusting to the probability of a seen term, $P_s(w_i|\widehat{\theta_D})$ [13].

In eq. 32, $\sum_{w_i \in V} \log(1 - P_u(w_i|\widehat{\theta_D}))$ and $\sum_{w_i \in Q} \log \frac{P_u(w_i|\widehat{\theta_D})}{(1 - P_u(w_i|\widehat{\theta_D}))}$ can be dropped

[13] Note that, for a matching term, the 2nd, 3rd and 4th addends add $\log \frac{P_u(w_i|\widehat{\theta_D}) \cdot (1 - P_s(w_i|\widehat{\theta_D}))}{(1 - P_u(w_i|\widehat{\theta_D}))}$. We just need to subtract this value for the matching terms and add the correct one, $\log P_s(w_i|\widehat{\theta_D})$.

for document ranking purposes (note that $P_u(w_i|\widehat{\theta_D})$ in eq. 28 does not depend on any document feature):

$$\log P(Q|\widehat{\theta_D}) \approx \sum_{w_i \in Q \cap D} \log \frac{P_s(w_i|\widehat{\theta_D}) \cdot (1 - P_u(w_i|\widehat{\theta_D}))}{P_u(w_i|\widehat{\theta_D}) \cdot (1 - P_s(w_i|\widehat{\theta_D}))} + \sum_{w_i \in D} \log \frac{(1 - P_s(w_i|\widehat{\theta_D}))}{(1 - P_u(w_i|\widehat{\theta_D}))} \quad (33)$$

Using the probabilities in eqs. 27 and 28 and re-arranging the expression, we obtain:

$$\log P(Q|\theta_D) \approx \sum_{w_i \in Q \cap D} \log \frac{\alpha_i}{\alpha_i - 1} \cdot \frac{\beta_i}{\beta_i - 1} + \sum_{w_i \in D} \log \frac{\beta_i - 1}{\beta_i} \quad (34)$$

This formula shows that the MB model can be implemented efficiently. The second sum is a document dependent term that can be computed at indexing time. The first addednd is a regular sum across matching terms. This means that the MB models require only a negligible space penalty (to store the values of $\sum_{w_i \in D} \log \frac{\beta_i - 1}{\beta_i}$) w.r.t. other very efficient models (e.g. vector-space models[Salton et al. 1975]). In Appendix B we sketch the algorithm that implements the MB model to rank documents given a query. To further analyze the behaviour of the MB retrieval formula, note that eq. 34 can be re-written as:

$$\log P(Q|\theta_D) \approx \sum_{w_i \in Q \cap D} \log \frac{\alpha_i}{\alpha_i - 1} + \sum_{w_i \in D, w_i \notin Q} \log \frac{\beta_i - 1}{\beta_i} \quad (35)$$

Since $\alpha_i = \mu P(w_i|C) + 1$ and $\beta_i = \frac{1}{P(w_i|C)} + \mu(1 - P(w_i|C)) - 1$ this retrieval formula is decomposed into two parts: 1) a sum over query-document matching terms with idf-like weights and 2) a sum over document terms that are not in the query. This second sum promotes long documents with non-query terms that have a high idf (note that β_i increases as $P(w_i|C)$ decreases). This is a very interesting feature of the retrieval formula derived from the MB distribution. It connects with the popular discussion on scope (a long document covers more material than others) vs verbosity (a long document covers a similar scope to a short document but simply uses more words) [Robertson and Walker 1994]. The second addend in eq. 35 appears to capture the scope of a document. It measures the non-query terms of a document establishing whether they are simply lexical tissue or if they are significant/informative terms.

So, given this derivation, we expect that the MB model will favor the retrieval of long documents (each non-query term in a document produces an augmentation in the retrieval score). This might be problematic for collections where the distribution of lengths is varied. But if the collection tends to be uniform in length then the second addend above might become a very valuable tool to select documents whose non-query terms are significant. We anticipate that this property will be very valuable in sentence retrieval.

Let us now pay attention to the effect of the smoothing parameter μ in the retrieval formula. Low smoothing values makes the retrieval function more influenced by the matching terms (because the weights of the matching terms are higher with small μs) and less influenced by the non-query document terms (because the weights of the non-query document terms are smaller with small μs). As μ increases, the retrieval score becomes increasingly more influenced by the non-matching document terms. That is, the influence is moved

from query terms to non-query terms as μ grows. On the other hand, with low μ values the relative importance of the matching words (idf effect) is more pronounced and very high μ values tend to a coordination level ranking in the matching sum.

3.3 MBB model

The derivation for the MBB model follows a similar technique to the one presented in the last section. The query likelihood given the MBB model (eq. 24) can be re-written as:

$$P(Q|\widehat{\theta_D}) = \prod_{w_i \in Q} P(w_i|\widehat{\theta_D})^{tf(i,Q)} \cdot (1 - P(w_i|\widehat{\theta_D}))^{n_Q - tf(i,Q)} \cdot \prod_{w_i \notin Q} (1 - P(w_i|\widehat{\theta_D}))^{n_Q} \quad (36)$$

$$\log P(Q|\widehat{\theta_D}) = \sum_{w_i \in Q} \log(P(w_i|\widehat{\theta_D})^{tf(i,Q)} \cdot (1 - P(w_i|\widehat{\theta_D}))^{n_Q - tf(i,Q)}) + \sum_{w_i \notin Q} \log(1 - P(w_i|\widehat{\theta_D}))^{n_Q} \quad (37)$$

In the MBB model the probabilities for seen and unseen terms are defined as (eq. 23):

$$P_s(w_i|\widehat{\theta_D}) = \frac{tf(i,D) + \alpha_i - 1}{n_D + \alpha_i + \beta_i - 2} \quad (38)$$

$$P_u(w_i|\widehat{\theta_D}) = \frac{\alpha_i - 1}{n_D + \alpha_i + \beta_i - 2} \quad (39)$$

where $\alpha_i = \mu P(w_i|C) + 1$ and $\beta_i = \frac{1}{P(w_i|C)} + \mu(1 - P(w_i|C)) - 1$. Then, we have the MBB model retrieval function such that:

$$\begin{aligned} &\log P(Q|\widehat{\theta_D}) = \quad (40)\\ &\sum_{w_i \in Q \cap D} \log(P_s(w_i|\widehat{\theta_D})^{tf(i,Q)} \cdot (1 - P_s(w_i|\widehat{\theta_D}))^{n_Q - tf(i,Q)}) \\ &+ \sum_{w_i \in Q, w_i \notin D} \log(P_u(w_i|\widehat{\theta_D})^{tf(i,Q)} \cdot (1 - P_u(w_i|\widehat{\theta_D}))^{n_Q - tf(i,Q)}) \\ &+ \sum_{w_i \notin Q, w_i \in D} \log(1 - P_s(w_i|\widehat{\theta_D}))^{n_Q} + \sum_{w_i \notin Q, w_i \notin D} \log(1 - P_u(w_i|\widehat{\theta_D}))^{n_Q} \end{aligned}$$

which, after similar mathematical manipulation as for the MB model retrieval function, can be shown to be equivalent to the following expression:

$$\begin{aligned} &\log P(Q|\widehat{\theta_D}) = \quad (41)\\ &\sum_{w_i \in Q \cap D} \log \frac{P_s(w_i|\widehat{\theta_D})^{tf(i,Q)} \cdot (1 - P_u(w_i|\widehat{\theta_D}))^{tf(i,Q)}}{P_u(w_i|\widehat{\theta_D})^{tf(i,Q)} \cdot (1 - P_s(w_i|\widehat{\theta_D}))^{tf(i,Q)}} \\ &+ \sum_{w_i \in Q} \log \frac{P_u(w_i|\widehat{\theta_D})^{tf(i,Q)}}{(1 - P_u(w_i|\widehat{\theta_D}))^{tf(i,Q)}} \end{aligned}$$

$$+ \sum_{w_i \in D} \log \frac{(1 - P_s(w_i|\widehat{\theta_D}))^{n_Q}}{(1 - P_u(w_i|\widehat{\theta_D}))^{n_Q}} + \sum_{w_i \in V} \log(1 - P_u(w_i|\widehat{\theta_D}))^{n_Q}$$

Here, we cannot drop any addend because the P_u probabilities in the MBB model depend on the document's length. Given eqs. 38 and 39 and re-arranging the expression, results in:

$$\log P(Q|\widehat{\theta_D}) = \tag{42}$$
$$\sum_{w_i \in Q \cap D} tf(i,Q) \log \frac{(tf(i,D) + \alpha_i - 1) \cdot (n_D + \beta_i - 1)}{(\alpha_i - 1) \cdot (n_D + \beta_i - 1 - tf(i,D))}$$
$$+ \sum_{w_i \in Q} tf(i,Q) \cdot \log \frac{\alpha_i - 1}{n_D + \beta_i - 1}$$
$$+ n_Q \cdot \sum_{w_i \in D} \log(1 - \frac{tf(i,D)}{n_D + \beta_i - 1}) + n_Q \cdot \sum_{w_i \in V} \log(1 - \frac{\alpha_i - 1}{n_D + \alpha_i + \beta_i - 2})$$

The implementation of the MBB model is more complicated than either the MN or MB models but, still some pre-computation strategies can be applied to compute the model relatively efficiently:

—The value of $\sum_{w_i \in V} \log(1 - \frac{\alpha_i - 1}{n_D + \alpha_i + \beta_i - 2})$ can be computed offline for all unique lengths in the collections,

—The value of $\sum_{w_i \in D} \log(1 - \frac{tf(i,D)}{n_D + \beta_i - 1})$ can be computed offline for each document, and

—The sum of these addends can be stored in an appropriate data structure, such that at query time, the pre-computed score is multipled by the length of the query n_Q.

—At query time, the most problematic component is $\sum_{w_i \in Q} tf(i,Q) \cdot \log \frac{\alpha_i - 1}{n_D + \beta_i - 1}$. It requires that for every unique document length at query time a score be computed. This is a significant source of complexity over the MB model (and over any standard IR algorithm). The number of steps needed to compute this sum for each unique length is $dls \cdot nu_Q$, where dls is the number of unique document lengths in the collection and nu_Q is the number of unique terms in the query. Further optimizations can be designed to improve the query response times (e.g. organize the documents into chunks of similar size and compute approximate values for the probability scores, instead of exact values).

—The sum across query-document matching terms is a regular requirement of standard IR methods, which does not introduce additional penalties.

The algorithm that implements the MBB model is shown in Appendix C. To further analyze the MBB model, Equation 42 can be re-organized as follows:

$$\log P(Q|\widehat{\theta_D}) = \tag{43}$$
$$\sum_{w_i \in Q \cap D} tf(i,Q) \cdot \log \frac{tf(i,D) + \alpha_i - 1}{n_D + \beta_i - 1 - tf(i,D)}$$
$$+ \sum_{w_i \in Q, w_i \notin D} tf(i,Q) \cdot \log \frac{\alpha_i - 1}{n_D + \beta_i - 1}$$

Collection	# Docs	# Terms	# Unique Terms	Avg. doc length
TREC-8 adhoc	528155	254333060	630086	481
AQUAINT	1033461	451890396	663295	437

Table I. Statistics of the collections used in the time measurement experiments

$$+n_Q \cdot \sum_{w_i \in D} \log(1 - \frac{tf(i,D)}{n_D + \beta_i - 1}) + n_Q \cdot \sum_{w_i \in V} \log(1 - \frac{\alpha_i - 1}{n_D + \alpha_i + \beta_i - 2})$$

Although the interpretation of this formula is less intuitive than the MB formula, the MBB formulation has some interesting characteristics:

—There is still a preference for long documents. The 3rd and 4th addends tend to promote long documents, especially when queries are long. In particular, the 3rd addend promotes documents whose terms are specific (because β_i is high when $P(w_i|C)$ is low).

—The second addend, involving the query terms that were not matched provides higher weights to short documents. On the other hand, the more common the (non-matching) query terms are, the higher the score from the second addend (i.e. β_i is low and α_i is high when $P(w_i|C)$ is high). The intuitive is that it is preferable to miss query terms when they are common words.

—Unlike the MB model, the sum across matching terms involves term weights with a term frequency component. However, these term weights grow with α_i and decrease with β_i. This means that specific terms (low $P(w_i|C)$) receive lower weight than common terms. This is counterintuitive and contrasts with the typical idf behavior found in most retrieval models (and, as argued in the previous section, found also for the MB model).

While some of the addends promote long documents and others tend promote short documents this conflict between addends is dominated by the tendency to favor longer documents. The scale of values for $n_Q \cdot \sum_{w_i \in D} \log(1 - \frac{tf(i,D)}{n_D + \beta_i - 1})$ tends to be greater than $\sum_{w_i \in Q, w_i \notin D} tf(i,Q) \cdot \log \frac{\alpha_i - 1}{n_D + \beta_i - 1}$) and this is magnified proportional to the length of the query.

3.4 Time measurements

Along this section, we have presented the retrieval functions of each of the models, where we derived the retrieval functions for MB and MBB models. Our derivation shows that MB(B) models can be implemented efficiently such that they can be applied in practical settings and also provide a deeper insight in the particular features of the MB models used to compute the query likelihood. To further support these claims, we implemented the MN, MB and MBB retrieval functions and obtained time measurements for a standard document retrieval task. For each model we recorded the offline time requirements, computed as the total time taken to calculate the document dependent factors for all the document collection (e.g. to compute $log \frac{\mu}{n_D + \mu}$ in eq. 25 for the MN model, or, compute $\sum_{w_i \in D} \log \frac{\beta_i - 1}{\beta_i}$ in eq. 34 for the MB model), and the online time requirements, computed as the average time required to serve queries. These time measurements correspond to the document retrieval experiments whose retrieval effectiveness is reported in sections 4.2 and 4.3. The statistics of these collection are shown in table I. For each collection we used a set of 50 natural language topic statements (TREC topics #401-#450 for the TREC-8 adhoc collection and the HARD Robust TREC 2005 Topics for the AQUAINT collection) to form short queries

		TREC-8			AQUAINT		
		MN	MB	MBB	MN	MB	MBB
Avg. time offline (secs)		0.126	241.47	1858.48	0.274	331.57	1855.56
Avg. time per query (online)(secs)	Short	0.097	0.101	0.185	0.255	0.251	0.371
	Long	5.159	5.505	9.145	1.012	1.046	1.447

Table II. Time requirements for the Multinomial (MN), Multi-variate Bernoulli (MB) and Extended Multi-variate Bernoulli (MBB) models. The offline figures refer to the time needed to get the document dependent factors for all the document collection. The online figures correspond to the average time required to produce the ranking of documents given a query.

consisting of the title of the topics and long queries consisting of the title, description and narrative. Terms were reduced to its syntactical root with Porter's stemmer [Porter 1980] and no stopword processing was applied. These experiments were done using the Lemur toolkit [lemur].

The time measurements results are presented in table II[14]. These results confirm our claims about the time efficiency of the MB model. Although the MB model's offline computations take more time than the MN's offline computations (the MN model simply needs the document length to get $\log \frac{\mu}{n_D+\mu}$ whereas the MB model requires to go on every document term to compute $\sum_{w_i \in D} \log \frac{\beta_i - 1}{\beta_i}$), the online time performance is roughly the same. This demonstrates that the MB model can be fully operative in a realistic setting. On the other hand, the MBB model takes significant time resources offline but, more importantly, its online performance is roughly half of the MN/MB online performance. As argued above, the requirement to go on every unique document length in the collection at query time is the major factor explaining such time penalty.

These efficient implementations of the MB(B) models are a huge gain w.r.t a direct implementation of such models. As a matter of fact, we also implemented a *direct* version of these models (traversing all the vocabulary terms at query time) and it took several hours per query.

4. EMPIRICAL COMPARISON

In the previous sections, we analyzed formally the MN and MB(B) models in the Bayesian framework and the subsequent retrieval functions, showing their differences theoretically and computationally. Now, we present a thorough experimental analysis of the MN and MB(B) models for different tasks and collections. This attempts to relate a complete contextualization for the MB(B) models that helps to locate their actual role in future IR systems. To study the behavior of MN and MB(B) models under very different scenarios, this section is broken up into a series of experiments organized into three parts: sentence retrieval, document retrieval and retrieval of summaries. In particular, our aim is to identify retrieval scenarios where the MB model is most appropriate and competitive. We believe that the special features of this model (and, particularly, its ability to select pieces of texts not only based on the matching terms but also based on how specific the (non-query) document/sentence terms are) have been under exploited.

[14] The TREC8 experiments were executed in a Intel Pentium III (1.4 Ghz) machine, with 2Gb RAM and running Debian GNU/Linux version 4.0. The AQUAINT experiments were run in a multimode cluster (40 dualcore itanium Montecity operating at 1.43 Ghz) with 320GB of RAM running Suse linux enterprise server 10.

In all our experiments we indexed the collections using the Lemur toolkit [lemur], developing our own extensions where needed. Since, the MB and MBB models are not provided with Lemur we implemented them using the efficient methods proposed in the previous section. For sentence retrieval, we also developed the code needed to transform the input data (tagged sentences) into a format readable by Lemur. In all our experiments, no stopword processing was done because the effects of stopword removal should be better achieved by exploiting language modeling techniques [Zhai and Lafferty 2001b], which can naturally cope with words having very different patterns of usage within the language/corpus. In this way, the comparison between Multinomial and multiple-Bernoulli models is not biased by any artificial choice of common words. Terms were reduced to its syntactical root with Porter's stemmer [Porter 1980].

The statistical language models were defined as follows. A language model is defined for each sentence/summary/document (using eqs. 6, 17 and 23, respectively) and models were smoothed using $\alpha_i = \mu P(w_i|C) + 1$ for the Multinomial model (i.e. Dirichlet smoothing) and $\alpha_i = \mu P(w_i|C) + 1, \beta_i = \frac{1}{P(w_i|C)} + \mu(1 - P(w_i|C)) - 1$ for the two multi-variate Bernoulli models (MB and MBB). Different values of the smoothing parameter μ were tested for each model. Final scoring was performed using the retrieval functions associated to each query likelihood.

For document retrieval and retrieval based on summaries of text, the main performance measures analyzed are: mean average precision (MAP), number of relevant documents retrieved, and precision at 10 docs. For sentence retrieval we adopt the evaluation methodology designed for the TREC novelty tracks, where the F measure is the main performance ratio. To examine the differences between the models in more detail we also applied significance tests between the best runs. These were performed using the Wilcoxon paired Sign test with a significance level set to 0.05. Across this section, a statistically significant difference between retrieval model A and retrieval model B will be denoted as $A >> B$ (meaning that retrieval model A is significantly better than B), whereas $A \approx B$ denotes that there is no statistically significant difference.

4.1 Sentence retrieval

Sentence retrieval is an important IR task, whose applications span a wide range of research topics such as question answering, topic detection and tracking, document or multi-document summarization [Murdock 2006]. To locate sentences relevant to a query is an interesting retrieval task whose characteristics might be adequate for a formulation based on a MB distribution. First of all, the lack of a non-binary term frequency component in the multiple-Bernoulli models seems less important for this task because sentences are short pieces of text. Second, the retrieval functions derived from the multiple-Bernoulli models look especially promising for sentence retrieval. In particular, the ability of MB models to select texts whose non-query terms are specific fits well here. In document retrieval this effect might be harming because it might promote unfairly long texts. In sentence retrieval the distribution of lengths is much narrower and we expect that the MB models would work well here to select sentences that contain some query terms but also other significative terms.

In our experiments, we used sentence retrieval benchmark collections from three TREC novelty tracks (2002, 2003, 2004). In this task, the basic input data is a set of TREC topics and a set of relevant documents for each topic. These relevant documents are selected from actual results from an effective retrieval algorithm. If there are 25 or fewer relevant

documents for the topic, then all the relevant documents are used. If there are more than 25 relevant documents, the top 25 ranked ones from that run are taken. Participants in the novelty track had to 1) locate those sentences in the relevant documents which are relevant to the topic and 2) filter out those sentences containing redundant material. The purpose of the second step is to provide the user with relevant and *novel* sentences[15]. We focus here our interest in the first stage: sentence retrieval. For each topic, the information available is the text of the topic itself and a rank containing at most 25 relevant documents. Hence, this is a task where the search space (i.e. the set of sentences in these relevant documents) is much smaller than typical search spaces in IR. As argued above, our hypothesis is that the multiple-Bernoulli models could be competitive for sentence retrieval in this scenario.

The characteristics of the datasets in the TREC-2002, TREC-2003 and TREC-2004 novelty collections [Harman 2002; Soboroff and Harman 2003; Soboroff 2004] are very different to each other. We therefore could test the two language modeling approaches under different circumstances. In the TREC-2002 novelty data, the topics and relevant documents were taken from old TREC tracks (TREC-6, TREC-7 and TREC-8) and a very low percentage of the sentences retrieved in the relevant documents were actually relevant (median around 2%). On the other hand, TREC-2003 and TREC-2004 participants used the AQUAINT collection and the topics were constructed specifically for the task. The average percentage of relevant sentences was around 40% and 20% in TREC-2003 and TREC-2004, respectively. Unlike the previous datasets, in the TREC-2004 data the set of retrieved documents available for each query contains some non-relevant documents. The process to obtain the set of relevant documents (i.e. the input to the novelty participants) and the methods to get the relevance judgments were also modified from 2003 [Soboroff and Harman 2003].

Queries were constructed from the TREC topics taking all its subfields (title, description and narrative). The background model $P(.|C)$ used to smooth with was built from the set of documents available for the task (1080, 1242, 1808 and documents in TREC-2002, TREC-2003 and TREC-2004, respectively). This simulates an environment in which a large reference collection is not available and the system has to smooth the probabilities from a small set of documents. Indeed, this is the strictest interpretation of this sentence retrieval task[16]. The performance of sentence retrieval algorithms is measured combining sentence set recall and precision through the F measure:

$$F = \frac{2 \times P \times R}{P + R} \tag{44}$$

where P is the fraction of retrieved sentences which are relevant and R is the fraction of the relevant sentences which are retrieved. This is a consistent performance ratio because it is meaningful even when the number of relevant sentences varies widely across topics [Harman 2002]. In our experiments, we used the top 5% of retrieved sentences for evaluating the TREC-2002 runs, and the top 70% (50%) of retrieved sentences for TREC-2003 (TREC-2004). For sentence retrieval applications it will be important to determine an ap-

[15] The notion of novelty is captured assuming that the user knows nothing at the time of the initial retrieval and all learning happens in the order of sentence retrieval.

[16] For instance, the group from the University of Massachusetts participating in the novelty track [Larkey et al. 2002] tested a tf/idf approach in which collection statistics (e.g. number of sentences in which a term appears) are taken from this small set of (relevant) documents.

	μ									
	10	100	1k	2k	3k	4k	5k	10k	50k	100k
MN	.089	.185	**.208**	.207	.205	.206	.207	.204	.202	.202
MB	.197	.209	**.215**	.212	.209	.213	.211	.205	.201	.203
MBB	.193	.211	.213	**.216**	.211	.211	.210	.209	.201	.202

Table III. Sentence retrieval (TREC-2002): The performance (F-measure) of the MN, MB and MBB models given μ.

propriate threshold but threshold tuning was not an objective here. The participants in the TREC novelty track were provided with a set of training topics to adjust their systems (and, especially, set the thresholds). We did not apply any threshold training but instead adopted the standard thresholds used in the respective tracks.

Table III reports the performance results for the MN and MB(B) models in the TREC-2002 sentence retrieval problem (novelty track). Different experiments were run with varying values of the smoothing parameter μ (best results in bold). The best results are found at similar levels of smoothing for both models. The MB model shows here a very consistent behavior. Its best run is slightly superior to the best Multinomial run and, furthermore, the improvement with respect to the Multinomial model is very consistent across most smoothing levels (in 9 out of 10 runs the MB model got better performance than the MN model and in 8 out of 10 runs the MBB model got better performance than the MN model). It is also remarkable that the difference in performance is especially large at low levels of smoothing ($\mu \leq 100$). Nevertheless, the significance tests showed that the one-to-one differences between the best results attainable by each model are not statistically significant: $MN \approx MB \approx MBB$. Unlike other IR tasks, the Multinomial model does not outperform the MB models. Moreover, the MB(B) models appear more consistent; invariant to different levels of smoothing (i.e. changes in μ, the average performance across MN runs is .192 whereas the average performance across MB and MBB runs is .208 and .208, respectively).

Similar experiments were repeated for TREC-2003 and TREC-2004 data. Results are shown in tables IV and V. Again, the behavior of the MB model is very competitive and, indeed, in both cases, the worst MB model is better than the best Multinomial run. Regarding low levels of smoothing, the same trend as in TREC-2002 data was found: the difference in performance is larger at low degrees of smoothing. These results, along with the ones reported for the TREC-2002 data, indicate clearly that the MB model is competitive for sentence retrieval and, actually, it retrieves relevant sentences more effectively than the Multinomial model. Regarding the MBB model, it performs slightly worse than the MB model but its worst runs are still as good as the best Multinomial runs. Again, the non-binary term frequency incorporated into the MBB model does not appear to be of significant benefit given the problem of sentence retrieval. The significance tests for TREC-2003 data show that $MB >> MBB >> MN$, whereas we obtained $MB \approx MBB >> MN$ on the TREC-2004 data.

This demonstrates the effectiveness of the implicit correction within the MB models. The MB retrieval function involves a promotion for sentences whose non-query terms are significant (high idf). On the other hand, there is not significant difference between MB and MBB. This provides some evidence to confirm the hypothesis: term frequency is not important for the sentence retrieval problem and, shows that the MB model, which simply handles a binary notion of term frequency, delivers comparable performance.

	μ									
	10	100	1k	2k	3k	4k	5k	10k	50k	100k
MN	.515	.540	.571	.573	**.574**	.574	.574	.574	.574	.574
MB	**.604**	.595	.588	.586	.586	.586	.585	.585	.585	.585
MBB	**.598**	.588	.581	.580	.579	.579	.579	.578	.576	.576

Table IV. Sentence retrieval (TREC-2003): The performance (F-measure) of the MN, MB and MBB models given μ.

	μ									
	10	100	1k	2k	3k	4k	5k	10k	50k	100k
MN	.336	.380	.393	.394	.394	**.395**	.395	.395	.395	.395
MB	**.409**	.406	.402	.401	.401	.401	.401	.400	.400	.399
MBB	**.408**	.403	.398	.397	.397	.397	.397	.396	.395	.395

Table V. Sentence retrieval (TREC-2004): The performance (F-measure) of the MN, MB and MBB models given μ.

These results confirm our initial hypothesis about the adequacy of MB models for sentence retrieval problems. Our experiments, involving different collections with varying percentage of relevant sentences, have shown that the MB models perform at least as well as the MN model. Indeed, the popular MN model was only comparable to the MB models with the TREC-2002 collection, which is a particularly difficult track/collection because the amount of relevant sentences is extremely low.

The multi-variate Bernoulli's retrieval function seems to be good at isolating the relevant sentences from the non-relevant ones. If a sentence does not contain significant terms (other than the query words) then the sentence will be penalized. Hence, retrieval using MB is not only a matter of covering the query topics but also specifying additional (and significant) material. This effect cannot be obtained through a Multinomial model. Other retrieval tasks, whose granularity is fine, could be also a good application scenario for LMs with Bernoulli-based formulations.

These experimental results are promising and, indeed, to the best of our knowledge, this is the first time that the MB model has been shown to outperform the standard Multinomial model. Since the *advanced* multiple Bernoulli model (MBB) is also better than the Multinomial model, it seems that the form of the multi-variate Bernoulli retrieval functions (eqs 35 and 43) is actually very effective at isolating the relevant material.

4.1.1 *Additional sentence retrieval experiments: MB model without the non-query terms component.* The reader might be wondering whether or not the good behavior of the MB model can be solely attributed to its binary-based estimation procedures rather than to the effect of the non-query terms component. To clarify this issue we conducted additional experiments with a new retrieval function based on the MB likelihood but without the non-query terms component (in the following this model will be referred to as MBWNQT). More specifically, the form of this *likelihood* is:

$$P(Q|\widehat{\theta_D}) = \prod_{w_i \in Q} P(w_i|\widehat{\theta_D}) \tag{45}$$

Following a derivation similar to the one shown in section 3.2, it can be proved that the

	μ									
	10	100	1k	2k	3k	4k	5k	10k	50k	100k
TREC-2002										
MB	.197	.209	**.215**	.212	.209	.213	.211	.205	.201	.203
MBWNQT	.206	.215	.215	.216	.216	**.218**	**.218**	.216	.212	.210
TREC-2003										
MB	**.604**	.595	.588	.586	.586	.586	.585	.585	.585	.585
MBWNQT	**.598**	.587	.582	.581	.581	.581	.581	.580	.580	.580
TREC-2004										
MB	**.409**	.406	.402	.401	.401	.401	.401	.400	.400	.399
MBWNQT	**.409**	.406	.400	.399	.398	.399	.398	.398	.399	.399

Table VI. Comparing the MB model with and without the non-query term component

subsequent retrieval function for this model is:

$$\log P(Q|\theta_D) \approx \sum_{w_i \in Q \cap D} \log \frac{\alpha_i}{\alpha_i - 1} \tag{46}$$

This model incorporates the estimations derived from the original MB model but skips the non-query terms component and, therefore, comparing it against the MB model we can check whether the MB estimations applied on the matching terms are enough to achieve optimal performance.

In table VI we compare the performance of the MB model against this new model for the three sentence retrieval collections. These results show that against the MB Model, the MBWNQT model performs (1) better on TREC-2002, (2) similar on TREC-2004, and (3) worse on TREC 2003. If we consider the differences between the TREC 2002-2004 in terms of the relevant material in each, we see that the percentage of relevant sentences, in TREC 2002 was 2%, TREC 2004 was 20%, and TREC 2003 was 40%. So as the relevant material in the collection increased the performance of the MB model improved. It would appear that if there are many non-relevant sentences, the MB model is likely to promote such non-relevant material because the model rewards sentences with informative terms (despite, regardless of whether the content is off topic, which is likely to be case when there is little relevant material available).

This comparison suggests that the non-query terms component might be avoided when the amount of relevant material is expected to be scarce. On the contrary, the non-query terms component appears to be a solid sentence retrieval tool when there are many relevant sentences. In such cases, the ability to select sentences that are not only good matches with the query but also good containers of additional terms is an interesting feature of the MB model.

4.2 Document retrieval

The next pool of experiments involved a typical document retrieval problem. The LM approaches were evaluated under this scenario. The retrieval task was the last available TREC ad-hoc track, which took place in TREC-8 [Voorhees and Harman 1999]. The data collection consists of approximately 2 gigabytes of text from TREC disks #4 and #5 (documents from the Financial Times, Federal Register, Foreign Broadcast Information Service and Los Angeles Times). Table VII depicts the basic statistics of the indexed collection.

A set of 50 natural language topic statements was available for the task (topics #401-#450) and, as usual, the top 1000 documents retrieved for each topic were used for evaluation. We report here three main performance measures, namely: mean average precision (MAP), number of relevant documents retrieved (r.r.d) and precision at 10 docs (p@10). In the experiments we used short and long queries. Short queries were formed from the TREC topic taking only the title subfield whereas long queries were obtained from all topic subfield (title, description and narrative) as done in [Zhai and Lafferty 2004].

Results for short queries are shown in table VIII. The first clear conclusion we can draw is that the regular MB model is not competitive here. This is not very surprising because, as argued before, some experimental results are available in the literature [Metzler et al. 2004] with similar outcomes. Looking at the performance at different levels of smoothing, we observe that the MB model is quite stable across all levels. Nevertheless, the worst Multinomial run ($\mu = 100k$) is still slightly better than any MB run, in terms of MAP, number of relevant retrieved documents and p@10. Regarding the MBB model, its performance is significantly better than the performance of the MB model and it is indeed comparable to the Multinomial model. After applying the sign test to compare the MAP values of the best runs, we observed that $MN \approx MBB >> MB$. As anticipated in section 3, the implicit promotion of long documents and the lack of a non-binary term frequency component is adversely affecting the performance of document retrieval.

The experimental results for long queries are summarized in table IX. While the Multinomial model maintains its best performance (and the best run even improves some of the ratios), both multi-variate Bernoulli models perform significantly worse with long queries. Furthermore, the stable behavior previously witnessed has now dissipated. Indeed, the performance for the MB model is especially poor at low smoothing levels. The significance tests between the best runs confirm that the MN model is clearly superior: $MN >> MBB >> MB$.

Leaving aside comparison with the Multinomial models for the time being (as clearly, a multiple Bernoulli distribution is not suitable for document retrieval with long queries), it is worth analyzing the reasons behind such poor performance. In the MB and MBB models the longer the query is the more long documents are retrieved. In the MB model (eq. 35), the weight given to a matching term ($\log \frac{\alpha_i}{\alpha_i - 1}$) is always greater than the weight given to a non-query document term ($\log \frac{\beta_i - 1}{\beta_i}$). Long queries give more weight to the query-document matching sum than short queries (because there are more query terms and less non-query terms). Also, this makes longer documents more favorable over shorter documents. Note also that the optimal μ value in the MB model is much higher with long queries. This is because the smoothing applied tends to compensate for the excessive promotion of long documents (recall that as μ increases the importance is moved from query terms to non-query terms). This shows that smoothing has also a conflicting role here: one for correcting document length (μ high means that less long documents are retrieved) and one for estimating the informativeness of query words (μ low means that more importance is given to the discriminative power of the terms -i.e. μ low means more idf effect on query terms). Long queries would need low μ to account for the high variance

# Docs	# Terms	# Unique Terms	Avg. doc length
528155	254333060	630086	481

Table VII. TREC-8 adhoc collection - Statistics

		μ									
Model	Measure	10	100	1k	2k	3k	4k	5k	10k	50k	100k
MN	*MAP*	.2245	.2367	**.2509**	.2473	.2433	.2398	.2375	.2277	.2037	.1959
	# r.r.d	2575	2666	**2824**	2806	2781	2753	2733	2647	2466	2438
	p@10	.4280	.4320	.4400	**.4580**	.4540	.4420	.4320	.4000	.3360	.3240
MB	*MAP*	.1816	.1834	.1845	.1849	.1849	.1854	**.1860**	.1859	.1860	.1847
	# r.r.d	2366	**2370**	2332	2329	2327	2327	2327	2323	2324	2326
	p@10	.3020	.3080	.3140	.3120	.3120	.3100	.3140	**.3180**	.3140	.3140
MBB	*MAP*	**.2442**	.2440	.2417	.2397	.2363	.2338	.2317	.2227	.2021	.1945
	# r.r.d	2753	**2762**	2758	2743	2728	2711	2699	2644	2469	2439
	p@10	.3000	**.4400**	.4360	.4320	.4260	.4120	.4120	.3940	.3320	.3180

Table VIII. Document retrieval on TREC8: MN, MB and MBB with short queries

		μ									
Model	Measure	10	100	1k	2k	3k	4k	5k	10k	50k	100k
MN	*MAP*	.1360	.1685	.2460	**.2517**	.2498	.2468	.2445	.2341	.2057	.1970
	# r.r.d	1590	1899	2705	**2792**	2782	2789	2782	2740	2604	2547
	p@10	.2880	.3820	**.4500**	.4480	.4480	.4300	.4320	.3980	.3140	.3140
MB	*MAP*	.0158	.0215	.0408	.0536	.0670	.0757	.0831	.1085	.1409	**.1430**
	# r.r.d	442	539	823	993	1104	1175	1249	1467	**1688**	1660
	p@10	.0640	.0660	.0920	.1060	.1280	.1440	.1520	.1800	**.2000**	.1980
MBB	*MAP*	.0211	.0306	.0657	.0817	.0916	.0989	.1059	.1219	**.1443**	.1440
	# r.r.d	540	706	1120	1298	1388	1473	1545	1766	**2089**	2077
	p@10	.0720	.0920	.1460	.1740	.1880	.1920	.1960	.2220	**.2280**	.2180

Table IX. Document retrieval on TREC8: MN, MB and MBB with long queries

in the quality of the terms but this would also encourage longer documents to be retrieved. In the MBB model (eq. 43), a similar situation arises. Note that the MN model corrects this length effect with a penalty for long texts which depends on the query length (eq. 25). The longer the query is the higher preference for short texts under the MN model. The performance statistics shown in the table indicates that the promotion of long documents given long queries leads to poorer retrieval performance for the MB models.

Recall that the MB models also favor documents that contain non-query terms with high idf. So if queries are short then many high idf non query terms indicate that the document contains more informative content; some of which is related to the topic and some of which is not. However, if the queries are long, more of the high idf terms related to the topic are explicitly mentioned, thus the non query terms which have high idf, while informative, are more likely to be unrelated to the topic. This means that, given a document, most of its non-query terms will be off-topic and, therefore, the sum across non-query terms (second addend in eq. 35) might be adversely affecting the system performance, because documents may be promoted because of its match on non-query terms, which are unlikely to be related to the query topic. For the time being, we shall defer any further discussion until section 5, where these issues will be carefully revisited, given the results obtained for all the tasks.

4.3 Summary Retrieval

To contrast between the retrieval of sentences and the retrieval of full text, here we consider the retrieval of documents based on summaries of text (i.e. the document representation is a summary of the document). The main goal of these experiments is to examine how the sparsity of the representation affects the retrieval approaches. Since the MB approaches are based on binary representations then we posit that when summaries are shorter then they will tend to perform better. In contrast, the MN model, which relies on the counts

Summary	# Docs	# Terms	# Unique Terms	Avg. doc length
Short	1033461	15449698	93376	15
Medium	1033461	51134199	206264	50
Large	1033461	100180609	233760	96
Full	1033461	451890396	663295	437

Table X. AQUAINT Ad Hoc Collection Statistics for each representation

of terms to obtain a better statistical estimate of the data, will perform better with larger summaries. Also, in each pool of experiments, the sizes of the summaries are uniform and, therefore, this retrieval problem might be beneficial to the MB(B) models.

This retrieval task was performed on the latest TREC HARD Track 2005, which consists of the AQUAINT news collection and HARD Robust TREC Topics. Summaries of the news articles within the collection were formed by extracting the first n terms from the document (usually referred to as lead sentence summarization). Since news articles are written in such a way as to summarize and repeat their contents, it is common to use such summaries as a baseline when comparing summarization techniques [Stokes et al. 2004]. We selected n to be 15, 50 and 100 (labeled as small, medium and large summaries, respectively). We also used the full text documents. Table X shows the basic statistics of the indexed collection, given the different summary lengths. The fifty HARD Robust 2005 topics were used to form short queries consisting of the title of the topics and long queries consisting of the title, description and narrative.

		μ									
Model	Measure	10	100	1k	2k	3k	4k	5k	10k	50k	100k
Small Summary											
MN	*MAP*	**0.052**	0.052	0.051	0.050	0.049	0.048	0.048	0.042	0.040	0.040
	#r.r.d	**1598**	1592	1597	1565	1565	1565	1565	1524	1523	1523
	p@10	**0.206**	0.206	0.204	0.202	0.198	0.198	0.198	0.182	0.148	0.142
MB	*MAP*	**0.049**	0.048	0.047	0.047	0.047	0.047	0.047	0.045	0.046	0.045
	#r.r.d	1525	1512	1545	**1551**	1550	1539	1539	1511	1503	1491
	p@10	**0.182**	0.178	0.164	0.168	0.172	0.166	0.168	0.164	0.172	0.168
MBB	*MAP*	**0.054**	0.053	0.052	0.050	0.048	0.048	0.048	0.041	0.038	0.039
	#r.r.d	**1579**	1574	1585	1564	1562	1557	1556	1514	1512	1514
	p@10	**0.226**	0.214	0.210	0.204	0.196	0.194	0.196	0.18	0.146	0.148
Medium Summary											
MN	*MAP*	0.082	**0.083**	0.083	0.082	0.079	0.077	0.075	0.071	0.064	0.061
	#r.r.d	2346	2338	**2349**	2333	2297	2249	2242	2236	2181	2170
	p@10	0.256	0.260	**0.262**	0.254	0.248	0.238	0.242	0.216	0.184	0.174
MB	*MAP*	**0.058**	0.058	0.058	0.058	0.058	0.056	0.056	0.057	0.057	0.058
	#r.r.d	2061	2069	2054	**2074**	2065	1987	1995	2001	2000	1993
	p@10	0.19	0.196	0.206	**0.212**	0.208	0.196	0.202	0.198	0.198	0.188
MBB	*MAP*	**0.082**	0.082	0.082	0.081	0.078	0.076	0.074	0.070	0.062	0.061
	#r.r.d	2276	2273	**2286**	2285	2259	2205	2197	2191	2133	2117
	p@10	0.258	0.254	**0.260**	0.250	0.250	0.242	0.242	0.214	0.182	0.170
Large Summary											
MN	*MAP*	0.100	0.102	**0.103**	0.101	0.100	0.097	0.095	0.088	0.074	0.070
	#r.r.d	2734	2753	**2759**	2692	2681	2653	2635	2523	2390	2382
	p@10	0.278	0.282	0.280	**0.288**	0.270	0.260	0.254	0.226	0.178	0.160
MB	*MAP*	0.059	0.059	0.059	0.059	0.060	**0.061**	0.061	0.061	0.061	0.061
	#r.r.d	2273	**2275**	2269	2222	2245	2248	2249	2247	2230	2236
	p@10	0.188	0.198	**0.204**	0.204	0.186	0.182	0.184	0.190	0.176	0.182
MBB	*MAP*	0.101	0.101	**0.102**	0.101	0.100	0.097	0.095	0.087	0.074	0.070
	#r.r.d	2737	**2738**	2731	2688	2674	2663	2637	2520	2384	2375
	p@10	**0.290**	0.290	0.280	0.276	0.266	0.258	0.252	0.230	0.180	0.154
Full Document											
MN	*MAP*	0.137	0.157	0.193	**0.196**	0.195	0.192	0.189	0.176	0.135	0.123
	#r.r.d	3398	3651	4046	**4127**	4114	4081	4060	3920	3500	3368
	p@10	0.29	0.344	0.422	0.428	0.434	0.438	**0.452**	0.426	0.348	0.322
MB	*MAP*	0.068	0.069	0.070	0.072	0.073	0.073	0.074	0.074	**0.075**	0.075
	#r.r.d	2656	2659	2664	2675	2687	2694	2725	2720	2728	**2736**
	p@10	0.144	0.152	0.158	0.156	0.156	0.152	0.152	0.154	**0.158**	0.156
MBB	*MAP*	0.177	0.181	0.193	**0.197**	0.196	0.194	0.192	0.181	0.143	0.126
	#r.r.d	3723	3762	4026	**4127**	4117	4097	4071	3966	3525	3393
	p@10	0.444	0.420	0.426	0.430	0.434	0.434	**0.446**	0.430	0.362	0.326

Table XI. Summary retrieval on AQUAINT (TREC HARD Robust 2005): MB, MBB and MN with short queries

Model	Measure	μ 10	100	1k	2k	3k	4k	5k	10k	50k	100k
						Small Summary					
MN	*MAP*	**0.050**	0.049	0.043	0.040	0.038	0.037	0.035	0.031	0.027	0.025
	#r.r.d	1675	**1714**	1682	1637	1561	1575	1519	1409	1282	1287
	p@10	**0.218**	0.212	0.188	0.174	0.154	0.144	0.142	0.120	0.100	0.100
MB	*MAP*	**0.022**	0.022	0.020	0.019	0.019	0.019	0.018	0.016	0.013	0.013
	#r.r.d	1110	**1178**	1017	963	953	971	956	930	888	879
	p@10	**0.142**	0.132	0.114	0.106	0.104	0.094	0.092	0.084	0.070	0.054
MBB	*MAP*	**0.023**	0.023	0.022	0.021	0.020	0.020	0.019	0.016	0.013	0.012
	#r.r.d	1133	**1151**	1080	1055	1028	1017	1015	927	850	831
	p@10	**0.140**	0.132	0.114	0.120	0.118	0.108	0.096	0.086	0.052	0.046
						Medium Summary					
MN	*MAP*	0.064	**0.079**	0.078	0.076	0.074	0.072	0.071	0.066	0.056	0.054
	#r.r.d	2110	2336	**2355**	2309	2266	2199	2179	2121	1831	1773
	p@10	0.250	**0.294**	0.290	0.274	0.258	0.254	0.252	0.226	0.152	0.142
MB	*MAP*	0.034	**0.037**	0.037	0.037	0.037	0.037	0.036	0.032	0.026	0.024
	#r.r.d	1375	1400	**1418**	1372	1320	1283	1207	1002	816	732
	p@10	0.164	**0.188**	0.176	0.160	0.158	0.160	0.150	0.134	0.118	0.110
MBB	*MAP*	0.037	0.042	0.046	**0.047**	0.047	0.047	0.047	0.046	0.037	0.034
	#r.r.d	1406	1534	1717	**1718**	1672	1604	1570	1484	1237	1212
	p@10	0.182	**0.212**	0.206	0.192	0.188	0.192	0.192	0.184	0.140	0.116
						Large Summary					
MN	*MAP*	0.056	0.093	0.100	**0.101**	0.101	0.099	0.098	0.093	0.078	0.073
	#r.r.d	2174	2758	2826	**2837**	2805	2764	2745	2679	2455	2351
	p@10	0.230	**0.312**	0.306	0.288	0.282	0.278	0.264	0.250	0.184	0.160
MB	*MAP*	0.040	0.044	**0.049**	0.049	0.049	0.049	0.049	0.048	0.041	0.037
	#r.r.d	1458	1606	**1650**	1625	1585	1581	1557	1479	1183	1119
	p@10	0.164	0.152	**0.168**	0.156	0.154	0.152	0.154	0.142	0.134	0.114
MBB	*MAP*	0.043	0.050	0.059	0.061	0.062	0.063	0.063	**0.064**	0.058	0.052
	#r.r.d	1620	1792	1992	2042	2053	**2062**	2058	2056	1747	1679
	p@10	0.164	0.186	0.174	0.168	0.164	**0.176**	0.168	0.164	0.146	0.146
						Full Document					
MN	*MAP*	0.069	0.113	0.198	0.210	**0.211**	0.209	0.206	0.196	0.166	0.152
	#r.r.d	2364	3040	4128	4266	**4283**	4242	4199	4098	3751	3625
	p@10	0.220	0.334	0.412	0.440	0.436	**0.444**	0.438	0.418	0.380	0.340
MB	*MAP*	0.030	0.035	0.049	0.055	0.058	0.060	0.062	**0.066**	0.066	0.058
	#r.r.d	1277	1389	1646	1727	1765	1793	1840	**1889**	1820	1779
	p@10	0.126	0.128	0.160	0.172	0.170	**0.172**	0.172	0.166	0.162	0.150
MBB	*MAP*	0.047	0.076	0.114	0.124	0.129	0.132	0.134	0.141	**0.144**	0.134
	#r.r.d	1694	2228	2869	3023	3091	3132	3171	**3283**	3156	3003
	p@10	0.168	0.238	0.290	0.304	0.324	0.322	0.320	**0.342**	0.326	0.308

Table XII. Summary retrieval on AQUAINT (TREC HARD Robust 2005): MB, MBB and MN with long queries

In Tables XI and XII we report the three main performance measures. Again, the best performing runs for each model given the smoothing parameter μ are shown in bold. From these tables a few general trends can be identified. As the summary length increases the retrieval performance improves for all models and types of queries. The performance of the MB(B) models dropped when long queries were employed, whereas the performance of the MN model increased. More smoothing is required as the summary length increases to attain the best performance. The MN model tends to provide good performance regardless of the summary length or query length. The MBB model is comparable to the MN model on short queries, but not so for long queries. While the MB model performs poorly on most occasions.

Table XIII provides a summary of the significance tests applied between the MAP values of the best performing retrieval models given μ. The significance tests confirm these general intuitions with a few notable exceptions. When the representation and the query length was short there was no difference between the models. When the query length was increased the MBB whilst better than the MB performed worse than the MN.

In these experiments, the retrieval of documents was based on summaries of the document text. This means that these experiments are oriented to evaluate the ability of the models to predict relevance given a portion of the document based on a smaller (but the same amount) of sample data for each document. Thus as the summaries are increased the accuracy of the predictions are improved (because there is more data upon which to estimate the document language models). Only, MN and MBB model take advantage of this as they cater for term frequency, whereas the MB model does not cater for term frequency

	Type of Query	
Summary	**Short**	**Long**
Small	$MN \approx MBB \approx MB$	$MN >> MBB \approx MB$
Medium	$MN \approx MBB >> MB$	$MN >> MBB >> MB$
Large	$MN \approx MBB >> MB$	$MN >> MBB >> MB$
Full	$MN \approx MBB >> MB$	$MN >> MBB >> MB$

Table XIII. Summary retrieval on AQUAINT (TREC HARD Robust 2005): results of significance tests between best runs on MAP

		μ									
Model	Measure	10	100	1k	2k	3k	4k	5k	10k	50k	100k
						Small Summary					
MB	*MAP*	**0.049**	0.048	0.047	0.047	0.047	0.047	0.047	0.045	0.046	0.045
	#r.r.d	1525	1512	1545	**1551**	1550	1539	1539	1511	1503	1491
	p@10	**0.182**	0.178	0.164	0.168	0.172	0.166	0.168	0.164	0.172	0.168
MBWNQT	*MAP*	**0.057**	0.057	0.056	0.056	0.056	0.056	0.056	0.055	0.055	0.055
	#r.r.d	**1524**	1511	1510	1510	1510	1510	1510	1475	1474	1474
	p@10	**0.222**	0.222	0.222	0.222	0.222	0.222	0.222	0.222	0.222	0.222
						Medium Summary					
MB	*MAP*	**0.058**	0.058	0.058	0.058	0.058	0.056	0.056	0.057	0.057	0.058
	#r.r.d	2061	2069	2054	**2074**	2065	1987	1995	2001	2000	1993
	p@10	0.19	0.196	0.206	**0.212**	0.208	0.196	0.202	0.198	0.198	0.188
MBWNQT	*MAP*	**0.064**	0.064	0.064	0.064	0.064	0.062	0.062	0.062	0.062	0.062
	#r.r.d	**2287**	2287	2256	2255	2253	2158	2160	2167	2167	2167
	p@10	0.186	**0.190**	0.190	0.190	0.190	0.190	0.190	0.190	0.190	0.190
						Large Summary					
MB	*MAP*	0.059	0.059	0.059	0.059	0.060	**0.061**	0.061	0.061	0.061	0.061
	#r.r.d	2273	**2275**	2269	2222	2245	2248	2249	2247	2230	2236
	p@10	0.188	0.198	**0.204**	0.204	0.186	0.182	0.184	0.19	0.176	0.182
MBWNQT	*MAP*	**0.064**	0.064	0.064	0.063	0.064	0.064	0.064	0.064	0.063	0.063
	#r.r.d	**2477**	2477	2447	2383	2405	2406	2403	2402	2369	2364
	p@10	**0.168**	0.168	0.168	0.168	0.168	0.168	0.168	0.168	0.168	0.168
						Full Document					
MB	*MAP*	0.068	0.069	0.070	0.072	0.073	0.073	0.074	0.074	**0.075**	0.075
	#r.r.d	2656	2659	2664	2675	2687	2694	2725	2720	2728	**2736**
	p@10	0.144	0.152	0.158	0.156	0.156	0.152	0.152	0.154	**0.158**	0.156
MBWNQT	*MAP*	0.058	**0.059**	0.059	0.059	0.059	0.059	0.058	0.058	0.058	0.058
	#r.r.d	2440	2444	2444	2444	2444	2444	**2446**	2446	2446	2446
	p@10	**0.122**	0.122	0.122	0.122	0.122	0.122	0.122	0.122	0.122	0.122

Table XIV. Summary retrieval on AQUAINT (TREC HARD Robust 2005): MB vs MBWNQT with short queries

		μ									
Model	Measure	10	100	1k	2k	3k	4k	5k	10k	50k	100k
						Small Summary					
MB	*MAP*	**0.022**	0.022	0.020	0.019	0.019	0.019	0.018	0.016	0.013	0.013
	#r.r.d	1110	**1178**	1017	963	953	971	956	930	888	879
	p@10	**0.142**	0.132	0.114	0.106	0.104	0.094	0.092	0.084	0.070	0.054
MBWNQT	*MAP*	**0.057**	0.056	0.052	0.048	0.048	0.046	0.045	0.041	0.040	0.040
	#r.r.d	1682	**1688**	1628	1580	1590	1563	1549	1434	1324	1309
	p@10	**0.248**	0.238	0.212	0.188	0.182	0.170	0.172	0.166	0.160	0.156
						Medium Summary					
MB	*MAP*	0.034	**0.037**	0.037	0.037	0.037	0.037	0.036	0.032	0.026	0.024
	#r.r.d	1375	1400	**1418**	1372	1320	1283	1207	1002	816	732
	p@10	0.164	**0.188**	0.176	0.160	0.158	0.160	0.150	0.134	0.118	0.110
MBWNQT	*MAP*	0.068	**0.069**	0.064	0.060	0.057	0.055	0.054	0.052	0.049	0.048
	#r.r.d	2369	**2389**	2316	2269	2200	2080	2060	1998	1774	1754
	p@10	**0.240**	0.230	0.216	0.206	0.186	0.182	0.178	0.162	0.150	0.146
						Large Summary					
MB	*MAP*	0.040	0.044	**0.049**	0.049	0.049	0.049	0.049	0.048	0.041	0.037
	#r.r.d	1458	1606	**1650**	1625	1585	1581	1557	1479	1183	1119
	p@10	0.164	0.152	**0.168**	0.156	0.154	0.152	0.154	0.142	0.134	0.114
MBWNQT	*MAP*	**0.076**	0.076	0.073	0.069	0.068	0.065	0.064	0.061	0.057	0.056
	#r.r.d	2547	**2589**	2586	2545	2526	2491	2432	2233	2037	1966
	p@10	**0.270**	0.262	0.246	0.228	0.218	0.210	0.206	0.196	0.180	0.176
						Full Document					
MB	*MAP*	0.030	0.035	0.049	0.055	0.058	0.060	0.062	**0.066**	0.066	0.058
	#r.r.d	1277	1389	1646	1727	1765	1793	1840	**1889**	1820	1779
	p@10	0.126	0.128	0.160	0.172	0.170	**0.172**	0.172	0.166	0.162	0.150
MBWNQT	*MAP*	0.077	0.079	**0.081**	0.081	0.081	0.080	0.079	0.076	0.072	0.070
	#r.r.d	2717	2801	**2851**	2845	2811	2773	2707	2666	2560	2512
	p@10	0.192	**0.196**	0.180	0.174	0.174	0.174	0.166	0.164	0.150	0.152

Table XV. Summary retrieval on AQUAINT (TREC HARD Robust 2005): MB vs MBWNQT with long queries

and performs poorly when text length is increased. These results confirm our initial intuitions about the MB model: it is less competitive as the size of the retrieval units increases. The MN model's performance steadily improves as the length of the summaries increases, in contrast to the MB model which hardly obtains any benefit from the larger summaries. The MN model makes a good use of the additional data to produce better statistical estimates based on the counts of terms while the MB model, based on a binary notion of term presence in texts, yields to very modest improvements of performance. The MBB model behaves similarly to the MN model with short queries, suggesting its performance is also improved used large summaries that improve the quality of the estimation.

However, the poor behavior of the MB(B) models with long queries deserves special attention. There is a significant degradation in performance that cannot be attributed here to any length effect (all summaries have the same size). We therefore posited whether the non-query term component in the MB retrieval functions was the cause of the problem. To further analyze this issue, we ran another series of experiments with the MBWNQT model. As explained in section 4.1.1, this model is based on the MB likelihood but without the non-query terms component (eqs. 45 and 46). We can study the effect of the non-query term component by comparing the MB and the MBWNQT models. This comparison is reported in Tables XIV (short queries) and XV (long queries). With short queries the models are comparable: MBWNQT looks more solid in terms of MAP but MB outperforms MBWNQT in terms of p@10 with medium to large summaries. In contrast, MBWNQT is clearly superior to MB when queries are long. This gives clear evidence to support the hypothesis that the non-query term component is harming with long queries. It is natural to believe that long queries will probably contain most of the relevant terms, so that non-query component will be introducing noise because of the unrelated high idf terms scored in this component. That is, a regular matching between query and document terms is good enough to score long queries and there is no need to check on the additional (non-query) document terms. On the contrary, short queries might be meagre and, therefore, a more elaborated matching function might be of help.

Recall that the MB(B) models also performed poorly with long queries in the document retrieval experiments reported in section 4.2. Those document retrieval results, together with the present summary experiments, provide strong evidence to claim that, given long queries: 1) the MB(B) models tend to retrieve longer documents, and 2) even when the lengths of the retrieval units are uniform, the non-query term component is detrimental to retrieval performance.

5. DISCUSSION

So far we have seen that the multi-variate Bernoulli models have shown to be very effective for sentence retrieval. However in document and summary retrieval, one of the main conclusions of this study is that MB(B) models performance degrades as both the size of the retrieved unit is increased and as queries become longer. This was in contrast to the Multinomial model which performed well regardless of the retrieval scenario. The differences in performance stemmed from the different theoretical derivations, which motivated further analysis of the behavior of these models. In this respect, the first point we paid attention to was how the MB(B) models behave with respect to document length. As argued throughout this paper our belief is that these models tend to promote the retrieval of long units of texts. This effect is more pronounced when queries are long. The general

tendency of MB(B) models to retrieve long documents is easily explained after analyzing the retrieval formulas in Equations 35 and 43. Both models incorporate components that give higher scores to long texts.

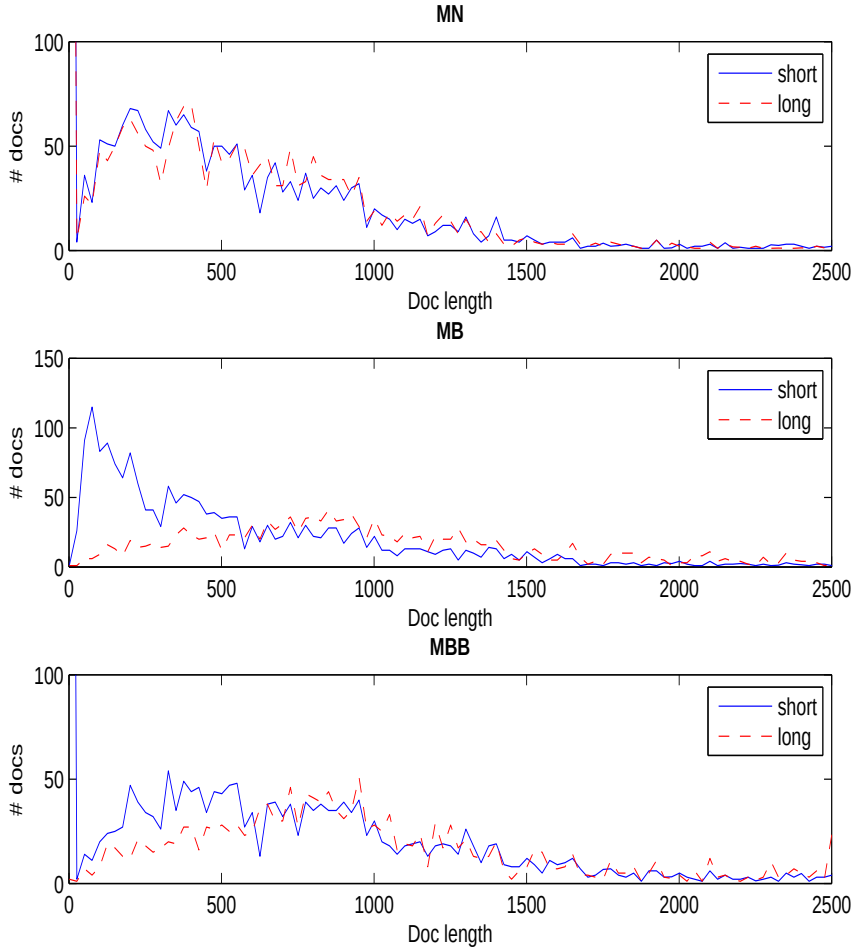

Fig. 1. Document retrieval on TREC8: Document length analysis. The plots show the distribution of lengths in the retrieved set of documents given short and long queries.

To further analyze this behavior, we computed the distribution of the length of documents returned in response to the different length queries. Since summaries are of equal length, we center our analysis on the full document retrieval runs performed on the TREC8 and AQUAINT collections. For each model, we took the set of documents retrieved by the

best run and, for each unique document length, we counted the number of documents retrieved of that length[17]. Table XVI summarizes the mean and median length of the retrieved documents by each approach and Figures 1 and 2 display the distribution of the number of retrieved documents given length (for clarity only the interpolated distribution is shown).

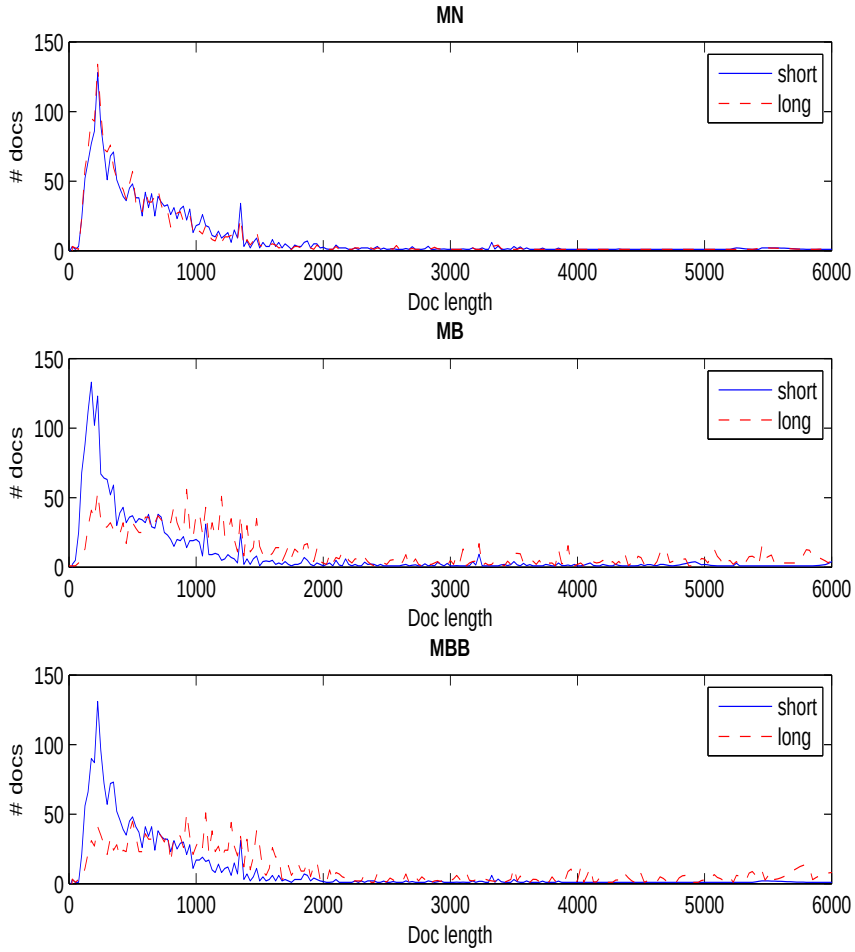

Fig. 2. Document retrieval on AQUAINT: Document length analysis. The plots show the distribution of lengths in the retrieved set of documents given short and long queries.

Clearly the MN model exhibits a very similar distribution for short and long queries. However, the MB models show a clear tendency to favor the retrieval of long documents

[17] These statistics were computed globally (i.e. if a document is retrieved by two different queries then it is counted twice).

	TREC8		AQUAINT	
	Short qs	Long qs	Short qs	Long qs
MN	713 (476)	707 (554)	638 (508)	566 (438)
MB	996 (416)	3804 (1101)	575 (377)	1129 (917)
MBB	1501 (743)	2519 (1014)	609 (479)	1060 (915)

Table XVI. Document retrieval on TREC8 and AQUAINT: Average (and median) length of retrieved documents

for long queries, as the distribution given these queries tends to be shifted to the right. For these models, large queries retrieve lower numbers of short documents. This happens in both collections and on both multi-variate Bernoulli models. Although the figures are indicative of the retrieval trends, we applied a statistical test to confirm that the lengths of the retrieved documents come from distributions with different medians. The Mann-Whitney U test, performed at the 5% significance level, rejected the hypothesis that the distributions of lengths have equal medians. This occurred in all comparisons and collections (MN with short queries versus MN with long queries, MB with short queries vs MB with long queries and MBB with short queries vs MBB with long queries). The long queries retrieved shorter documents than the short queries for the MN; whereas for the MB(B) models the long queries retrieved longer documents than the short queries. However, while all are significantly different, it is clear from the figure that the difference between the distribution of lengths for long and short queries was far more pronounced for the MB(B) models.

This analysis provided evidence confirming the above hypothesis: the MB(B) models do tend to retrieve longer documents when queries are longer. However, this retrieval behavior does not seem to benefit the retrieval performance given the results reported in the previous sections. Short queries (mainly consisting of keywords) on the MB(B) perform relatively well, but verbose queries boost the retrieval of long pieces of text. Longer documents, where most of these are considered non-relevant, are ranked higher, which has a detrimental effect on retrieval performance. These plots show empirically that the MN retrieval function (eq. 25) retrieves a similar distribution of documents regardless of the type of query (note that the second addend in eq. 25 is dependent on the query length: the penalty for long documents grows as queries are longer). On the contrary, the MB models' retrieval formula (eq. 35) does not incorporate any query-dependent document length retrieval correction and, therefore, the retrieval of long texts is favored with long queries. The MBB model, whose retrieval function was shown in eq. 42, contains some query-dependent corrections and, actually, the trends reported in table XVI confirm that it retrieves fewer long documents than the MB model, for longer queries. Regardless, the MN model is much more stable than the MBB model w.r.t the length of the query.

To further analyze the behavior of the MB model against document length, we conducted additional experiments applying the MB model with the non-query component removed (MBWNQT, section 4.1.1). In this way, we can check whether the boosting of the retrieval of long documents with long queries should be attributed to the non-query term component or, on the contrary, the sum across matching terms is the main reason behind the promotion of long texts. The retrieval function of the MBWNQT model was presented in eq. 46. We ran document retrieval experiments similar to those reported along this section and extracted the subsequent document length retrieval trends. We observed that the behavior of the MBWNQT model was similar to the behavior of the MB model with long queries (e.g. mean length of retrieved documents: 3804 for the MB model and 3870 for the MBWNQT in TREC-8; 1129 for the MB model and 1140 for the MBWNQT in AQUAINT).

In contrast, the models showed a distinctively different trend with short queries. More specifically, the MB model without the non-query component retrieved significantly longer documents with short queries (e.g. mean length of retrieved documents: 996 for the MB model and 1654 for the MBWNQT in TREC-8; 575 for the MB model and 848 for the MBWNQT in AQUAINT). Given these results, we can conclude that the promotion of long documents given long queries cannot be attributed to the non-query terms component (the model without this component retrieves a similar amount of long documents for this type of query). It is also interesting to observe that the MB and MBWNQT models were roughly equivalent with long queries but showed distinct tendencies with short queries. If we compare their retrieval functions (MB model: eq. 35; MBWNQT: eq. 46) we can find a natural explanation. Both formulas are based on the same sum across matching terms ($\sum_{w_i \in Q \cap D} \log \frac{\alpha_i}{\alpha_i - 1}$) but the MB model includes an additional sum across document terms that are not in the query ($\sum_{w_i \in D, w_i \notin Q} \log \frac{\beta_i - 1}{\beta_i}$). As queries become longer, the importance of this sum in the final retrieval score diminishes and, therefore, the MB model will tend to produce ranks which are similar to the ones generated by MBWNQT. With short queries, the importance of this factor is higher (and, conversely, the importance of the matching terms sum is lower). Note also that a matching term weight is $\log \frac{\alpha_i}{\alpha_i - 1}$ while a document non-matching term weight is $\log \frac{\beta_i - 1}{\beta_i}$. Since $\log \frac{\alpha_i}{\alpha_i - 1} > \log \frac{\beta_i - 1}{\beta_i}$, a shorter document will tend to be favored more with the MB model than with the MBWNQT model (the average term weight is lower with the MB model and, hence, a short document has more chances to get a retrieval score that is higher than the score assigned to a long document). Although both models have a clear tendency to promote long texts, the tendency is less pronounced in the MB model.

These findings also explain the good performance of the MB approach for sentence retrieval. The MB(B) models lead to a natural selection of sentences that not only cover some query words but also go beyond by mentioning some other significant terms. This feature, which is particularly useful in sentence retrieval (short sentences are less likely to be relevant), is not present in the MN model, whose retrieval function accounts only for query-document matching terms (plus a length correction factor).

5.1 Assessments

Given the specific characteristics of the MB(B) models with respect to the MN model, in terms of length and performance, it is reasonable to suspect that distinctly different set of documents are being returned. More generally, most retrieval models are strongly focused on the overlap between query and document, whereas the non-query terms are usually ignored (a novel model feature). Since we have evaluated these models under the TREC evaluation conditions, whose relevance assessments are incomplete due to pooling [Voorhees and Harman 2005]. It might be the case that the number of assessed documents retrieved by the MB models is substantially lower than the number of assessed documents retrieved by the MN model due to system omission [Zobel 1998], which may mean the system performance of the MB models is underestimated.

To investigate whether this may be the case, we considered the percentage of assessed documents returned for each model [Baillie et al. 2007]. This only makes sense for retrieval of documents and summaries because these collections all use pooling in the assessment procedure, and to the best of our knowledge no MB(B) models contributed to these pools. Whereas for sentence retrieval in the TREC novelty tracks pooling was not used. Instead,

	MN	MB	MBB
Short qs	23.05	15.15	22.56
Long qs	22.78	11.62	17.05

Table XVII. Document retrieval on TREC8: The average percentage of retrieved documents assessed in the top 1000 retrieved.

	Summary	MN	MB	MBB
Short qs	Small	8.7	7.5	8.3
	Medium	15.6	10.3	14.9
	Large	19.9	11.3	19.4
	Full	38.7	16.8	38.5
Long qs	Small	8.4	5.2	7.1
	Medium	15.1	7.9	10.4
	Large	20.1	9.7	15.9
	Full	38.9	11.4	30.4

Table XVIII. Retrieval of summaries on AQUAINT: The percentage of retrieved documents assessed, averaged over all queries on the top 1000 documents.

MB		MBB		MN	
Short qs	Long qs	Short qs	Long qs	Short qs	Long qs
.2048	.1683	.2584	.1730	.2611	.2659

Table XIX. Document retrieval on TREC8: bpref values

all sentences were assessed (and thus provide an "absolute" measurement of system performance). The levels of assessments for document retrieval and summary retrieval are shown in tables XVII and XVIII, respectively. The percentages are based on the assessment level at 1000 document averaged over all topics (where 1000 is the measurement depth used to compute MAP). The results show quite a difference in percentage of assessed document. In all cases, the MN model retrieves more assessed documents than the MBB model which in turn retrieves more assessed documents than the MB model (i.e MN > MBB > MB). For short queries the MBB retrieves only slightly less assessed documents over the MN, while the MB model retrieved somewhat less assessed documents than both the MN and MB models. For long queries, the differences in terms of assessed documents become more pronounced between the difference models. When changing from short to long queries, the influence of query length on assessed documents is dramatic for the MB and MBB, but not for the MN. For instance, on TREC 8, the relative change in assessment from short to long queries is 1.2%, 23.3% and 24.4% for the MN, MB and MBB models. If we consider the changes in assessment, w.r.t the trends in performance, there appears to be a correspondence between them; where lower levels of assessment correspond to lower system performance.

From these results one might be tempted to infer that MB models' performance is underestimated because of system omission bias[18][Zobel 1998]. To check this claim, we employed a different measure of system performance which accounts for incompleteness, called b-pref [Buckley and Voorhees 2004]. The b-pref measure was specially designed

[18]When a system is system has not contributed to the pool of judged documents then its performance may be under estimated [Zobel 1998].

	MB		MBB		MN	
Summary	Short qs	Long qs	Short qs	Long qs	Short qs	Long qs
Small	.134	.129	.134	.124	.129	.147
Medium	.155	.137	.160	.145	.163	.188
Large	.154	.148	.180	.164	.180	.204
Full	.163	.178	.240	.233	.239	.287

Table XX. Retrieval of summaries on AQUAINT: bpref values

to compare the effectiveness of the systems on the basis of judged documents only. Other measures, such as MAP or P@10, make no distinction in pooled collections between documents that are explicitly judged as non-relevant and documents that are assumed to be non-relevant because they are unjudged. In contrast, b-pref computes the number of times judged non-relevant documents are retrieved before relevant documents. The b-pref scores for document and summary retrieval are shown in Tables XIX and XX. Despite, accounting for the incompleteness, the b-pref scores still indicate that the MN model is significantly better than or at worst comparable to the MB(B) models (as the significance tests resulted in the same conclusions as for MAP). Although the MB(B) models retrieved many un-assessed documents, the b-pref values indicate that these models brought much more judged non-relevant documents than relevant documents. Clearly, the novel characteristics of the MB(B) models are detrimental for document retrieval but they appear beneficial for fine granularity retrieval tasks, such as sentence retrieval.

6. CONCLUSIONS

In this paper, we have analyzed the multi-variate Bernoulli models against the popular Multinomial model specifically examining the differences between the models and how these affect the retrieval performance both in terms of efficiency and effectiveness. First of all, we contextualized the Multinomial and multi-variate Bernoulli models under the framework of Bayesian analysis, which helped to analyze these LM approaches in a principled way. Second, we derived the retrieval functions associated to the multi-variate Bernoulli models. This derivation showed how the MB models handle important issues such as inverse document frequency and document length and how they related to smoothing. This derivation also demonstrated that the MB models can be implemented efficiently. Third, we performed a number of experiments to understand how these LM approaches behave under different retrieval scenarios. The main conclusions derived from this work can be enumerated as follows:

—Multi-variate Bernoulli models can support current retrieval systems because optimized algorithms can be designed to implement the retrieval function of the multi-variate Bernoulli model. In the MB model, the resulting retrieval method is as efficient as popular IR techniques (e.g. vector space based on tf/idf).

—The task of sentence retrieval is especially suitable for the MB(B) models. The ability to account for the quality of non-query sentence terms appears as a useful feature and, indeed, the MB(B) models perform clearly better than MN models. Sentences are usually centered on a single theme and, thus, the distinctive component supplied by the multi-variate Bernoulli models works properly (it is unlikely that we promote sentences that deviate from query topics).

—Despite its simplicity, the basic MB model, whose underlying binary representations

are equivalent to the ones handled by the classic Binary Independence Model, retrieves relevant sentences at least as effectively as the MBB model (and, in some cases, significantly better). It seems therefore that the use of non-binary term frequency information is not beneficial for improving the performance in this sentence retrieval task.

—Multi-variate Bernoulli models come up against difficulties when retrieving larger pieces of text, whereas the MN model is always at least as good as the MB(B) models. In particular, the general tendency of the MB models to retrieve long texts leads to poor retrieval performance. This behavior is reasonable for sentence retrieval (searching for sentences that go beyond the query topics) but it is not desirable in document retrieval.

—The MBB model outperforms the MB model for document retrieval and for the retrieval of summaries, where the MB model becomes less competitive as the size of the summaries increases. The MB(B) models were inferior to the MN model on these tasks.

—The multi-variate Bernoulli models do not work well with long queries. With this sort of queries, which mention explicitly most of the relevant terms, there is no need to promote documents/summaries with high idf non-query terms. Actually, this promotion is detrimental to performance because these terms are likely to be unrelated to the query and thus introduce noise.

—In document retrieval, a very distinctive retrieval trend is produced by the multi-variate Bernoulli models. These models tend to retrieve substantially more long documents (especially when queries are long). This resulted in significantly poorer performance. In the future, it would be interesting to explore this issue further analyzing whether any adjustment in the MB models is possible to overcome this limitation.

—Despite a disparity in assessment between the MN and MB(B) models the b-pref scores confirmed the results obtained by MAP, and showed that the MN model is superior to the MB(B) models for document and summary retrieval.

—The points above reflect the peculiarities of multi-variate Bernoulli models, which retrieve pieces of text using a *dual* method (i.e. how much of the query is covered by the text vs how good is the text as a whole). These characteristics are not standard in standard retrieval models (like MN).

In conclusion, multi-variate Bernoulli models look especially suitable for fine granularity retrieval tasks where typical queries are short. Although we analyzed here a sentence retrieval task, we believe that other IR problems, such as element retrieval, could be better handled by the MB(B) models. For other tasks, however, the MN is more reliable, performing well under all circumstances regardless of document length and query length, whilst the MB model was not.

Acknowledgements

The authors would like to thank Nebojsa Gvozdenovic and Stephen Roberston for their useful comments and suggestions regarding the efficiency of the Multi-variate Bernoulli Models. Thanks also to Donald Metzler and Hugo Zaragoza for clarifying several issues regarding the MBB model and MN models, respectively. Finally, we would also like to thank the anonymous reviewers of this paper for their very constructive feedback. 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

AMATI, G. 2006. Frequentist and bayesian approach to information retrieval. In *Proc. of the 28th European Conference on Information Retrieval Research, ECIR-06*. 13–24.

AMATI, G. AND 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. AND LOSADA, D. E. 2006. An efficient computation of the multiple-bernoulli language model. In *Proc. of the 28th European Conference on Information Retrieval Research, ECIR-06*. 480–483.

AZZOPARDI, L. AND ROELLEKE, T. 2007. An alternative formulations of relevance within the language modeling framework. In *Proceedings of the International Conference in Theory of Information Retrieval*. Alma Mater, 125–134.

BAILLIE, M., AZZOPARDI, L., AND RUTHVEN, I. 2007. A retrieval evaluation methodology for incomplete relevance assessments. In *To appear in Proc. 29th European Conference on Information Retrieval Research, ECIR'2007*. Springer Verlag, LNCS, Rome, Italy.

BERNARDO, J. AND SMITH, A. 1994. *Bayesian Theory*. Wiley.

BUCKLEY, C. AND VOORHEES, E. 2004. Retrieval evaluation with incomplete information. In *Proc. SIGIR-04, the 27th ACM Conference on Research and Development in Information Retrieval*. Sheffield, UK, 25–32.

CALLAN, J. AND CONNELL, M. 2001. Query-based sampling of text databases. *ACM Trans. Inf. Syst. 19*, 2, 97–130.

CROFT, W., Ed. 2000. *Advances in information retrieval. Recent research from the center of intelligent information retrieval*. Kluwe academic publishers.

CROFT, W. B. AND LAFFERTY, J. 2003. *Language Modeling for Information Retrieval*. Kluwer Academic.

DEGROOT, M. 1988. *Probability and Statistics*. Addison-Wesley.

HARMAN, D. 2002. Overview of the trec 2002 novelty track. In *Proc. TREC-2002, the 11th text retrieval conference*.

HAUFF, C. AND AZZOPARDI, L. 2005. Age dependent document priors in link structure analysis. In *Proc. 27th European Conference on Information Retrieval Research, ECIR'2005*, D. Losada and J. M. Fernandez-Luna, Eds. Springer Verlag, LNCS 3408, Santiago de Compostela, Spain, 552–554.

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

HIEMSTRA, D. 2001. Using language models for information retrieval. Ph.D. thesis, University of Twente, Enschede.

KAMPS, J. 2005. Web-centric language models. In *Proc. ACM Conference on Information and Knowledge Management (CIKM)*.

KRAAIJ, W. 2004. Variations on language modeling for information retrieval. Ph.D. thesis, University of Twente.

KRAAIJ, W., WESTERVELD, T., AND HIEMSTRA, D. 2002. The importance of prior probabilities for entry page search. In *Proc. 25th ACM Conference on Research and Development in Information Retrieval, SIGIR'02*. Tampere, Finland, 27–34.

LAPLACE, P. 1995. *Philosophical essay on probabilities*. Sources in the History of Mathematics and Physical Sciences. Springer Verlag.

LARKEY, L., ALLAN, J., CONNELL, M., BOLIVAR, A., AND WADE, C. 2002. Umass at trec 2002:cross language and novelty tracks. In *Proc. TREC-2002, the 11th text retrieval conference*.

LAVRENKO, V. 2004. A generative theory of relevance. Ph.D. thesis, University of Massachusetts Amherst.

lemur. The lemur toolkit. http://www.lemurproject.org/.

LIDSTONE, G. J. 1920. Note on the general case of the bayes-laplace formula for inductive a priori probabilities. *Transactions of the Faculty of Actuaries 8*, 182–192.

LOSADA, D. 2005. Language modeling for sentence retrieval: A comparison between multiple-bernoulli models and multinomial models. In *Information Retrieval and Theory Workshop*. Glasgow, UK.

LOSADA, D. AND AZZOPARDI, L. 2008. An analysis on document length retrieval trends in language modeling smoothing. *Information Retrieval (in press)*.

MANNING, C. D. AND SCHUTZE, H. 2000. *Foundations of Statistical Language Processing*. MIT Press, Cambirdge, Massachusetts.

MCCALLUM, A. AND NIGAM, K. 1998. A comparison of event models for naive bayes text classification. In *Proc. AAAI/ICML-98 Workshop on Learning for Text Categorization*. AAAI press, 41–48.

METZLER, D., LAVRENKO, V., AND CROFT, W. B. 2004. Formal multiple-bernoulli models for language modeling. In *Proc. 27th ACM Conference on Research and Development in Information Retrieval, SIGIR'04*. ACM press, Sheffield, UK, 540–541.

MILLER, D., LEEK, T., AND SCHWARTZ, R. 1999. A hidden markov movel information retrieval system. In *Proc. of SIGIR-99, the 22nd ACM Conference on Research and Development in Information Retrieval*. Berkeley, 214–221.

MURDOCK, V. 2006. Aspects of sentence retrieval. Ph.D. thesis, University of Massachusetts.

PONTE, J. AND CROFT, W. B. 1998. A language modeling approach to information retrieval. In *Proc. 21st ACM Conference on Research and Development in Information Retrieval, SIGIR'98*. Melbourne, Australia, 275–281.

PONTE, J. M. 1998. A language modeling approach to information retrieval. Ph.D. thesis, University of Massachusetts Amherst.

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

RABINER, L. R. 1989. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of IEEE*. Vol. 77. 257–286.

ROBERTSON, S. AND WALKER, S. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proc. SIGIR-94, the 17th ACM Conference on Research and Development in Information Retrieval*. Dublin, Ireland, 232–241.

ROSENFELD, R. 2000. Two decades of statistical language modeling: Where do we go from here? *Proceedings of the IEEE 88*, 8.

SALTON, G., WONG, A., AND YANG, C. 1975. A vector space model for automatic indexing. *Communications of the ACM 18*, 613–620.

SOBOROFF, I. 2004. Overview of the trec 2004 novelty track. In *Proc. TREC-2004, the 13th text retrieval conference*.

SOBOROFF, I. AND HARMAN, D. 2003. Overview of the trec 2003 novelty track. In *Proc. TREC-2003, the 12th text retrieval conference*.

STOKES, N., NEWMAN, E., CARTHY, J., AND SMEATON, A. F. 2004. Age dependent document priors in link structure analysis. In *Proc. 26th European Conference on Information Retrieval Research, ECIR'2004*, S. McDonald and J. Tait, Eds. Springer Verlag, LNCS 2997, Sunderland, UK, 209–222.

VOORHEES, E. AND HARMAN, D. 1999. Overview of the eight text retrieval conference. In *Proc. TREC-8, the 8th text retrieval conference*.

VOORHEES, E. M. AND HARMAN, D. K., Eds. 2005. *TREC: Experiment and Evaluation in Information Retrieval*. MIT Press, Cambridge, Massachusetts 02142.

ZARAGOZA, H., HIEMSTRA, D., AND TIPPING, M. 2003. Bayesian extension to the language model for ad hoc information retrieval. In *Proc. 26th ACM Conference on Research and Development in Information Retrieval, SIGIR'03*. Toronto, Canada, 4–9.

ZHAI, C. AND LAFFERTY, J. 2001a. Model-based feedback in the language modeling approach to information retrieval. In *Proc. of the 10th International Conference on Information and Knowledge Management, CIKM-2001*. ACM press, Atlanta, USA, 403–410.

ZHAI, C. AND LAFFERTY, J. 2001b. A study of smoothing methods for language models applied to adhoc information retrieval. In *Proc. 24th ACM Conference on Research and Development in Information Retrieval, SIGIR'01*. New Orleans, USA, 334–342.

ZHAI, C. AND LAFFERTY, J. 2002. Two-stage language models for information retrieval. In *Proc. 25th ACM Conference on Research and Development in Information Retrieval, SIGIR'02*. Tampere, Finland, 49–56.

ZHAI, C. AND LAFFERTY, J. 2004. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems 22*, 2, 179–214.

ZHAI, C. X. 2002. Risk minimization and language modeling in text retrieval. Ph.D. thesis, Carnegie Mellon University.

ZOBEL, J. 1998. How reliable are the results of large-scale information retrieval experiments? In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM Press, 307–314.

Appendix A

In equation 17, we set $\alpha_i = \mu P(w_i|C) + 1$ in order to smooth the estimations using the probabilities in the fallback model.

Since the prior distribution (eq. 16) follows a multiple-Beta distribution, the expectations are equal to $\frac{\alpha_i}{\alpha_i+\beta_i}$. It is quite natural to force that the expectations are equal to $p(w_i|C)$.

$$\begin{aligned}
E[P(\theta)] &= P(w_i|C) = \frac{\alpha_i}{\alpha_i + \beta_i} \\
P(w_i|C) &= \frac{\mu P(w_i|C) + 1}{\mu P(w_i|C) + 1 + \beta_i} \\
P(w_i|C) \cdot (\mu P(w_i|C) + 1 + \beta_i) &= \mu P(w_i|C) + 1 \\
\mu P(w_i|C)^2 + P(w_i|C) + \beta_i P(w_i|C) &= \mu P(w_i|C) + 1 \\
\beta_i P(w_i|C) &= \mu P(w_i|C) + 1 - \mu P(w_i|C)^2 - P(w_i|C) \\
\beta_i P(w_i|C) &= 1 + P(w_i|C) \cdot (\mu - \mu P(w_i|C) - 1) \\
\beta_i &= \frac{1 + P(w_i|C) \cdot (\mu - \mu P(w_i|C) - 1)}{P(w_i|C)} \\
\beta_i &= \frac{1}{P(w_i|C)} + \mu(1 - P(w_i|C)) - 1
\end{aligned}$$

Appendix B

We describe the computation complexity according to the number of term score calculations required. For the MB algorithm, the loop 1-7 involves $|COL| \cdot |V| \cdot s$ iterations, where s is the sparsity expressed as the percentage of non-zero entries in the document-term matrix. This is computed offline and so does not directly affect on-line performance. At query time, the online computations take $|COL| \cdot |Q| \cdot s$ (loop 8-17) iterations, where $|Q|$ is the number of query terms. Under this derivation a very significant reduction in the run time of the MB retrieval model can be achieved which makes it comparable to other state of the art retrieval models.

Appendix C

The MBB algorithm is more complicated than the MB algorithm. Before query time, we have $dls \cdot |V|$ iterations (loop 1-8) and $|COL| \cdot |V| \cdot s$ (loop 9-15) iterations, where dls is the number of unique document lengths in the collection and s is the sparsity expressed as the percentage of non-zero entries in the document-term matrix. This is affordable because it does not affect query time performance. At query time we have the usual inverted file search (loop 16-22), which needs $|COL| \cdot |Q| \cdot s$ iterations, but it also requires to compute the sum across query terms ($\sum_{w_i \in Q} tf(i,Q) \cdot \log \frac{\alpha_i - 1}{n_D + \beta_i - 1}$) for each unique length in the

Algorithm 1 MB algorithm (offline:steps 1-7, online:steps 8-16)

1: **for all** D $\in$ COL **do**
2: DOCDEPSUM[D] = 0
3: **for all** $w_i \in$ D **do**
4: $\beta_i = \frac{1}{P(w_i|C)} + \mu(1 - P(w_i|C)) - 1$
5: DOCDEPSUM[D] = DOCDEPSUM[D] + $\log \frac{\beta_i - 1}{\beta_i}$
6: **end for**
7: **end for**
8: **for all** qt $\in$ Q **do**
9: **for all** D in the posting lists of qt (inverted file access) **do**
10: **if** RSV[D] is unassigned **then**
11: RSV[D] = DOCDEPSUM[D]
12: **end if**
13: $\alpha_i = \mu P(qt|C) + 1$
14: $\beta_i = \frac{1}{P(w_i|C)} + \mu(1 - P(w_i|C)) - 1$
15: RSV[D] = RSV[D] + $\log \frac{\alpha_i}{\alpha_i - 1} \cdot \frac{\beta_i}{\beta_i - 1}$
16: **end for**
17: **end for**

Algorithm 2 MBB algorithm (offline:steps 1-15, online:steps 16-33)

1: **for all** unique doc length l in the collection **do**
2: SUMV[l] = 0
3: **for all** $w_i \in$ V **do**
4: $\alpha_i = \mu P(w_i|C) + 1$
5: $\beta_i = \frac{1}{P(w_i|C)} + \mu(1 - P(w_i|C)) - 1$
6: SUMV[l] = SUMV[l] + $\log(1 - \frac{\alpha_i - 1}{l + \alpha_i + \beta_i - 2})$
7: **end for**
8: **end for**
9: **for all** D $\in$ COL **do**
10: SUMD[D] = 0
11: **for all** $w_i \in$ D **do**
12: $\beta_i = \frac{1}{P(w_i|C)} + \mu(1 - P(w_i|C)) - 1$
13: SUMD[D] = SUMD[D] + $\log(1 - \frac{tf(i,D)}{length(D) + \beta_i - 1})$
14: **end for**
15: **end for**
16: **for all** qt $\in$ Q **do**
17: **for all** D in the posting lists of qt (inverted file access) **do**
18: $\alpha_i = \mu P(qt|C) + 1$
19: $\beta_i = \frac{1}{P(qt|C)} + \mu(1 - P(qt|C)) - 1$
20: RSV[D] = RSV[D] + $tf(qt, Q) \log \frac{(tf(qt,D) + \alpha_i - 1) \cdot (length(D) + \beta_i - 1)}{(\alpha_i - 1) \cdot (length(D) + \beta_i - 1 - tf(qt,D))}$
21: **end for**
22: **end for**
23: **for all** D whose RSV[D] is assigned **do**
24: **if** SUMQ[length(D)] is unassigned **then**
25: SUMQ[length(D)]=0
26: **for all** qt $\in$ Q **do**
27: $\alpha_i = \mu P(qt|C) + 1$
28: $\beta_i = \frac{1}{P(qt|C)} + \mu(1 - P(qt|C)) - 1$
29: SUMQ[length(D)]= SUMQ[length(D)] + $tf(qt, Q) \log \frac{\alpha_i - 1}{length(D) + \beta_i - 1)}$
30: **end for**
31: **end if**
32: RSV[D]= RSV[D] + SUMQ[length(D)] + length(Q) * SUMD[D] + length(Q) * SUMV[length(D)]
33: **end for**

scored documents (loop 23-33). In the worst case, this requires to traverse every unique document length in the collection taking $|Q| \cdot dls$ iterations. Still, the MBB algorithm proposed here is significantly better than a direct computation of the MBB likelihood, which would involve $|V| \cdot |COL|$ steps at query time. Although the loop 23-33 in the MBB algorithm is an important penalty, further optimizations can be designed to improve the query response times (e.g. organize the documents into chunks of similar size and compute approximate values for the probability scores, instead of exact values).