A Pólya Urn Document Language Model for Improved Information Retrieval

Ronan Cummins, University of Cambridge
Jiaul H. Paik, University of Maryland, College Park
Yuanhua Lv, Microsoft Research

The multinomial language model has been one of the most effective models of retrieval for over a decade. However, the multinomial distribution does not model one important linguistic phenomenon relating to term-dependency, that is the tendency of a term to repeat itself within a document (i.e. word burstiness). In this article, we model document generation as a random process with reinforcement (a multivariate Pólya process) and develop a Dirichlet compound multinomial language model that captures word burstiness directly.

We show that the new reinforced language model can be computed as efficiently as current retrieval models, and with experiments on an extensive set of TREC collections, we show that it significantly outperforms the state-of-the-art language model for a number of standard effectiveness metrics. Experiments also show that the tuning parameter in the proposed model is more robust than in the multinomial language model. Furthermore, we develop a constraint for the verbosity hypothesis and show that the proposed model adheres to the constraint. Finally, we show that the new language model essentially introduces a measure closely related to idf which gives theoretical justification for combining the term and document event spaces in tf-idf type schemes.

1. INTRODUCTION

Language modelling approaches to information retrieval have become increasingly popular since the original works [Ponte and Croft 1998; Hiemstra 1998, 2001; Lavrenko and Croft 2001; Zhai and Lafferty 2001a]. They afford a particularly appealing view of the retrieval problem due in part to the principled nature in which a retrieval function can be mathematically derived. The query likelihood method [Ponte and Croft 1998] is one of the most widely-adopted approaches to retrieval, and ranks documents based on the likelihood of their document language model generating the query string. The most widely-accepted multinomial language model treats the doc-

ument model as a multinomial distribution over the terms, where the parameters of each document model are estimated using the observations from the actual document smoothed with the entire collection using the Dirichlet prior smoothing method [Zhai and Lafferty 2001a].

One main deficiency with using a multinomial distribution as a language model is that all term occurrences are treated independently. The term-independence assumption in information retrieval is often adopted in theory and practice as it renders the retrieval problem tractable, simplifies the implementation of many models, and has been shown to be suitably effective. Although retrieval approaches that incorporate term-dependencies [Metzler and Croft 2005; Zhao and Yun 2009; Lv and Zhai 2009a; Cummins and O'Riordan 2009; Bendersky and Croft 2012] have been shown in general to be more effective, they are computationally more complex. Therefore, a language modelling approach that has the same complexity as a unigram language model but also incorporates dependencies, would be a useful contribution as it would likely exhibit increased effectiveness at no extra computational cost. In fact, the use of the multinomial distribution in the standard language modelling approach ignores two types of dependencies; namely, the dependency between distinct terms[1] (word types) and the dependency between recurrences of the same term (word tokens). It is this second type of dependency that we address in this article.

It is well known that once a term occurs in a document, it is more likely to re-appear in the same document. This phenomenon is known as *word burstiness* [Church and Gale 1995; Madsen et al. 2005], and is a type of dependency that is not modelled in the multinomial language model [Zhai and Lafferty 2004]. Essentially, word burstiness can be defined as the tendency of an otherwise rare term to occur multiple times in a document, and can be seen as a form of *preferential attachment* [Simon 1955; Mitzenmacher 2003]. One theory for this phenomenon is that an author tends to sample terms previously written in the same document to form *association* [Simon 1955]. The process of association of similar concepts throughout a document using the same lexical form may aid coherence, readability, and understanding. For example, if an author starts to use the term **pavement** in an article, he/she intuitively tends to continue its usage throughout the document, rather than changing to one of its synonyms (e.g. **sidewalk** or **footpath**).

On the other hand, queries are requests for information and are generated with a different motive in mind. When requesting or searching for information a user is more likely to expand the vocabulary used in the query (and possibly make use of synonyms) in the hope of matching those query-terms contained in relevant documents. Furthermore, queries are usually much shorter than documents and as a result, we assume that queries are less likely to exhibit word-burstiness. That is not to say that a certain term could not appear multiple times in a query, it simply suggests that the reason for it reappearing is different than in a document. For these reasons we model documents and queries using different generative assumptions.

This article presents the **SPUD** (Smoothed Pólya Urn Document) language model that incorporates word burstiness only into the document model. We use the Dirichlet compound multinomial (DCM), also known as the multivariate Pólya distribution, to model documents in place of the standard multinomial distribution, while we use the standard multinomial to model query generation. We show that this new retrieval model obtains significantly increased effectiveness compared to the current state-of-the-art model on a range of datasets for a number of effectiveness metrics. This article is organized as follows. Section 2 introduces notation used in the remainder of the article and also presents a comprehensive review of relevant research. Section 3 reviews

---

[1]This is the traditional term-independence assumption.

the standard language modelling approach. Section 4 presents the SPUD language model. Section 5 outlines efficient forms of the new retrieval functions, and provides deep insights into the proposed functions. The experimental design and results are presented in Section 6. Section 7 presents a discussion of the results and Section 8 concludes with a summary.

2. RELATED RESEARCH

In this section we review related work in language models and word burstiness, before outlining the main contributions of this work. Table I introduces notation used in the remainder of this article.

Table I. Feature Notation

Key	Description
$c(t, d)$	frequency of term t in document d
$c(t, q)$	frequency of term t in query q
$|d|$	length of document d (i.e. number of word tokens)
$|\vec{d}|$	length of document vector (# of distinct terms in document d)
cf_t	collection frequency (frequency of t in the entire collection)
df_t	document frequency (number of documents in which t occurs)
$|q|$	length of query q (i.e. number of word tokens)
$|c|$	number of tokens in the entire collection c
n	number of documents in the collection
$|v|$	vocabulary of the collection (# of distinct terms in the collection)

2.1. Query Likelihood

The predominant method of ranking documents using the language modelling approach remains the query likelihood method of Ponte and Croft [1998]. In the query likelihood method, documents are ranked based on the likelihood of their document model, $\mathcal{M}_d$, generating the query string. The following equation shows how the query likelihood, $p(q|\mathcal{M}_d)$, is calculated for a unigram multinomial language model:

$$p(q|\mathcal{M}_d = \boldsymbol{\theta}_{dm}) = \prod_{t \in q} p(t|\boldsymbol{\theta}_{dm})^{c(t,q)} \qquad (1)$$

where q is the query string and $\boldsymbol{\theta}_{dm}$ is the multinomial document language model. The effectiveness of this retrieval method crucially depends on the estimation of the document model $\boldsymbol{\theta}_{dm}$. It is typically estimated using the actual document d and is smoothed with the background language model which is estimated from the entire collection c. When using a multinomial, the query likelihood method (Eq. 1) can be rewritten in a rank equivalent form as follows:

$$log\ p(q|\mathcal{M}_d = \boldsymbol{\theta}_{dm}) = \sum_{t \in q} (log\ p(t|\boldsymbol{\theta}_{dm}) \cdot c(t, q)) \qquad (2)$$

which shows that, as with most other retrieval functions (e.g. BM25 [Robertson et al. 1994]), the scoring function comprises a summation of query-term weights. If $p(t|\boldsymbol{\theta}_{dm})$ is estimated using only the maximum-likelihood estimates of a term occurring in a document (i.e. $c(t, d)/|d|$), over-fitting would occur. For instance, this would result in

any document that did not contain *all* query-terms not being retrieved, as its document model deemed to have generated the query with a probability of zero (see Eq. 1). It should also be noted that when substituting the maximum likelihood probabilities $(c(t,d)/|d|)$ into Eq. (2), the weight of each term becomes $log(c(t,d)/|d|)$ which has the effect of reducing the weight contribution of successive occurrences of the same term to a document score. This non-linear term-frequency effect has been often reported as a useful heuristic in IR [Fang et al. 2004; Fang and Zhai 2005; Cummins and O'Riordan 2007; Clinchant and Gaussier 2010, 2011; Lv and Zhai 2012]. However, in the multinomial query likelihood retrieval method this non-linearity is only the consequence of a mathematical transformation, and the actual dependency between successive occurrences of the same term is *not* modelled[2].

2.2. Advances in Language Models

Since the initial work applying language models [Ponte and Croft 1998; Hiemstra 1998] to information retrieval, there have been a number of advances in terms of both theory and practice. Graph-based models [Gao et al. 2004; Metzler and Croft 2005; Blanco and Lioma 2012; Bendersky and Croft 2012] that capture aspects of term-dependency have been shown to improve retrieval performance over unigram models. Furthermore, positional-based language models [Zhao and Yun 2009; Lv and Zhai 2010] have been proposed and incorporate term dependencies that often span several terms. In general, the incorporation of term-dependency information in larger web collections has been shown to be beneficial to retrieval quality.

Although many language modelling approaches to information retrieval use the query-likelihood approach to ranking, it is not the only means of inducing a ranking using language models. In particular, relevance-based language models [Lavrenko and Croft 2001] estimate a relevance model from which all relevant documents for a particular information need are assumed to have been drawn. The approach to ranking in that work is similar to the classic probabilistic document retrieval approaches [Spärck-Jones et al. 2000], where documents are ranked based on the odds of being drawn from the relevant class compared to non-relevant class. The relevance-based language modelling approach provides a principled mechanism in which the retrieval model can be updated as relevant and non-relevant documents become known. This approach led to the development of pseudo-relevance query expansion language models [Abdul-jaleel et al. 2004; Diaz and Metzler 2006; Lv and Zhai 2010].

The language modelling approach has now become a starting point from which more complex models can be built. Aside from pseudo-relevance query expansion, other approaches such as *latent Dirichlet allocation* (LDA) have been incorporated into ad-hoc retrieval [Wei and Croft 2006]. In essence, improving the retrieval effectiveness of the standard language modelling approach to information retrieval can ultimately benefit any of the myriad of approaches which depend upon it (e.g. pseudo-relevance feedback).

2.3. Word Burstiness

The modelling of word burstiness in documents has been addressed before in text-related tasks, but it has not been incorporated with the query likelihood method in information retrieval. Madsen et al. [2005] use the DCM distribution to model word burstiness and demonstrate its effectiveness on document classification. They estimate a DCM model for each class from training data. They then classify an unseen document to a specific class according to the mostly likely generative DCM class model. They

---

[2]This is an important point as some previous work tends to suggest that a non-linear term-frequency factor in a linear combination of term-weights can capture some aspect of dependency or *burstiness*. We contend that this is not the case for language models that use a multinomial as their basis.

show that this DCM model outperforms the more standard multinomial model. The information retrieval task is somewhat different as it deals with both documents and queries. In our work we have different generative assumptions for both documents and queries. A further difference is that in the classification task there are a number of documents from which we can infer a particular class model, while in the query-likelihood approach to information retrieval we have access to only one instance of a document from the document model.

Due to the complexity of estimating parameters for the DCM, Elkan [2006] developed an approximate distribution (the EDCM) and demonstrated its effectiveness for clustering. We make use of this approximation later in this article. The DCM has also been used in a hypergeometric language model [Tsagkias et al. 2011] for modelling the characteristics of very long queries. In other work, a two-stage language modelling approach has been developed [Goldwater et al. 2011] that generates words according to the power-law characteristics of natural language. They decompose the language generation process into a *generator*, which creates instances of word types, and an *adaptor* which has the tendency to repeat those specific word types. Further arguments which link preferential attachment to the power-law characteristics of natural language are reviewed by Mitzenmacher [2003]. Cowans [2004] uses a hierarchical Dirichlet process to arrive at a ranking function which is reported as being superior to BM25. Related work [Sunehag 2007] provides some interesting connections between the traditional tf-idf weighting scheme and the two-stage generator-adaptor models. Our work is more extensive and actually develops a document language model from which retrieval functions are derived.

In recent work, an extension of earlier information-based approaches [Amati and Van Rijsbergen 2002] is developed that incorporates burstiness in a log-logistic retrieval function [Clinchant and Gaussier 2009, 2011]. The authors develop a means for identifying if a term-frequency distribution is *bursty*. They conclude that the frequency distribution must be a type of power-law (or Pareto-type) distribution. Our work is much more in the spirit of generative language modelling where the term-frequency aspect occurs naturally from the model (in our case a hierarchical Bayesian approach) to introduce dependencies between subsequent occurrences of the same term. Our model also exhibits power-law characteristics consistent with the work by Clinchant and Gaussier [2011].

The work most similar to ours uses the DCM distribution to develop a probabilistic relevance-based language model [Xu and Akella 2008, 2010]. For each query they estimate a relevant and non-relevant DCM model and it is assumed that all documents are generated from either of those two models. However, our work does not assume a relevance model and instead, assumes that each document is generated from a different document model. This means that we model burstiness on a per document basis, rather than modelling burstiness for a set of relevant (and non-relevant) documents. It is more likely that different documents are bursty to different degrees as they were written by different authors, and this is not modelled in the relevance-based approach of Xu and Akella. Our model is a query-likelihood approach using different generative assumptions for both the document and query, and leads to retrieval functions that are distinct from those in the aforementioned relevance-based approach.

Although Xu and Akella [2008] report some improvement in retrieval effectiveness over the multinomial query likelihood retrieval method on some test collections, their experiments were restricted to relatively small collections (less than a million documents) and used only short keyword queries. It is unclear if their results extend to a more general retrieval scenario. We perform a more robust analysis by using their best approach (DCM-L-T) as one of our main baselines on a variety of different query lengths and collection sizes. We also discuss the difference between our approach and

the relevance-based DCM approach of Xu and Akella in our discussion section (Section 7.1).

2.4. Contributions

To our knowledge no existing work has developed a document language model for information retrieval using the generative assumptions outlined in this work. Therefore, the main contributions of this article are as follows:

- We propose a new family of document language models that capture word burstiness in a probabilistic manner.
- We develop closed-form expressions for the retrieval functions derived from the new language model, and show that our retrieval functions are as efficient as traditional bag-of-words retrieval functions.
- We show that the proposed language model implements several important retrieval heuristics not captured in the multinomial language model, such as modelling the *scope hypothesis* and the *verbosity hypothesis* separately.
- We show that the modelling of word burstiness in the new language model leads to significant improvement in retrieval effectiveness for ad hoc retrieval and for downstream methods such as pseudo-relevance feedback.

We now briefly review the query likelihood retrieval method and the multinomial language model.

3. MULTINOMIAL LANGUAGE MODEL

In this section we review details of the multinomial query likelihood model and some useful approaches to smoothing.

3.1. Document and Background Models

As outlined earlier, it is the selection of the generative model and the subsequent estimation of the document language model that is crucial to retrieval effectiveness using the query likelihood retrieval method. It has been shown [Zhai and Lafferty 2001a, 2004] that effective estimates of the probability of term occurrences for the multinomial document language model θ_{dm} can be found as follows:

$$p(t|\hat{\boldsymbol{\theta}}_{dm}) = (1 - \pi) \cdot p(t|\hat{\boldsymbol{\theta}}_d) + \pi \cdot p(t|\hat{\boldsymbol{\theta}}_c) \tag{3}$$

where $\hat{\boldsymbol{\theta}}_{dm}$ is the estimated smoothed document language model and π is a smoothing parameter which controls the amount of probability mass that should be redistributed from the background multinomial $p(t|\hat{\boldsymbol{\theta}}_c)$ to the document multinomial $p(t|\hat{\boldsymbol{\theta}}_d)$. This prevents over-fitting of the document model because in most retrieval formulations both $p(t|\boldsymbol{\theta}_d)$ and $p(t|\boldsymbol{\theta}_c)$ are estimated using maximum likelihood estimates $c(t, d)/|d|$ and $cf_t/|c|$ respectively. The background multinomial is estimated using all documents in the entire collection and therefore all tokens in the corpus are treated as independent observations. The background model can be viewed as the most likely single model to have generated *all* of the documents. It has been shown that the choice of smoothing greatly affects the retrieval effectiveness of the multinomial language model [Zhai and Lafferty 2004].

3.2. Smoothing

One of the simplest forms of smoothing uses linear-interpolation, also called Jelinek-Mercer smoothing, where π_{jm} is assigned a value in the range $(0 - 1)$. In this linear smoothing approach, the parameter is usually set by experimentally tuning π_{jm} on

training data. Typically there has been no guidance on the setting of this parameter as the effectiveness of this smoothing approach is quite sensitive to specific parameter values. However, a more effective smoothing method for the multinomial language model uses Bayesian smoothing in the form of a Dirichlet prior on the background multinomial. For this approach π_{dir} is defined as follows:

$$\pi_{dir} = \frac{\mu}{\mu + |d|} \qquad (4)$$

where μ is the concentration parameter and is the sum of the individual $|v|$-Dirichlet parameters. This concentration parameter is also assigned a value based on experimentation, though it has been found that it achieves a relatively stable performance when $\mu = 2000$ [Zhai and Lafferty 2004]. The Dirichlet prior parameter μ can be interpreted as the number of pseudo-counts of the background multinomial prior to the document data. Intuitively, this type of smoothing gives a greater credence to probability estimates that are derived from longer documents, compared to those derived from shorter documents, as the longer documents are likely to be more accurate representations of the document model. The prior parameters (pseudo-counts) of the $|v|$-component Dirichlet distribution are $\alpha_t = \mu \cdot p(t|\hat{\boldsymbol{\theta}}_c)$ for all $t \in v$ and are updated using the document observations to $\alpha_t = \mu \cdot p(t|\hat{\boldsymbol{\theta}}_c) + c(t, d)$ for all $t \in v$. Therefore, the concentration parameter of the Dirichlet distribution changes from μ to $\mu + |d|$ once the distribution has been updated. Throughout this article we will continue the convention of specifying a $|v|$-component Dirichlet using the parameters of a multinomial distribution (with $|v|$-1 degrees of freedom) multiplied by a concentration parameter (i.e. μ).

It has been shown that the query likelihood model with Dirichlet prior smoothing and the model with Jelinek-Mercer smoothing can be implemented as efficiently as traditional retrieval functions, which only use weights from terms that are common to both document and query[3].

4. A SMOOTHED PÓLYA URN DOCUMENT MODEL

In this section we first introduce the generalised Pólya urn model and outline some of its important characteristics. We then show how this can be used to model document generation before specifying the query likelihood approach for the new model. Finally, we outline how the parameters of the SPUD model are estimated and smoothed.

4.1. A Pólya Urn Process

Consider a process that starts with an urn containing m balls in total, where each ball is one of $|v|$ distinct colours. Starting at time $i = 0$, a ball is sampled with replacement from the urn, and a ball of the same colour is replicated and added to the urn. This process continues until $|d|$ balls have been sampled from the urn. The total number of balls in the urn at the end of the process is $m + |d|$. This is a typical description of the multivariate Pólya urn model which uses sampling with reinforcement. We use this process as a conceptual model for document generation, where the different colours represent distinct terms, where the initial counts of the $|v|$ different coloured balls in the urn represent the document model, and where the $|d|$ observations drawn represent the actual document.

This multivariate Pólya urn model has recently been described in an alternative manner as consisting of a multinomial and the Chinese restaurant process [Sunehag

---

[3]See the original source [Zhai and Lafferty 2004] for the derivations of these efficient retrieval functions.

2007; Goldwater et al. 2011]. Again, consider an urn that contains m balls of $|v|$ different colours, but now also consider a bag d that is initially empty. For all times starting at time $i = 0$, a ball is chosen from the urn with probability $m/(m+i)$ and from the bag with probability $i/(m+i)$, and each time it is replaced from where it was drawn. For each draw, a ball of the same colour that was drawn is generated and placed in the bag. In this alternative description, the number of balls m in the urn remains static, while the number of balls in the bag d is i at any particular time. The non-reinforced urn can be modelled as a multinomial and the bag can be modelled as the Chinese restaurant process.

This two-stage generative process has been outlined recently by Goldwater et al. [2011] and Sunehag [2007], and while the entire process is identical to the multivariate Pólya urn model described previous, it may be more intuitive in terms of a generative story of document creation. This is because the document is modelled as a separate entity that starts empty, and ends after $|d|$ terms have been drawn. We re-introduce the alternative description here only to motivate the application of this process to that of document generation. This is very much in the spirit of that proposed by Simon [1955] where an author generates a document by drawing words from some distribution and also by drawing words from those previously used in the document in order to create association. For the reminder of the article, when we refer to an *urn*, we mean a Pólya urn by default, unless otherwise stated.

It is well-known that the distribution of colours in the multivariate Pólya process follows the DCM (multivariate Pólya distribution). It is also known that the Pólya urn is an example of a bounded martingale process [Pemantle 2007], where the proportions of colours in the urn converges to a Dirichlet distribution. During the process, the drawing, subsequent replication, and addition of an observation (which must be identically distributed to the initial distribution) only serves to reinforce the initial distribution. Therefore, all subsequent balls drawn from the Pólya urn are identically distributed, but are not independent. Furthermore, the process is *exchangeable*, meaning that the ordering of the outcomes can be swapped to result in the same probability distribution. Therefore, the document model remains a bag-of-words because the ordering of the terms in the document is not modelled.

4.2. Document Generation as a Pólya process

We use the Pólya urn, and therefore the DCM, as a model for document generation where the author generates an actual document d by drawing $|d|$ terms from the reinforced document model. Intuitively, different documents are written in different styles (some styles exhibiting more word burstiness than others), and therefore, the degree of reinforcement will be document specific. Consequently, we assume that each document is drawn from a different document DCM, and therefore we need to estimate the parameters of a different document DCM for each document d.

The probability density function for the DCM is as follows:

$$p(d|\boldsymbol{\alpha}) = \int_{\boldsymbol{\theta}} p(d|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{\alpha})d\boldsymbol{\theta} \tag{5}$$

where $\boldsymbol{\alpha}$ is the initial $|v|$-dimensional parameter vector of a Dirichlet distribution. Conceptually, one can think of drawing a multinomial $\boldsymbol{\theta}$ from a Dirichlet distribution specified by $\boldsymbol{\alpha}$, and subsequently drawing a sample d from the multinomial. The parameters of the DCM can be interpreted as the initial number of instances of each coloured ball in the Pólya urn. Therefore, the sum of the DCM parameter vector $\sum_{t \in v} \alpha_t$ can be interpreted as the initial number of balls in the urn (i.e. $m_d = \sum_{t \in v} \alpha_t$) and is the concentration parameter. This is the factor that controls burstiness on a document level,

and when m_d is large the model exhibits low burstiness as adding balls to the urn changes the state of the urn very little. In fact, when $m_d \to \infty$, the DCM tends to the multinomial distribution (i.e. no burstiness) [Elkan 2006]. Conversely, if there are very few balls in the urn initially (i.e. $m_d \to 0$), the model exhibits high burstiness as the first ball drawn alters the initial state of the urn by reinforcement quite substantially. Therefore, the problem lies in estimating the initial parameters of the document DCM $\boldsymbol{\alpha}_d$ given that the document d was generated by this reinforced random process. For consistency, the notation we use to specify the $|v|$-dimensional parameter vector of the DCM is similar to that of the Dirichlet distribution (i.e. using a multinomial distribution and a concentration parameter).

Furthermore, given that documents only contain a subset of the terms in the collection, we do not wish to assign zero probabilities to terms that do not occur in a document. Therefore, we smooth each document DCM $\boldsymbol{\alpha}_d$ with a background DCM model $\boldsymbol{\alpha}_c$. The background model is the single model most likely to have generated *all* documents given our reinforced process, and therefore, we estimate the parameters of a background DCM $\boldsymbol{\alpha}_c$, given all of the n documents. There are different ways in which we can smooth these two DCM models and we will outline these in Section 4.6. In general, we are not restricted to smoothing only two DCM models to construct our document model, and any number of plausible DCM models could be combined to help explain observations in the document. However, in this article we confine ourselves to smoothing only two DCM models for each document d.

4.3. Non-Reinforced Query Likelihood

Once the parameters of the document model ($\mathcal{M}_d = \boldsymbol{\alpha}_{dm}$) have been estimated, we need to rank these document models with respect to a query. In the multinomial language model, both the document and query are assumed, for the purposes of ranking, to have been generated from a multinomial. This simplifies the estimation of the document model and the estimation of the query likelihood given the document model.

As mentioned earlier, we assume that documents and queries are generated differently. More specifically we assume that queries do not exhibit word burstiness. This in fact simplifies the query likelihood given our new document model. We assume that the documents are generated from a DCM document model $\boldsymbol{\alpha}_{dm}$, and that the query is generated from the document model (urn) using sampling with replacement (no reinforcement). Modelling query generation in this manner means that each term in the query is treated independently. Consequently, documents are ranked according to following query likelihood formula:

$$log \; p(q|\mathcal{M}_d = \mathbb{E}[\boldsymbol{\theta}_{dm}|\boldsymbol{\alpha}_{dm}]) = log \prod_{t \in q} p(t|\mathcal{M}_d)^{c(t,q)} = \sum_{t \in q} (log \; p(t|\mathcal{M}_d) \cdot c(t,q)) \quad (6)$$

where $\mathbb{E}[\boldsymbol{\theta}_{dm}|\boldsymbol{\alpha}_{dm}]$ is the expected multinomial of the DCM document model for document d.

4.4. Estimation of the Document DCM

We now estimate the parameters of the document DCM $\boldsymbol{\alpha}_d$ using the observations from the actual document d. Given only one sample (i.e. the document) it is not possible to fully specify the maximum likelihood estimates of the document DCM[4]. The maximum likelihood estimates of the multinomial inferred from one document will be equal to the expected value of the estimated DCM. Therefore, the maximum likelihood estimates

---

[4]The minimum number of samples needed to estimate both the expected value (a multinomial) and the concentration parameter (burstiness) is two.
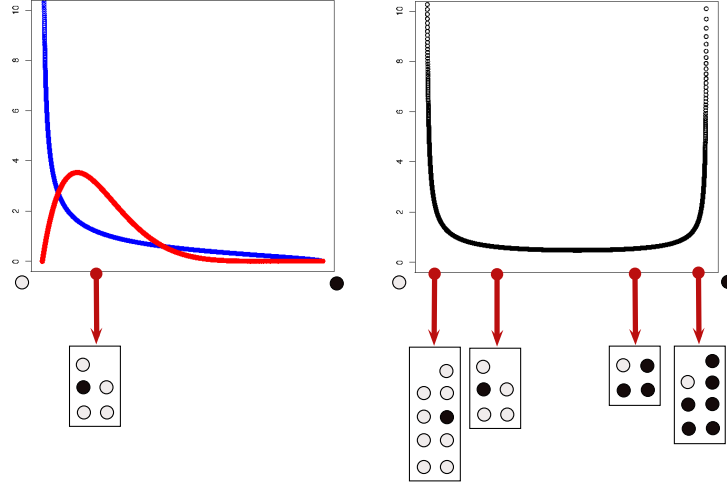
Fig. 1. Documents generated from multinomials drawn from a Dirichlet distribution for both document (left) and background language models (right)

of the multinomial from which the terms in the document were drawn (i.e. $c(t,d)/|d|$) will be proportional to the maximum likelihood estimates of the document DCM (i.e. $\hat{\theta}_d \propto \hat{\alpha}_d$). This is only true in the case where there is one sample.

Fig. 1 (left) shows this graphically for a simplified two-dimensional model that uses white and black balls to represent terms. The x-axis represents multinomials of varying parameter values. Points on the left-hand side of the x-axis represent multinomials where the probability of drawing a white ball are high, while points on the right-hand side of the x-axis represent multinomials where the probability of drawing a black ball are high. The Dirichlet distribution represents the likelihood of drawing these multinomials. In Fig 1 (left), the expectation of both of the two-dimensional Dirichlet distributions, shown by the red and blue curves, are equal and represent the multinomial (red arrow) inferred from the document.

Therefore, when we have only one multinomial (inferred from a document), we can only specify the location (expected multinomial) and not the shape (concentration parameter) of the DCM. In order to completely define the parameters of the document DCM, we also have to define the concentration $m_d = \sum_{t \in v} \alpha_{d_t}$, which can be interpreted as the level of belief associated with the maximum likelihood estimates of the expected multinomial. Therefore, the initial parameters of the $|v|$-component document DCM are estimated as follows:

$$\hat{\alpha}_d = m_d \cdot \hat{\theta}_d = (m_d \cdot p(t_1|d), m_d \cdot p(t_2|d), ..., m_d \cdot p(t_{|v|}|d)) \tag{7}$$

where $p(t|d) = c(t,d)/|d|$ for all $t \in v$ and where m_d is the initial mass that controls the burstiness of the document model. Although estimation of the parameters of the DCM using multiple data vectors is computationally expensive [Minka 2000], we can see that estimating the parameters of each document DCM is trivial if a suitable value for m_d can be found.

Given that m_d is the level of belief associated with the expected document multinomial $\hat{\alpha}_d$, it would seem intuitive to aim to minimise this belief in the absence of evidence (an Occam's razor type argument). A minimum setting can be arrived at by

determining the minimum initial number of balls in the urn that could have generated the document. Given a document d, the minimum number of balls initially in the urn is the number of distinct coloured balls drawn. Therefore, we estimate the concentration parameter m_d of the document DCM as $\hat{m}_d = |\vec{d}|$. This is the maximum amount of burstiness that is supported using this argument. In Fig. 1 (left) our estimate of m_d for the document model is $m_d = 2$, which leads to the shape of the Dirichlet in blue. Setting m_d according to this parsimonious principle ensures that we have not over-fitted to our data.

4.5. Estimation of the Background DCM

For the DCM document models, there exists dependencies between successive occurrences of the same term in a document, and therefore, the estimation of the background DCM is more complex than for the multinomial distribution. In fact, in the entire collection, the only occurrences of the same term that are independent of each other are those in different documents. This leads to the introduction of a document boundary into the background DCM of the new language model, something that is lacking in the multinomial language model.

The estimation of a background DCM using all n document vectors is, as mentioned previously, computationally expensive. However, Elkan [2006] has shown that, for textual data, very close approximations to the maximum likelihood estimates of the DCM (via the EDCM) are proportional to $\sum_{j=1}^{n} I(c(t, d_j) > 0)$ for all $t \in v$, where I is the indicator function. These approximations are accurate for textual data because most terms do not occur in all n documents, and furthermore, it has been shown that the approximations make little difference to the effectiveness of the model for text-related tasks. It can be seen that this approximation relates to the number of documents in which a term occurs (i.e. the document frequency[5]). Using an appropriate normalisation factor we obtain a probability estimate as follows:

$$p(t|\hat{\boldsymbol{\theta}'}_c) = \frac{\sum_{j=1}^{n} I(c(t, d_j) > 0)}{\sum_{t' \in v} df_{t'}} = \frac{df_t}{\sum_{t' \in v} df_{t'}} = \frac{df_t}{\sum_{j=1}^{n} |\vec{d_j}|} \tag{8}$$

where n is the number of documents in the collection and the numerator is the document frequency of a term. The normalisation factor can be re-written and comprises the summation of all document vectors in the collection so that $\sum_{t \in v} p(t|\hat{\boldsymbol{\theta}'}_c) = 1$.

This probability distribution can be viewed as the expected multinomial drawn from the background EDCM. The estimates of the background DCM, which are approximately proportional to these probability estimates, are defined in a similar manner to the document DCM by introducing one concentration parameter m_c. This results in the following parameter estimates for the background DCM:

$$\hat{\boldsymbol{\alpha}}_c = (m_c \cdot p(t_1|\hat{\boldsymbol{\theta}'}_c), m_c \cdot p(t_2|\hat{\boldsymbol{\theta}'}_c), ..., m_c \cdot p(t_{|v|}|\hat{\boldsymbol{\theta}'}_c)) \tag{9}$$

where m_c is the belief in the expected value of the Dirichlet (i.e. $p(t|\hat{\boldsymbol{\theta}'}_c)$) and can be interpreted as a type of document *word burstiness* throughout the collection.

Fig. 1 (right) shows a graphical example of a two-dimensional background Dirichlet. As before, the x-axis determines parameter values of the multinomials, and the black curve shows the likelihood of drawing these multinomials. The four document samples shown in the figure exhibit high levels of burstiness as they contain a disproportionate number of balls of one specific colour. This is because areas of higher likelihood in

---

[5]This introduces an idf-like measure into this language model and is discussed in a later in Section 5.2.4

Fig. 1 lead to multinomials with one component that contains most of the probability mass. The convex shape of this curve is due to a low m_c concentration parameter, and therefore models high levels of word burstiness. Although the expectation of this over-dispersed two-dimensional Dirichlet has a low likelihood, it is nonetheless *expected* in the statistical sense. Essentially, the use of a DCM explains greater term-frequency variation in the n documents in the collection.

4.6. Smoothing and Retrieval Models

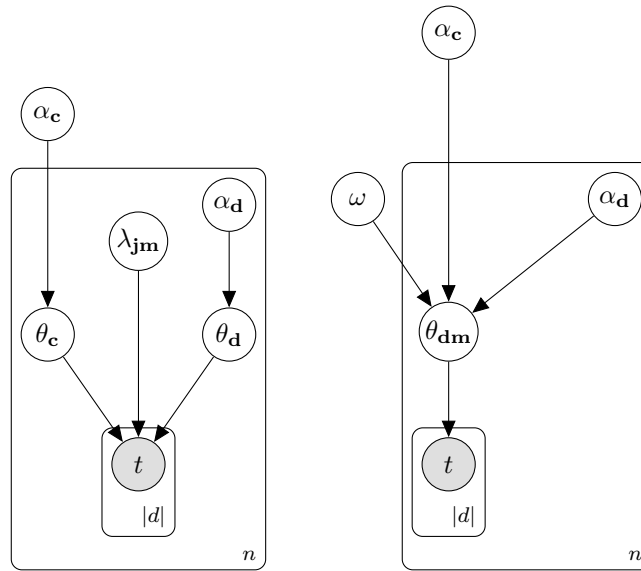We now present two smoothing methods which can be used to linearly combine K multiple DCM models.

Fig. 2.   Document generation in the SPUD model for both types of smoothing

4.6.1. Linear Smoothing of Expected Multinomials. Conceptually, both the background and document DCM can be thought of as a Pólya urn. The first approach to smoothing treats each of these models as distinct Pólya urns. A document is generated by drawing with reinforcement, balls from the K urns according to a certain probability. Essentially this smoothing approach linearly combines the expected values (multinomials) of the Dirichlets. This general smoothing approach is as follows:

$$p(d|\boldsymbol{\alpha}_{dm}) = \sum_{i=1}^{K} \lambda_i \cdot p(d|\boldsymbol{\alpha}_i) \qquad (10)$$

where $\sum_{i=1}^{K} \lambda_i = 1$ and $\boldsymbol{\alpha}_i$ is the ith DCM model. In this work we only linearly combine two models, the document DCM and the background DCM, and therefore the SPUD retrieval model using this smoothing approach is defined as follows:

$$\mathbb{E}[\boldsymbol{\theta}_{dm}|\boldsymbol{\alpha}_{dm}] = (1 - \lambda_{jm}) \cdot \boldsymbol{\theta}_d + \lambda_{jm} \cdot \boldsymbol{\theta'}_c \qquad (11)$$

where λ_{jm} is the smoothing parameter and can be interpreted as the probability of selecting a term from the background DCM. We note that this formulation is identical to that of Hiemstra [1998]. Fig. 2 (left) shows the graphical model for the DCM language model with this type of linear smoothing (Jelinek-Mercer).

One of the main motivations for smoothing the document model with a background model is that the background model assigns mass to terms unseen in the document. Therefore, λ_{jm} can be interpreted as the probability of drawing a previously unseen term from the background model, and $1 - \lambda$ as the probability of drawing a previously seen term (i.e. a repeated term from the document). During the generation of the document d, at least $|\vec{d}|$ previously unseen terms were drawn. This leads to an estimate of $\hat{\lambda}_{jm} = |\vec{d}|/|d|$ as the probability of drawing an unseen term for that document model. This is the proportion of distinct terms in the document and is the estimate of drawing from the background multinomial $\boldsymbol{\theta'}_c$. The SPUD retrieval model with this type of smoothing is denoted SPUD_{jm} and it has no free parameters. We note that the estimation of λ_{jm} is not a consequence of the DCM model, and can therefore be applied to the multinomial language model that uses Jelinek-Mercer smoothing.

4.6.2. Linear Smoothing of DCMs. The second approach to smoothing uses a linear mixture of the DCM models. Conceptually, this approach to smoothing combines the contents of the K urns into one single Pólya urn. A document is then generated by drawing with reinforcement from this single urn. This smoothing approach is a more complete Bayesian approach to smoothing and the parameters of the document model are as follows:

$$\boldsymbol{\alpha}_{dm} = \sum_{i=1}^{K} \omega_i \cdot \boldsymbol{\alpha}_i \tag{12}$$

where $\sum_{i=1}^{K} \omega_i = 1$ and $\boldsymbol{\alpha}_i$ is the ith DCM language model. The ω parameters are linear mixing parameters that determine the relative weight of the DCM language models. It is worth noting that each of the DCMs has a concentration parameter m_i which act to weight the vector appropriately. Given the document DCM and background DCM estimated previously, the smoothing is as follows:

$$\hat{\boldsymbol{\alpha}}_{dm} = (1 - \omega) \cdot m_d \cdot \hat{\boldsymbol{\theta}}_d + \omega \cdot m_c \cdot \hat{\boldsymbol{\theta'}}_c \tag{13}$$

where ω is the linear mixing parameter. Fig. 2 (right) shows the graphical model for this DCM mixture model. The expected multinomial drawn from this DCM mixture model is easily computed using the individual parameters of the DCM mixture model over the normalisation constant. This DCM mixture retrieval model is denoted SPUD_{dir} due to the mixing of the Dirichlets. Although it seems that the DCM mixture model still has two unknown parameters (i.e. m_c and ω), these can either be combined to form one single parameter[6], or m_c can be estimated using numerical methods as outlined in the original work introducing the EDCM [Elkan 2006]. We outline the details of these approaches in the next section.

5. RETRIEVAL MODEL IMPLEMENTATION

In this section we outline the composition of the SPUD retrieval methods using both types of smoothing presented in the preceding section. We then present some retrieval intuitions that aid in understanding the retrieval aspects of the new model.

———————

[6]This is analogous to the tuning parameter μ in the multinomial language model using Dirichlet prior smoothing.

5.1. Retrieval Functions

Similarly to the implementation of the standard multinomial models [Zhai and Lafferty 2004], our approach can be computed efficiently using a summation that only involves terms common to both document and query. The SPUD_{jm} retrieval function using linear smoothing is as follows:

$$\text{SPUD}_{jm}(q,d) = \sum_{t\in q}(log((1-\hat{\lambda}_{jm})\cdot\frac{c(t,d)}{|d|}+\hat{\lambda}_{jm}\cdot\frac{df_t}{\sum_j^n|\vec{d_j}|})\cdot c(t,q)) \quad (14)$$

where $\hat{\lambda}_{jm} = |\vec{d}|/|d|$. This is rank equivalent to the following:

$$\text{SPUD}_{jm}(q,d) = |q|\cdot log(\hat{\lambda}_{jm}) + \sum_{t\in q\cap d}(log(1+\frac{(1-\hat{\lambda}_{jm})\cdot c(t,d)\cdot\sum_j^n|\vec{d_j}|}{|\vec{d}|\cdot df_t})\cdot c(t,q)) \quad (15)$$

The SPUD_{dir} retrieval function can be computed in a somewhat similar form to the multinomial language model using Dirichlet prior smoothing as follows:

$$\text{SPUD}_{dir}(q,d) = \sum_{t\in q}(log(\frac{(1-\omega)\cdot|\vec{d}|\cdot\frac{c(t,d)}{|d|}+\omega\cdot m_c\cdot\frac{df_t}{\sum_j^n|\vec{d_j}|}}{(1-\omega)\cdot|\vec{d}|+\omega\cdot m_c})\cdot c(t,q)) \quad (16)$$

which is rank equivalent to the following:

$$\text{SPUD}_{dir}(q,d) = |q|\cdot log(\frac{\mu'}{\mu'+|\vec{d}|}) + \sum_{t\in q\cap d}(log(1+\frac{|\vec{d}|\cdot c(t,d)\cdot\sum_j^n|\vec{d_j}|}{\mu'\cdot|d|\cdot df_t})\cdot c(t,q)) \quad (17)$$

where μ' is a combination of ω and m_c as follows:

$$\mu' = \frac{\omega}{1-\omega}\cdot m_c \quad (18)$$

As μ' is the only parameter that has not been estimated so far, we now outline two approaches to finding suitable values for it. The first approach is to experimentally tune μ' on training data in a similar manner to the Dirichlet prior smoothing parameter μ in the multinomial language model [Zhai and Lafferty 2001a]. Alternatively, m_c can be estimated from the n samples of observations using Newton's method [Elkan 2006] as follows:

$$m_c^{new} = \frac{\sum_j^n|\vec{d_j}|}{\sum_j^n\psi(|d_j|+m_c)-n\cdot\psi(m_c)} \quad (19)$$

where $\psi(x) = \frac{d}{dx}log\Gamma(x)$ is the digamma function and Γ is the gamma function. When estimating m_c from the data using this method, ω is the parameter that requires experimental tuning. However, we expect that the one setting for the hyperparameter ω will perform robustly across many test collections. Experiments for both of these approaches to determining a suitable values the free parameters are are outlined in Section 6.4.

5.2. Length Normalisation and Document Boundary Retrieval Intuitions

We now examine some retrieval intuitions and existing hypotheses that help explain the differences between the SPUD retrieval functions and the multinomial retrieval

functions. For most of the analysis in this section we focus on the best performing multinomial model (MQL_{dir}) and its counterpart from the SPUD model ($SPUD_{dir}$). Robertson and Walker [1994] outlined two hypotheses concerning the length of a document, namely the *verbosity hypothesis* and the *scope hypothesis*, which we now examine.

5.2.1. Verbosity Hypothesis. The *verbosity hypothesis* captures the intuition that some documents are longer than others simply because they are more verbose. Such documents do not describe more topics, they are simply more *wordy*. This hypothesis captures an aspect of document length that is independent of relevance. However, the initial description of this hypothesis [Robertson and Walker 1994] does not outline any formal means of determining whether a particular retrieval function is consistent with the hypothesis. We now outline a retrieval constraint[7] which helps to determine this.

LNC2*. *If document d and d' are two documents, where d' is constructed by concatenating d with itself k times where $k > 0$, and if $s(q, d)$ is the score returned from a retrieval function s which is used to rank d with respect to q, then $s(q, d) = s(q, d')$.*

This states that if a document is concatenated with itself any number of times, the retrieval score of that document should not change for a given query, and therefore it should not change rank. We call this constraint LNC2* as this is stricter than LNC2 outlined by Fang et al. [2004], which only states that $s(q, d) \leq s(q, d')$. Essentially if a scoring function s adheres to LNC2*, then we deem s to be consistent with the verbosity hypothesis.

Consider a relevant document d that is ranked in a certain position according to $s(q, d)$. If d is replaced in the collection with d', d' should not be ranked lower than the initial document d. Therefore, $s(q, d')$ should certainly not be less than $s(q, d)$ simply due to the verbosity of d'. Now consider a non-relevant document d of a given length. If d is replaced in the collection with d', d' should not be ranked in a higher position than d originally was. Therefore, given that we do not know the relevance of d a priori, we argue that in general $s(q, d')$ should not increase simply due to the increased verbosity of d'.

The maximum likelihood estimate of a term in a document (i.e. $c(t, d)/|d|$) will not change if that document is concatenated with itself any number of times. However, in the multinomial language model using Dirichlet priors smoothing (MQL_{dir}), LNC2* is only satisfied when $c(t, d)/|d| = cf_t/|c|$ which is not often the case. For this model, if there are many query-term matches in d, the more verbose document d' will nearly always be ranked higher than d (i.e. $s(q, d') > s(q, d)$)[8], while if there are very few query-term matches in d the verbose document d' will nearly always be ranked lower than d (i.e. $s(q, d') < s(q, d)$). However, if we examine the $SPUD_{dir}$ method in Eq. (17), we can see that the document vector length $|\vec{d}|$ is used as one form of document length normalisation. The document vector length $|\vec{d}|$ will remain unchanged for the concatenated document d', and therefore $SPUD_{dir}(q, d) = SPUD_{dir}(q, d')$.

In general the multinomial model not only over-promotes recurrences of query terms but over-penalises recurrences of non-query terms in a given document. Fig. 3 (left) shows the increase in weight as the term-frequency increases for both MQL_{dir} and

---

[7]Just prior to publication we found that a similar constraint has been previously been outlined in [Na et al. 2008].

[8]We note that we are ignoring the effect that creating a longer document d' would have on the background collection model. For an extremely large collection this effect would be negligible. Furthermore, we note that the multinomial language model that uses Jelinek-Mercer smoothing adheres to LNC2*, while the $SPUD_{jm}$ does not. However, there are other reasons for the generally weaker performance of the standard multinomial language model with Jelinek-Mercer smoothing.

$SPUD_{dir}$. We can see that MQL_{dir} gives a greater weight to terms with higher frequencies than $SPUD_{dir}$. This is because the aspect of document length that is affected as term-frequency increases is different for both retrieval functions. It is important that term-frequency is analysed considering the change in document length that an increase in term-frequency brings about. Fig. 3 (right) also shows the penalisation due to recurrences of non-query terms for both MQL_{dir} and $SPUD_{dir}$. We can see that MQL_{dir} penalises recurrences of non-query terms more than $SPUD_{dir}$. In the $SPUD_{dir}$ function, recurrences of the same non-query term will always decrease the score of a document due to more off-topic verbosity.
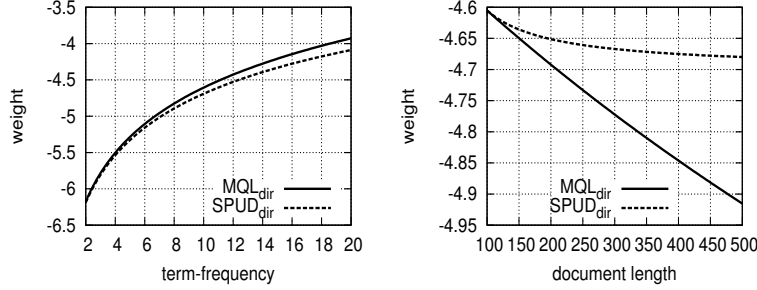

Fig. 3. Change in weight as term-frequency increases for MQL_{dir} and $SPUD_{dir}$ in a document that initially contains 100 distinct terms (left). Change in weight as recurring non-query terms are added to a document that initially contains 100 distinct terms (right).

Interestingly, it can be seen that the $SPUD_{dir}$ formula in Eq. (17) contains the ratio between the term-frequency $c(t, d)$ and the average term-frequency in the document $|d|/|\vec{d}|$. This average term-frequency normalisation idea was first proposed by Singhal et al. [1996], but in general was not shown to improve retrieval effectiveness substantially until recent research [Paik 2013]. It is this part of the $SPUD_{dir}$ retrieval model that deals specifically with the verbosity hypothesis, while the document length normalization component, the left-hand side of Eq. (17), deals with the scope hypothesis by replacing the original document length with the document vector length. We now discuss this further.

5.2.2. Scope Hypothesis. The *scope hypothesis* captures the alternative intuition that documents may be longer because they cover many different topics. It has been noted in the original work regarding the scope hypothesis [Robertson and Walker 1994] that many Newswire documents in the original TREC corpora seemed as if they consisted of multiple different news articles concatenated together. In the multinomial language model there is no difference in the normalisation applied when a term occurs for the first time (i.e. an increase in scope) as opposed to when a term repeats itself (i.e. an increase in verbosity). This difference is modelled in the new SPUD language models and can be viewed as being modelled separately for $SPUD_{dir}$. In Eq. (17), we can see that the factor $|q| \cdot log(u'/(u' + |\vec{d}|))$ leads to a penalisation only for the occurrence of distinct terms (i.e. when the scope broadens[9]). If the term re-occurs, it is not penalised by the part of the retrieval function which deals with scope.

For the $SPUD_{dir}$ model, adding a non-query term into a document for the first time will lead to penalisation by the normalisation aspect that deals with scope. However, it

---

[9] We assume that the number of distinct terms in a document is a crude measure of scope.

should be noted that the verbosity aspect of a document is also affected by the addition of previously unseen non-query terms and this actually promotes existing query-terms. Therefore, the overall document score does not necessarily decrease when a new non-query term is added.

In the SPUD_{dir} model, the magnitude of the document score penalisation for the first occurrence of a non-query term is quite similar to the penalisation applied by MQL_{dir} (See Fig. 3), but recurrences are *not* penalised as much. Given these observations, we hypothesise that the SPUD_{dir} retrieval method does not penalise long documents as much as MQL_{dir}. Recent research has studied the over penalisation of long documents by many retrieval functions including MQL_{dir} [Lv and Zhai 2011]. They built upon work by Singhal et al. [1996] which showed that most ranking functions retrieve long documents with a likelihood less than their likelihood of relevance. We replicate that analysis by binning according to length, relevant documents and then estimating the probability that a document occurs in a given bin (length) given that it is relevant. The same procedure is applied to retrieved documents where a document is deemed retrieved if it occurs in the top 1000 documents of the ranked list. We use the same binning strategy as Lv and Zhai [2011] (i.e. 5000) and compared the MQL_{dir} and SPUD_{dir} retrieval functions. The aspect of length used in this analysis is that of the number of word tokens in the document (i.e. $|d|$).
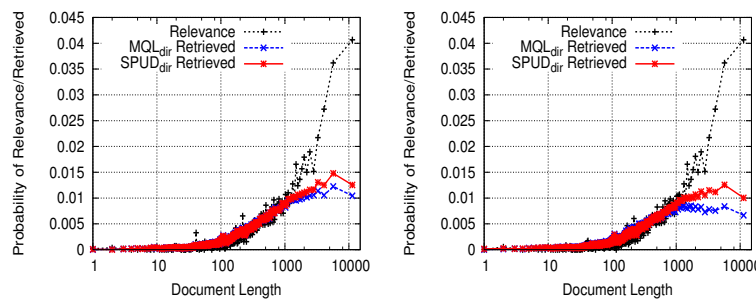

Fig. 4. Probability of retrieval/relevance for MQL_{dir} and SPUD_{dir} methods for trec-9/10 collection for short queries (left) and medium length queries (right).

Fig. 4 shows the probability of relevance (in black) and the probability of retrieval for both MQL_{dir} (in blue) and SPUD_{dir} (in red) on one collection. Firstly, we note that the trends are consistent with the previous approaches [Singhal et al. 1996; Lv and Zhai 2011]. Furthermore, we can see that longer documents have a higher likelihood of being retrieved by the SPUD_{dir} approach compared to the MQL_{dir} approach. This confirms our intuitions that the SPUD_{dir} model does not penalise long documents as much as MQL_{dir} and that we would expect the SPUD method to retrieve long documents with a probability closer to their likelihood of relevance. We investigate this further in the experimental section (Section 6.5).

5.2.3. Background model. The new background model in the SPUD brings about some other interesting retrieval characteristics. Given the sample collection in Table II of four documents and two terms (t_1 and t_2), we might wish to determine the most likely one term string, $q = \{t_1\}$ or $q = \{t_2\}$, generated from the background model. If we assume a multinomial background model estimated using maximum likelihood, then $p(t_1|\hat{\boldsymbol{\theta}}_c) = 8/15$ and $p(t_2|\hat{\boldsymbol{\theta}}_c) = 7/15$, suggesting that term $q = \{t_1\}$ is the more likely. However, intuitively we see that the high frequency of t_1 in document d_1 is unduly

Table II. Sample collection
of four documents and two
terms

docs	t_1	t_2
d_1	8	2
d_2	0	1
d_3	0	3
d_4	0	1

biasing the estimates, especially as term t_1 only appears in one document. Term t_2 occurs in all of the documents, and therefore, is a word used more widely in the collection (possibly by more authors in general). The SPUD model takes the document boundary into account yielding estimates of $p(t_1|\hat{\boldsymbol{\theta}'}_c) = 1/5$ and $p(t_2|\hat{\boldsymbol{\theta}'}_c) = 4/5$ respectively. This probability is similar to that proposed in one of the first language modelling approaches [Hiemstra 1998], and has recently been re-examined as being potentially theoretically valid [Roelleke 2012].
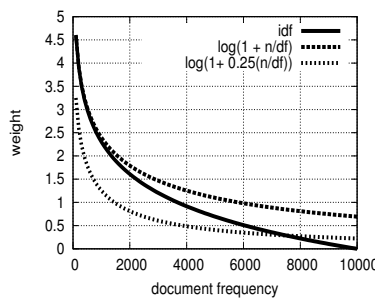

Fig. 5. idf and global weightings derived from the SPUD model

As seen in this toy example, the proposed model uses the document frequency in its approximation for the parameters of the background DCM. Furthermore, the normalisation component used in the background model can be written as $\sum_j^n |\vec{d_j}| = n \cdot |\vec{d}|_{avg}$, where $|\vec{d}|_{avg}$ is the average document vector length. Therefore, in the SPUD_{dir} retrieval formula, the weight assigned to a query-term that occurs in a document comprises of the following factor as per the right-hand side of Eq. (17):

$$\sum_{t \in q \cap d} log(1 + \delta \cdot \frac{n}{df_t}) \cdot c(t,q) \tag{20}$$

where $\delta = |\vec{d}| \cdot |\vec{d}|_{avg} \cdot c(t,d)/(\mu' \cdot |d|)$. We can see that this factor can be viewed as a new family of idf. Unlike the traditional idf measure, this factor is document-length, document-vector-length, and term-frequency specific[10]. We have found that δ typically ranges from 0.05 to 0.5 for query terms on many of the collections used in this work. Fig. 5 shows the weight assigned by idf and by Eq. (20) as the document frequency changes. This suggests that the global weighting factor in our new approach is closely

---

[10]We note that Sunehag [2007] has previously derived the traditional idf from a Pólya process using slightly different assumptions.

related to idf. This crude comparison by no means validates the traditional idf in a theoretical perspective, nevertheless it does present a theoretical means by which aspects of both term-frequency and document frequency combine in one model. In contrast, the multinomial language modelling approach treats terms that are completely independent of each other, written by different authors on different topics, similarly to terms that are highly dependent on each other (e.g. terms that are repeated, possibly due to *association*, in a document written by one author on a particular topic). Discovering a theoretical justification for the combination of both term-frequency and idf is problematic as they appear to lie in different event spaces[11]. The *preferential attachment* captured in the SPUD model is a promising generative theory justifying tf-idf type schemes.

A practical consideration is whether there is substantial difference between the probability of a term given either background model (multinomial or DCM) when estimated from data. Therefore, we estimated the background probability of seeing a term for both models for all query terms on one of the test collections used in our experiments. We analysed 1530 query-terms from the trec-9/10 test collection and we found a high linear correlation (0.954) between the estimated probabilities for the terms. This is to be expected as the probabilities are fundamentally capturing similar information about a term. However, there are examples where the estimated probabilities of actual query-terms are quite different. Table III shows the top and bottom 10 terms when ranked according to the ratio of their probabilities (i.e. $p(t|\alpha_c)/p(t|\theta_c)$). The bottom 10 terms show those that the background multinomial gives a much higher probability to when compared to the background DCM. It is interesting that the term *el*, which has much higher probability in the multinomial model, is a stopword from a different language. This receives a relatively high probability estimate from a multinomial because it appears many times, but receives a much lower probability estimate from the background DCM because many of these appearances come from few documents (i.e. the term is quite bursty). The background DCM regards these terms as less general than the multinomial, as the occurrences have actually occurred in fewer documents. Conversely, the top 10 terms show those that the multinomial model has estimated as less general but which the background DCM has estimated as being more general. These terms are less bursty but have occurred in many documents in the collection.

Therefore, given that there exists query-terms where the probability of occurrence under our new model is quite different, we would expect this to impact retrieval effectiveness. We evaluate the effect that the new document normalisation and background model have on retrieval effectiveness separately in Section 6.5.

6. EXPERIMENTS

In this section we outline the experiments used to evaluate the new SPUD methods. We first outline the experimental design and methodology, before presenting the experiments.

6.1. Experimental Design

We carry out four experiments to evaluate different aspects of the new SPUD query likelihood models. The first experiment evaluates the retrieval effectiveness of the new SPUD retrieval methods against a number of baselines. The second experiment evaluates the robustness of the tuning parameters in the SPUD retrieval methods. The third experiment presents an analysis of the retrieval intuitions outlined in the preceding section. Finally, we evaluate the best SPUD retrieval method when incorporating it into a pseudo-relevance feedback framework.

---

[11]See [Robertson 2004] for a thorough review of theoretical attempts to justify idf with term-frequency.

Table III. Ratio of estimated query-term probabilities of DCM to multinomial model $(p(t|\boldsymbol{\alpha}_c)/p(t|\boldsymbol{\theta}_c))$ for a number of query-term on trec-9/10

terms	Bottom 10		Top 10	
1	vike	0.3461802368	funnel-shap	2.0327764077
2	el	0.3910938927	undergon	1.9616724275
3	cancer	0.4098663257	pejor	1.9517391904
4	patient	0.415028517	tartin	1.9495633384
5	cell	0.4180174157	superstiti	1.9149743114
6	student	0.4289654263	gynt	1.9071815267
7	drug	0.4726560064	interest	1.8928123508
8	system	0.4950172629	unsuccess	1.8803346212
9	law	0.5064728539	work	1.8709780476
10	infect	0.51433562	run-awai	1.8666031964

6.2. Datasets

Table IV shows the characteristics of the TREC[12] test collections used in the experiments. We use a wide variety of TREC collections that are of varying sizes and include collections of Web documents, Newswire articles, and medical abstracts. In our experiments we evaluate short keyword queries (2-3 terms) consisting of the title field of the trec topic, medium queries (6-10 terms) consisting of both the title and description fields of the topics, and long verbose queries (10-30 terms) consisting of the title, description, and narrative fields of the topic. We remove standard stopwords and apply stemming using Porter's stemmer. It is worth noting that the ohsumed test collection contains only description length queries (i.e. medium length queries), while there are only title length queries available for the mq-07 and mq-08 test collections.

Table IV. Test Collection Details

label	collection	# docs	# topics	topic range	query length		
					short (title)	medium (title+desc)	long (title+desc+narr)
ohsu	ohsumed	293,856	63	001-63	n/a	5.0	n/a
robust-04	fr, ft,la, fbis	528,155	250	301-450, 601-700	2.5	10.3	31.4
trec-8	wt2g	221,066	50	401-450	2.4	9.0	27.5
trec-9/10	wt10g	1,692,096	100	451-550	2.6	9.3	24.3
gov2	gov2	25,205,179	150	701-850	2.8	8.6	33.3
mq-07	gov2	25,205,179	1778	1-10k	3.1	n/a	n/a
mq-08	gov2	25,205,179	784	10k - 20k	3.7	n/a	n/a

6.3. Retrieval Effectiveness

The first experiment evaluates the retrieval effectiveness of the SPUD model against its counterpart, the standard multinomial query likelihood language model. We compare the SPUD_{jm} retrieval function against the multinomial query likelihood function with Jelinek-Mercer smoothing (MQL_{jm}). We tune the MQL_{jm} function for each set of queries to optimise mean average precision (MAP) on each test collection where the parameter space $\pi_{jm} \in \{0.1, 0.2, ..., 0.9, 1.0\}$. Therefore, we are confident that the effectiveness of the MQL_{jm} retrieval function is close to its optimal on each collection. On the other hand, we do not tune SPUD_{jm} as it has no free-parameters.

We compare the SPUD_{dir} retrieval function against its counterpart, the multinomial query likelihood function with Dirichlet-prior smoothing (MQL_{dir}). Similarly, we tune

---

the MQL_{dir} function to optimise MAP on each test collection where the parameter space $\mu \in \{250, 500, ..., 2250, 2500\}$. We report the effectiveness of the SPUD_{dir} retrieval function for same parameter setting as MQL_{dir} (i.e. $\mu = \mu'$). This evaluation favours MQL_{dir} as SPUD_{dir} may not be tuned optimally.

We also use the DCM-L-T retrieval function [Xu and Akella 2008] which has a tuning parameter γ. We tuned γ for each set of queries on each collection over the parameter space $\gamma \in \{0.1, 0.2, ..., 0.9, 1.0\}^{13}$.

6.3.1. Retrieval Effectiveness Results. Tables V and VI show the retrieval effectiveness (MAP and NDCG@20) of MQL_{jm} compared to SPUD_{jm}, and MQL_{dir} compared to SPUD_{dir} for short title queries (2-3 terms on average). We can see that on most of the test collections the SPUD retrieval methods demonstrate a significant increase in effectiveness for both MAP and NDCG@20 over their corresponding MQL methods.

Table V. MAP of SPUD models vs MQL models (▲ means two-sided t-test $p < 0.01$, △ means $p < 0.05$) and SPUD models vs DCM-L-T (● means two-sided t-test $p < 0.01$ compared to DCM-L-T, ○ means $p < 0.05$ compared to DCM-L-T).

	short queries					
	robust-04	trec-8	trec-9/10	gov2	mq-07	mq-08
DCM-L-T	0.248	0.306	0.187	0.288	0.409	0.413
MQL_{jm}	0.231	0.246	0.135	0.245	0.396	0.419
SPUD_{jm}	0.236	0.255	0.154△	0.276▲	0.411▲	0.430▲
MQL_{dir}	0.247	0.308	0.192	0.303	0.420	0.427
SPUD_{dir}	0.252▲	0.319△	0.200△	0.314▲●	0.431▲●	0.445▲○

Table VI. NDCG@20 of SPUD models vs MQL models (▲ means two-sided t-test $p < 0.01$, △ means $p < 0.05$) and SPUD models vs DCM-L-T (● means two-sided t-test $p < 0.01$ compared to DCM-L-T, ○ means $p < 0.05$ compared to DCM-L-T).

	short queries					
	robust-04	trec-8	trec-9/10	gov2	mq-07	mq-08
DCM-L-T	0.423	0.449	0.298	0.455	0.465	0.495
MQL_{jm}	0.385	0.356	0.220	0.379	0.458	0.495
SPUD_{jm}	0.398 ▲	0.384△	0.243△	0.418▲	0.474△	0.503
MQL_{dir}	0.423	0.466	0.309	0.470	0.488	0.500
SPUD_{dir}	0.432 ▲	0.477	0.322	0.492▲●	0.500▲○	0.513▲○

Tables VII and VIII show the retrieval effectiveness (MAP and NDCG@20) of MQL_{jm} compared to SPUD_{jm}, and MQL_{dir} compared to SPUD_{dir} for medium length queries (6-10 terms on average). Again we can see that on most of the test collections the SPUD models demonstrate an increase in effectiveness for both MAP and NDCG@20. All of these increases are significant in the case of SPUD_{dir}. For long queries (10-30 terms on average) we see a similar trend. A point worth emphasising is that the increases in effectiveness are also present at the top of the ranked lists as demonstrated by NDCG@20.

The SPUD_{dir} approach outperforms the previous DCM relevance-based model (DCM-L-T) on most test collections. We have found that the DCM-L-T performs similarly to MQL_{dir} for short queries on some of the smaller collections, but we find that the DCM-L-T approach performs quite poorly on the larger gov2, mq-07, and mq-08

---

[13]The original paper does not outline a recommended parameter space. However when tuning from $0.1-1.0$, a maximum stationary point for effectiveness was found for each set of queries.

test collections and for all medium and long queries. Statistical significance tests (using a two-sided t-test indicated by ∘ and •) show that the best performing SPUD model (SPUD$_{dir}$) outperforms the DCM-L-T approach on some collections for short queries and consistently outperforms a tuned DCM-L-T approach for longer queries. We discuss some possible reasons for these results in Section 7.1.

Table VII. MAP of SPUD models vs MQL models (▲ means two-sided t-test $p < 0.01$, △ means $p < 0.05$) and SPUD models vs DCM-L-T (• means two-sided t-test $p < 0.01$ compared to DCM-L-T, ∘ means $p < 0.05$ compared to DCM-L-T).

	medium length queries				
	robust-04	trec-8	trec-9/10	gov2	ohsu
DCM-L-T	0.266	0.296	0.181	0.256	0.255
MQL$_{jm}$	0.277	0.283	0.191	0.276	0.239
SPUD$_{jm}$	0.280	0.291	0.203▲∘	0.299▲•	0.248△
MQL$_{dir}$	0.281	0.325	0.238	0.315	0.253
SPUD$_{dir}$	0.289▲•	0.347▲•	0.247▲•	0.329▲•	0.270▲•

Table VIII. NDCG@20 of SPUD models vs MQL models (▲ means two-sided t-test $p < 0.01$, △ means $p < 0.05$) and SPUD models vs DCM-L-T (• means two-sided t-test $p < 0.01$ compared to DCM-L-T, ∘ means $p < 0.05$ compared to DCM-L-T).

	medium length queries				
	robust-04	trec-8	trec-9/10	gov2	ohsu
DCM-L-T	0.435	0.436	0.318	0.401	0.396
MQL$_{jm}$	0.455	0.412	0.329	0.431	0.397
SPUD$_{jm}$	0.456	0.440△	0.344△•	0.463▲•	0.391
MQL$_{dir}$	0.465	0.478	0.393	0.484	0.399
SPUD$_{dir}$	0.479▲•	0.500▲•	0.403△•	0.502▲•	0.415▲∘

Table IX. MAP of SPUD models vs MQL models (▲ means two-sided t-test $p < 0.01$, △ means $p < 0.05$) and SPUD models vs DCM-L-T (• means two-sided t-test $p < 0.01$ compared to DCM-L-T, ∘ means $p < 0.05$ compared to DCM-L-T).

	long queries			
	robust-04	trec-8	trec-9/10	gov2
DCM-L-T	0.239	0.225	0.181	0.235
MQL$_{jm}$	0.284	0.269	0.211	0.265
SPUD$_{jm}$	0.288△•	0.269 •	0.206 ∘	0.285 ▲•
MQL$_{dir}$	0.283	0.283	0.248	0.296
SPUD$_{dir}$	0.296▲•	0.314▲•	0.254 •	0.323 ▲•

Fig.6 shows the performance of MQL$_{dir}$ vs SPUD$_{dir}$ for each query on two separate test collections. This query specific analysis indicates that SPUD$_{dir}$ is robust on all ranges of queries (from easy to difficult). For longer queries on the robust-04 dataset, there are one or two high performing queries which drop over 0.1 in average precision. However, in general there are very few queries which severely under perform compared to MQL$_{dir}$. On the trec-9/10 web documents, the increase in performance is stable across all types of queries for all query lengths.

Table X. NDCG@20 of SPUD models vs MQL models (▲ means two-sided t-test $p < 0.01$, △ means $p < 0.05$) and SPUD models vs DCM-L-T (● means two-sided t-test $p < 0.01$ compared to DCM-L-T, ○ means $p < 0.05$ compared to DCM-L-T).

	long queries			
	robust-04	trec-8	trec-9/10	gov2
DCM-L-T	0.404	0.346	0.318	0.400
MQL_{jm}	0.469	0.430	0.354	0.535
SPUD_{jm}	0.476 ●	0.431 ●	0.356 ●	0.541 ●
MQL_{dir}	0.467	0.446	0.406	0.572
SPUD_{dir}	0.483▲●	0.475△●	0.409 ●	0.599 ▲●

Fig. 6. Average precision of all short, medium, and long queries for MQL_{dir} vs SPUD_{dir} on robust-04 dataset (top) and trec-9/10 (bottom)

6.4. Robustness

The second experiment evaluates the robustness of the SPUD models with respect to different parameter settings. In addition, we evaluate the retrieval effectiveness of the SPUD_{dir} model when the parameter μ' is derived from the estimated parameter m_c using Newton's method [Elkan 2006].

6.4.1. Robustness Results. Fig. 7 shows the performance of MQL_{jm} over different tuning parameter values (i.e. π_{jm}) and the performance of the SPUD_{jm} model. We can see that SPUD_{jm}, which has no free parameters, outperforms SPUD_{jm} over all parameter values. This trend is consistent on all test collections used here.

For the SPUD_{dir} function, we can estimate m_c using Newton's method as outlined in Eq. (19) given the n documents as data. We found that an initial value of $m_c = 200$ was suitable so that the process converged within 20 iterations. This computation can be done off-line and we used the resulting setting of m_c to estimate μ' by tuning the hyperparameter ω to a fixed value. We set $\omega = 0.8$ which is demonstrated in Fig. 8 as a reasonable setting. Fig. 8 shows the performance (MAP) of the SPUD_{dir} model for different values of ω when m_c is estimated using Newton's method. The relationship between ω and μ' in Eq. (18) essentially suggests that $\mu' = 4 \cdot m_c$ is a suitable parameter
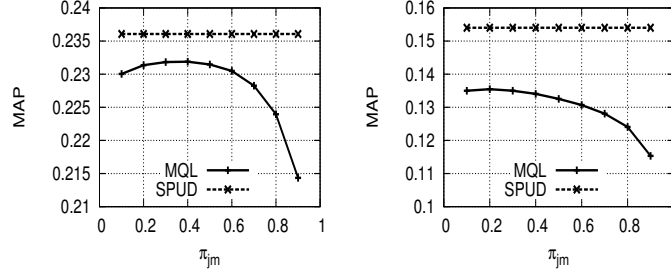
Fig. 7. Robustness comparison of MQL_{jm} and SPUD_{jm} on robust-04 (left) and trec-9/10 (right) for short queries

value for μ'. Although m_c is the only parameter that is expensive to estimate in the SPUD_{dir} model, it is practically feasible to do so offline. When the parameter μ' is computed in this way (i.e. $\mu' = 4 \cdot m_c$), we denote this $\text{SPUD}_{est_{\mu'}}$ in the experimental results and figures that follow.
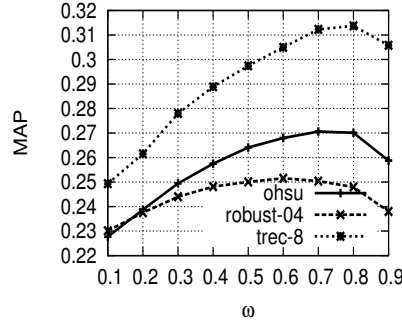

Fig. 8. Tuning of ω in SPUD_{dir} on robust-04 (short), trec-8 (short), and ohsu (medium) collections when m_c is estimated using Eq. (19)

Fig. 9 shows the performance of MQL_{dir}, SPUD_{dir}, and $\text{SPUD}_{est_{\mu'}}$ over different parameter settings on a number of test collections. We can see that SPUD_{dir} outperforms MQL_{dir} over all parameter values. We can see that the parameter μ' is as robust as the parameter μ, as it tends to follow the same trend.

More importantly we see that near optimal effectiveness can usually be achieved by using the automatically estimated value of m_c found using Newton's method. This is rather encouraging as it means that the setting of $\omega = 0.8$ is robust and that we can effectively and safely eliminate from SPUD_{dir} the free parameters. In particular, this automatic optimal estimation can be seen when we examine in Fig. 9 the trec-9/10 collection (which contains long Web documents) and the robust-04 collection (which has shorter documents). For the robust-04 collection, the retrieval effectiveness decreases sharply when μ' becomes greater than 1000. On the other hand, for the trec-9/10 the effectiveness is more stable when μ' is greater than 1000. One probable reason for this is that the average length of the documents in those collections is very different. However, the automatically estimated $\text{SPUD}_{est_{\mu'}}$ is close to optimal on both collections.
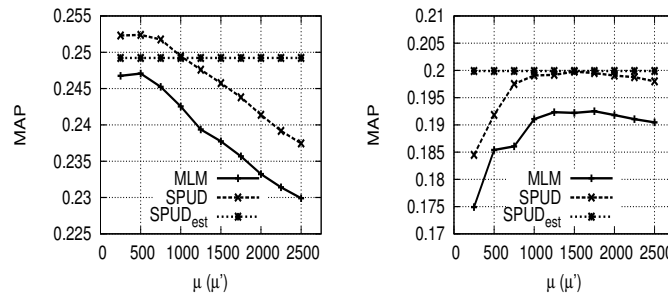
Fig. 9. Robustness of SPUD$_{dir}$ and MQL$_{dir}$ over different values of the tuning parameter μ (or μ') on robust-04 (short) and trec-9/10 (short) respectively

Table XI reinforces this observation. Table XI shows the characteristics of the average length of documents in the collections and the value of m_c that is estimated on each collection. We can also see that m_c is correlated with the lengths of documents in the collections. Furthermore, in the same table we can see that close to the optimal effectiveness is possible by setting $\omega = 0.8$ for SPUD$_{est_{\mu'}}$. This is because m_c is essentially performing the tuning on a per collection basis. The parameter m_c has a very intuitive interpretation as the initial mass of the background Pólya urn.

Table XI. MAP comparison of SPUD$_{dir}$ model for well-tuned μ' and SPUD$_{est_{\mu'}}$ which uses an automatically estimated value of μ'

	robust-04	trec-8	trec-9/10	ohsu	gov2	mq-07	mq-08
$|\vec{d}|_{avg}$	162	242	157	68	181	181	181
$|d|_{avg}$	265	558	344	104	529	529	529
$\hat{m_c}$	258	421	326	112	234	234	234
$\hat{u'} = 4 \cdot \hat{m_c}$	1034	1688	1308	448	936	936	936
	short queries						
SPUD$_{dir}$	0.252	0.319	0.200	n/a	0.314	0.431	0.445
SPUD$_{est_{\mu'}}$	0.249	0.320	0.199	n/a	0.314	0.429	0.443
	medium queries						
SPUD$_{dir}$	0.289	0.347	0.247	0.270	0.332	n/a	n/a
SPUD$_{est_{\mu'}}$	0.287	0.344	0.246	0.270	0.329	n/a	n/a
	long queries						
SPUD$_{dir}$	0.296	0.314	0.254	n/a	0.323	n/a	n/a
SPUD$_{est_{\mu'}}$	0.295	0.307	0.255	n/a	0.322	n/a	n/a

6.5. Analysis of Retrieval Model Aspects

In this third experiment we aim to evaluate the retrieval effectiveness of the new background model (i.e. α_c) and the new smoothing methods in the SPUD model separately in a piece-wise fashion. We gradually adapt parts of the multinomial query likelihood functions until the SPUD retrieval functions are comprised. The experiment pinpoints the parts of the SPUD retrieval functions that lead to changes in retrieval effectiveness. This piece-wise adaptation provides evidence that the individual retrieval intuitions outlined in Section 5.2 are valid. Furthermore, we conduct an analysis of the retrieval characteristics of the best performing methods.

6.5.1. Results of the Analysis of Retrieval Model Aspects. Table XII, which also contains a column for a *hybrid* model, outlines the parameter values for the functions used in this experiment. Essentially, this *hybrid* retrieval function differs from the SPUD retrieval functions only in the fact that it uses different parameter estimates for λ_{jm} and m_d that effect the smoothing for SPUD$_{jm}$ and SPUD$_{dir}$ respectively. The changes to these parameter estimates makes the *hybrid* model closer to the multinomial retrieval functions. The only difference between the MQL and *hybrid* model is that *hybrid* uses the expected multinomial of the background DCM (i.e. θ'_c in Eq. 8) as its background model.

Table XII. Decomposition of Retrieval Functions

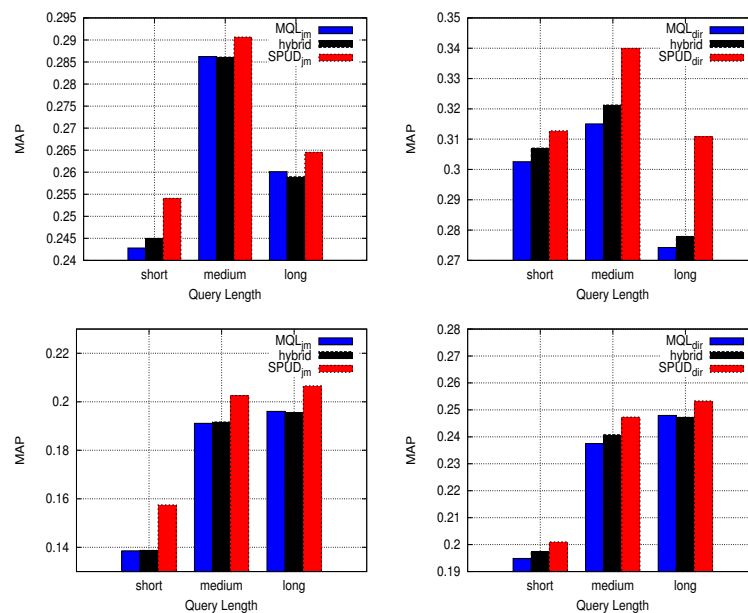| Smoothing	Colour	MQL Multinomial	*hybrid* DCM	SPUD DCM
Jelinek-Mercer (jm)	Blue	$\pi_{jm} = 0.2$	$\lambda_{jm} = 0.2$	$\lambda_{jm} = |\vec{d}|/|d|$
Dirichlet (dir)	Red	$\mu = 2000$	$\mu' = 2000, m_d = |d|$	$\mu' = 2000, m_d = |\vec{d}|$

Fig. 10. Analysis of performance gains from different parts of the SPUD retrieval models for trec-8 (top) and trec-9/10 (bottom) for short, medium, and long queries. Models that use Jelinek-Mercer smoothing are on the left-hand, side while those that use Dirichlet smoothing are on the right-hand side.

Fig. 10 shows the effectiveness of the functions that use Jelinek-Mercer smoothing (left-hand side) and those that use a type of Dirichlet smoothing (right-hand side) on two test collections. In general, the use of the new background DCM model aids retrieval as we can see an increase in effectiveness for the *hybrid* (in black) retrieval functions over the MQL functions (in blue). We note that the magnitude of the difference is small, and that in some cases the performance decreases slightly. In general, the introduction of a document boundary into the estimation of the background language model is more effective for SPUD$_{dir}$ than for SPUD$_{jm}$.

However, the different smoothing techniques introduced in the SPUD model yield a greater increase in performance i.e. when comparing the SPUD function (in red) to the $hybrid$ function (in black). The smoothing in the SPUD model, amongst other factors, affects document length normalisation and improves retrieval effectiveness substantially. The results in Fig. 10 demonstrate[14] that the new retrieval characteristics brought about by both the background DCM and the document DCM positively influence retrieval effectiveness. The results of these experiments further validate the use of the DCM as a more plausible document model than the multinomial. This is because the changes to the query-likelihood retrieval method that the new background and new document model bring about, increase the retrieval effectiveness for $SPUD_{dir}$ method over MQL.

Previously in Section 5.2.2 we analysed the lengths of documents retrieved in the top 1000 documents by both MQL_{dir} and $SPUD_{dir}$. We found that $SPUD_{dir}$ was more likely to retrieve longer documents. We now look at the length characteristics of the top 20 documents returned per query by both $SPUD_{dir}$ and MQL_{dir} to determine if the differences in length are correlated with increased performance in terms of NDCG@20. Firstly Fig.11 confirms that on average $SPUD_{dir}$ retrieves documents with a longer vector length than MQL_{dir} in the top 20. Table XIII shows the correlation between the differences in average length and the differences in NDCG in the top 20 documents across a number of representative test collections. We report a small but insignificant correlation between the increase in average vector length and query effectiveness (as measured by NDCG@20). Although this correlation analysis is somewhat inconclusive, we can confirm that on average $SPUD_{dir}$ retrieves documents with a longer vector length (i.e. greater number of distinct terms), and that the overall evidence seems to suggest that this is leading to increase effectiveness.

Fig. 11. Difference in average vector length of the top 20 returned documents for $SPUD_{dir}$ and MQL_{dir} on trec-8 (left) and gov2 (right) web collections for short queries.

Table XIII. Linear correlation of Δ average document length in top 20 and Δ NDCG@20 over short queries sets for Web collections

avg_len	trec-8	trec-9/10	gov2
$|d|$	0.0525	0.0462	-0.0161
$|\vec{d}|$	0.0838	0.0622	0.0216

---

[14]Results on other test collections used in this work are consistent with those reported in Fig. 10.

6.6. Pseudo-Relevance Feedback

Finally, we evaluate the SPUD model in a pseudo-relevance feedback setting. Pseudo-relevance feedback is a useful approach for expanding short queries when the user has not entered a sufficiently long query. In essence, the pseudo-relevance model is responsible for the selection and weighting of candidate expansion query-terms from the top k documents of an initial retrieval run. We adapt the state-of-the-art RM3 [Abdul-jaleel et al. 2004; Diaz and Metzler 2006] approach to select and weight terms according to the SPUD_{dir} retrieval approach. The pseudo-relevance model based on SPUD_{dir} is estimated from an initial ranking as follows:

$$p(t|q_e) = \sum_{\boldsymbol{\alpha}_{dm} \in \hat{R}_\alpha} p(t|\boldsymbol{\alpha}_{dm}) \frac{p(q|\mathcal{M}_d = \mathbb{E}[\boldsymbol{\theta}_{dm}|\boldsymbol{\alpha}_{dm}])}{\sum_{\boldsymbol{\alpha'}_{dm} \in \hat{R}_\alpha} p(q|\mathcal{M}_d = \mathbb{E}[\boldsymbol{\theta}_{dm}|\boldsymbol{\alpha'}_{dm}])} \quad (21)$$

where $\hat{R}_\alpha$ is the set of pseudo-relevant document models (i.e. it is the top k document models from an initial retrieval run). If we replace $p(t|\boldsymbol{\alpha}_{dm})$ with $p(t|\boldsymbol{\theta}_{dm})$ and $p(q|\mathcal{M}_d = \mathbb{E}[\boldsymbol{\theta}_{dm}|\boldsymbol{\alpha}_{dm}])$ with $p(q|\boldsymbol{\theta}_{dm})$ in Eq. (21), we recover RM3. The final query model is then estimated by linearly smoothing this estimated relevance model $p(t|q_e)$ with the original query as follows:

$$p(t|q') = \tau \cdot p(t|q) + (1 - \tau) \cdot p(t|q_e) \quad (22)$$

where τ controls the weight of the initial query. The new query model is then used to query the corpus using the initial retrieval method (i.e. SPUD_{dir}). We set the number of pseudo-relevant documents $k = 20$ and generate a pseudo-relevance model of 50 terms. We smooth the pseudo-relevance model with the original query model by setting $\tau = 0.5$. The parameter u' (and u in MQL_{dir}) is set to 2000 during ranking and is set to 0 only during the expansion step. These expansion parameters settings are set according to the literature [Abdul-jaleel et al. 2004; Lv and Zhai 2009b, 2010]. We note that the pseudo-relevance model here does not follow a DCM relevance model (i.e. we do not treat all relevant documents as being drawn from a DCM relevance model), but is simply an adaptation of the RM3 model which we refer to as **PURM**[15]. We only use short title queries in this experiment as are the types of queries to which query expansion is typically applied [Carpineto and Romano 2012].

6.6.1. Pseudo-Relevance Feedback Results. Table XIV shows the results of the pseudo-relevance feedback experiment. Firstly, we can see that when the SPUD_{dir} approach is used as the retrieval method with the RM3 expansion approach, it leads to a significant improvement over the MQL approach. This is encouraging, but hardly surprising, as the SPUD_{dir} approach has a more effective initial retrieval. However, when the retrieval method is static, and only the expansion approach is allowed to vary, the PURM approach outperforms the RM3 approach. The absolute increase in effectiveness when using the new PURM expansion approach is quite low, but nevertheless is significant on trec-8 and gov2. This low increase in effectiveness is to be expected as the only difference between the RM3 expansion approach and the PURM approach (when u and u' are set to 0) is that the PURM approach uses the SPUD retrieval score to weight terms, while the RM3 approach uses the MQL retrieval score. Overall, while this validates that the SPUD_{dir} document retrieval score is useful in the expansion step of

---

[15]Essentially, the PURM expansion model with $u' = 0$ only differs from RM3 with $u = 0$ in the fact that the document retrieval score used to weight the expansion term is different. Therefore, we would expect only a small difference in effectiveness.

pseudo-relevance expansion approaches, the main increase in effectiveness comes from the better ranking of SPUD_{dir} compared to MQL_{dir}.

A point worth noting is that the performance of the feedback approaches on the mq-07 and mq-08 test collections are worse than for the initial retrieval run (no expansion). It has been reported that pseudo-relevance feedback varies depending on the type and quality of the test collection with results showing little or no improvement when using parts of the million query track data (i.e. mq-07 and mq-08) [Meij 2010]. One possible reason for this is that during the creation of the mq-07 and mq-08 test collections a shallow pool depth was used in order to judge more queries than is usual for trec collections. As pseudo-relevance feedback tends to increase average precision by increasing recall, the lower number of judged documents for the million query track collections could affect the natural behaviour of query expansion approaches on this collection.

Table XIV. MAP of pseudo-relevance feedback approaches of SPUD_{dir}-PURM, SPUD_{dir}-RM3, and MQL_{dir}-RM3 (▲ means two-sided t-test $p < 0.01$ compared to MQL_{dir}-RM3, while △ means $p < 0.05$ compared to MQL_{dir}-RM3. ● means two-sided t-test $p < 0.01$ compared to SPUD_{dir}-RM3, ○ means $p < 0.05$ compared to compared to SPUD_{dir}-RM3.)

Methods		short queries					
Ranking	Expansion	robust-04	trec-8	trec-9/10	gov2	mq-07	mq-08
MQL_{dir}	None	0.232	0.308	0.191	0.303	0.428	0.440
MQL_{dir}	RM3	0.258	0.322	0.212	0.308	0.395	0.417
SPUD_{dir}	RM3	0.265▲	0.338▲	0.218▲	0.319	0.404	0.428
SPUD_{dir}	PURM	0.266▲	0.340▲●	0.220▲	0.324 ▲○	0.408▲	0.429▲

7. DISCUSSION

In this section we discuss the main findings, limitations, and the broader impact of this work.

7.1. Comparison With Previous Work

The results of experiments in Section 6.3.1 have shown that the SPUD_{dir} method significantly outperforms the DCM-L-T of Xu and Akella [2008]. In particular, the effectiveness of DCM-L-T for longer queries, which was not presented in the original work, is particularly poor. The manner in which the initial query is used in that relevance-based model leads to a non-linear query term-frequency aspect. This is likely to affect the retrieval effectiveness for longer queries as it has been shown that the query term-frequency aspect should be close to linear [Robertson and Walker 1994].

There are several other disadvantages to the DCM-L-T method. While the complexity of most retrieval functions is linear with respect to the number of unique terms (word types) in common to both query and document, the complexity of the approach by Xu and Akella [2008] is linear with respect to the sum of the query-term frequencies (i.e. all instances of query-terms) in the document. This adversely affects retrieval time. Conversely, the SPUD model outlined in this work is as efficient at query time as the multinomial language model.

In the DCM-L-T approach, the estimation of the parameters for both the relevant and the non-relevant DCM document models do not have closed-form expressions. This is not of major concern for the estimation of a non-relevant model in a static collection[16]

---

[16]For a dynamically changing collection where new documents are discovered and indexed frequently, this may become an issue.

(which can often be estimated off-line), but is a major disadvantage for the inference of the relevant model, which must be estimated on-line at query time. In fact, one of the major difficulties with the previous relevance-based approach is estimating the set of pseudo relevant documents needed in order to infer the relevance model. Therefore, a number of computationally expensive estimation techniques are compared in order to find parameters that are the most effective in terms of retrieval. However, it was found that a manual tuning of γ is more effective than any of these estimation techniques.

7.2. Estimating Free Parameters

In Section 6.4 we have shown that both SPUD retrieval methods are more robust in terms of parameter settings than their multinomial counterparts. We have shown that for the SPUD_{dir} model, the background model is weighted approximately four times more than the document model, and that this setting (via $\omega = 0.8$) is robust across different collections. More extensive research would need to be conducted to determine if this setting is universal. Some prior research into Microblog retrieval suggests that a smaller μ parameter value in the multinomial query likelihood model is more effective on collections that contain smaller documents [Han et al. 2012; Kim et al. 2012]. This is consistent with our results (emphasised by results on the *ohsu* collection which contains short documents) as the estimate of m_c is correlated with document length (see Table XI). This provides further evidence that our free hyperparameter ω is more robust than the free parameter μ in the multinomial model.

Furthermore, although it has been suggested [Zhai and Lafferty 2004] that the parameter μ in the original multinomial language model may be affected by query length, we have found that the most effective SPUD retrieval method is robust across queries of different length. More work would need to be conducted to see if the optimal value of ω varies according to query length. Recent research [Tsagkias et al. 2011] has investigated a different generative model for queries, and this would also be an interesting future direction to explore.

The background model in SPUD is only an efficient approximation to the DCM. Although, it has been shown [Elkan 2006] that this EDCM approximation is quite accurate and has been shown to be useful for text clustering, more extensive work would need to be conducted to determine if the approximation is close to optimal in terms of retrieval effectiveness.

7.3. Theoretical Discussions

7.3.1. Term and Document Event Spaces. Aspects of both term-frequency and inverse document frequency have been at the core of many successful ranking functions over the years. The work outlined here helps to explain why both of these features have been so useful. In particular, the generative assumptions made in our document model help explain why term-frequency is such a useful and salient measure of topicality. In other words, we argue that it is because authors have preferential attachment for the content words within-documents that term-frequency is such useful measure of topicality. Furthermore, these generative assumptions lead to power-law characteristics of term-frequency in text [Simon 1955; Goldwater et al. 2011], and therefore appear to be more plausible models.

Interestingly, it is because of within-document *preferential attachment* that inverse document frequency is such an accurate measure of term-specificity. Essentially, when analysing the collection-wide characteristics of terms, for the most part we need only count the first occurrence of a term within a document, as all other occurrences depend upon this. While we did not derive idf as it appeared in its original form [Spärck-Jones 1972], our analysis shows that the best retrieval formula derived from the SPUD language model, contains characteristics closely related to that of idf (see Fig. 5). By cap-

turing burstiness in our framework we have been able to successfully combine the term event space used within each document, with the document event space used at the collection-wide level (which comes about as a close approximation to the background DCM). Others [Robertson 2004; Roelleke and Wang 2008] have argued that Harter's *eliteness* hypothesis [Harter 1975a,b], which is essentially a binary latent variable for each term, acts a *bridge* between the term space and the document space. We have found that there are alternative generative explanations for tf-idf type schemes. We believe that the SPUD language model is an important step towards developing a probabilistic generative theory explaining such schemes.

7.3.2. Relevance. We note that our retrieval model is a query-likelihood model which does not explicitly model relevance; however it is not difficult to place the same document model in a relevance framework. The KL (Kullback-Leibler) divergence, which measures the amount of information lost when one distribution is used to model another theoretical distribution, has been used in information retrieval to compare document models to query models. As this introduces the idea of a query model, it seems reasonable to imagine that this query model is a best initial approximation of the *true* relevance model (which can be updated as relevance information becomes known). Therefore, one can think about ranking documents according to the negative KL-divergence of a document model $\mathcal{M}_d$ and a true relevance model $\mathcal{M}_r$ as follows:

$$- KL(\mathcal{M}_r || \mathcal{M}_d) = - \sum_{t \in v} p(t|\mathcal{M}_r) \cdot log \frac{p(t|\mathcal{M}_r)}{p(t|\mathcal{M}_d)} \tag{23}$$

It is also well-known that the query-likelihood function is rank equivalent to the KL-divergence between a query model and document model as a special case [Zhai and Lafferty 2001b; Zhai 2008]. The above equation is rank equivalent to the SPUD retrieval functions when $p(t|\mathcal{M}_r)$ is estimated using $c(t,q)/|q|$ and when $p(t|\mathcal{M}_d)$ is estimated using the new document models presented in this article (i.e. α_{dm}).

7.3.3. Document Length Normalisation. In Section 5.2.1 we defined a constraint to capture the verbosity hypothesis. We have shown that the best performing SPUD retrieval method adheres to this constraint. We have seen that in general the multinomial model MQL_{dir} over-penalises long documents and the SPUD_{dir} model is more likely to retrieve longer documents (See Fig. 4). This is because the multinomial model does not model the distinction between word-types and word-tokens, and ultimately over-penalises documents with recurrences of non-query terms. This result builds on recent research [Lv and Zhai 2011; Cummins and O'Riordan 2012] that developed further constraints regarding document length normalisation. It would be interesting future research to determine if the SPUD_{dir} function adheres to these constraints also.

The SPUD model significantly outperforms a highly tuned multinomial model (MQL) for all query lengths. This is because the SPUD model incorporates two types of document length normalisation. One aspect of normalisation (verbosity) regulates the term-frequency with respect to the document length as longer documents (those with many word tokens) are more likely to contain higher term-frequencies. The another aspect (scope) normalises longer documents (those with more word types) as they are more likely to contain more distinct query-terms. This second aspect of normalisation is crucially dependant on query length. The SPUD model is the first model to combine these two aspects of document normalisation in a theoretically principled framework.

Interestingly, recent research has developed a two-stage document length normalisation framework [Na 2015] which incorporates both verbosity and scope normalisation into retrieval methods. It is appealing that the SPUD retrieval methods derived from our probabilistic framework contain these aspects of normalisation naturally.

7.4. Broader Impact

While we have argued that the new SPUD model addresses a number of theoretically interesting questions in IR, we have demonstrated that it also practically useful in a retrieval scenario. Given that the SPUD model is essentially a method for determining principled term-weights for document vectors, the model is likely to be useful in other areas where term-weights are used in vector representations of longer texts. This includes areas such as text classification, text clustering, and more specialised NLP tasks (e.g. keyword extraction, automatic summarisation).

7.5. Recommended Retrieval Function

The recommended retrieval function is $SPUD_{dir}$ in Eq. (17). This function has one free parameter μ' which we recommend setting to $4 \cdot m_c$, where m_c can be found by applying Newton's method to Eq. (19). Alternative μ' can be experimentally tuned on training data which is the current method of setting u in the multinomial language model.

7.6. Future Directions

The most effective SPUD method, introduced in Eq. (12), linearly combines the background DCM model with the document DCM language model. This could be extended to include more language models. For example, if we had information relating to authorship, we could estimate an author specific DCM language model that would explain textual characteristics specific to an author, as it may be the case that certain authors are generally more verbose than others. Smoothing this DCM model with both the document and background models may further improve performance. This may be particular useful in areas such as expert search.

The document model outlined in this work models word burstiness in a document specific manner. Previous work [Kwok 1996] has shown that certain terms are more bursty than others (i.e. they are more likely to repeat). This suggests that incorporating a term-specific aspect of burstiness may increase retrieval effectiveness even further. This could be modelled using a more general urn model where the level of reinforcement varies per term.

A further interesting direction is to consider integrating the document model outlined here with a model that incorporates the traditional notion of term-dependence. The details regarding such a combination have not been discussed here but would present interesting future research.

8. CONCLUSION

We have introduced a new family of language model (namely SPUD) based on a Pólya urn process. We have shown that a query likelihood retrieval method based on this model is superior to that of the state-of-the-art multinomial language model. Interestingly, we have shown that the new model can be computed as efficiently as the multinomial language model. Essentially, this means that the SPUD retrieval method can be used in place of the multinomial query likelihood method in many different retrieval applications and domains.

We have outlined a number of intuitions that help to motivate the new model. For example, we developed a constraint for the verbosity hypothesis and have shown that the most effective SPUD method, the $SPUD_{dir}$ model, adheres to this constraint. Furthermore, we have shown that the free hyperparameter (i.e. $\omega = 0.8$) in the $SPUD_{dir}$ method is robust across various collections. This essentially reduces the need for experimental tuning. Given the principled nature of the approach developed, it can be used in a variety of IR tasks. We have shown that it is useful for downstream retrieval methods, as we have used it to estimate a pseudo-relevance based model (PURM) that

demonstrates improved retrieval effectiveness on test collections when compared to a pseudo-relevance model based on the multinomial (RM3).

Future work will look to improve retrieval effectiveness by incorporating multiple DCM language models for modelling a document. Furthermore, we aim to investigate the query likelihood method using different generative assumptions for the query. In this work, we assumed a sampling-with-replacement strategy for query generation. However, different sampling strategies, such as those employed by Friedman urn's [Freedman 1965] might better model query generation.

Acknowledgements

REFERENCES

N. Abdul-jaleel, J. Allan, W. B. Croft, O. Diaz, L. Larkey, X. Li, M. D. Smucker, and C. Wade. Umass at trec 2004: Novelty and hard. In *Proceedings of TREC-04*, 2004.

G. Amati and C. J. Van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems*, 20:357–389, October 2002. ISSN 1046-8188.

M. Bendersky and W. B. Croft. Modeling higher-order term dependencies in information retrieval using query hypergraphs. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 941–950, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1472-5.

R. Blanco and C. Lioma. Graph-based term weighting for information retrieval. *Inf. Retr.*, 15(1):54–92, Feb. 2012. ISSN 1386-4564.

C. Carpineto and G. Romano. A survey of automatic query expansion in information retrieval. *ACM Comput. Surv.*, 44(1):1:1–1:50, Jan. 2012. ISSN 0360-0300.

K. W. Church and W. A. Gale. Poisson mixtures. *Natural Language Engineering*, 1: 163–190, 1995.

S. Clinchant and É. Gaussier. Retrieval constraints and word frequency distributions: a log-logistic model for IR. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, Hong Kong, China, November 2-6, 2009*, pages 1975–1978, 2009.

S. Clinchant and E. Gaussier. Information-based models for ad hoc ir. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 234–241, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0153-4.

S. Clinchant and É. Gaussier. Retrieval constraints and word frequency distributions a log-logistic model for IR. *Information Retrieval*, 14(1):5–25, 2011.

P. J. Cowans. Information retrieval using hierarchical dirichlet processes. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 564–565, New York, NY, USA, 2004. ACM. ISBN 1-58113-881-4.

R. Cummins and C. O'Riordan. An axiomatic comparison of learned term-weighting schemes in information retrieval: Clarifications and extensions. *Artificial Intelligence Review*, 28(1):51–68, June 2007. ISSN 0269-2821.

R. Cummins and C. O'Riordan. Learning in a pairwise term-term proximity framework for information retrieval. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 251–258, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-483-6.

R. Cummins and C. O'Riordan. A constraint to automatically regulate document-length normalisation. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 2443–2446. ACM, 2012.

F. Diaz and D. Metzler. Improving the estimation of relevance models using large external corpora. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 154–161, New York, NY, USA, 2006. ACM. ISBN 1-59593-369-7.

C. Elkan. Clustering documents with an exponential-family approximation of the dirichlet compound multinomial distribution. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 289–296, New York, NY, USA, 2006. ACM. ISBN 1-59593-383-2.

H. Fang and C. Zhai. An exploration of axiomatic approaches to information retrieval. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 480–487, New York, NY, USA, 2005. ACM. ISBN 1-59593-034-5.

H. Fang, T. Tao, and C. Zhai. A formal study of information retrieval heuristics. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 49–56, New York, NY, USA, 2004. ACM. ISBN 1-58113-881-4.

D. A. Freedman. Bernard friedman's urn. *The Annals of Mathematical Statistics*, 36 (3):pp. 956–970, 1965. ISSN 00034851.

J. Gao, J.-Y. Nie, G. Wu, and G. Cao. Dependence language model for information retrieval. In M. Sanderson, K. Jrvelin, J. Allan, and P. Bruza, editors, *SIGIR*, pages 170–177, New York, NY, USA, 2004. ACM. ISBN 1-58113-881-4.

S. Goldwater, T. L. Griffiths, and M. Johnson. Producing power-law distributions and damping word frequencies with two-stage language models. *Journal of Machine Learning Research*, 12:2335–2382, July 2011. ISSN 1532-4435.

Z. Han, X. Li, M. Yang, H. Qi, S. Li, and T. Zhao. Hit at trec 2012 microblog track. In *Proceedings of TREC '12*, 2012.

S. P. Harter. A probabilistic approach to automatic keyword indexing. part i. on the distribution of specialty words in a technical literature. *Journal of the American Society for Information Science*, 26(4):197–206, 1975a. ISSN 1097-4571.

S. P. Harter. A probabilistic approach to automatic keyword indexing. part ii. an algorithm for probabilistic indexing. *Journal of the American Society for Information Science*, 26(5):280–289, 1975b. ISSN 1097-4571.

D. Hiemstra. A linguistically motivated probabilistic model of information retrieval. In *Research and Advanced Technology for Digital Libraries, Second European Conference*, ECDL '98, pages 569–584, 1998.

D. Hiemstra. *Using language models for information retrieval*. Univ. Twente, 2001. ISBN 978-90-75296-05-1.

Y. Kim, R. Yeniterzi, and J. Callan. Overcoming vocabulary limitations in twitter microblogs. In *Proceedings of TREC '12*, 2012.

K. L. Kwok. A new method of weighting query terms for ad-hoc retrieval. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '96, pages 187–195, New York, NY, USA, 1996. ACM. ISBN 0-89791-792-8.

V. Lavrenko and W. B. Croft. Relevance based language models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 120–127, New York, NY, USA, 2001. ACM. ISBN 1-58113-331-6.

Y. Lv and C. Zhai. Positional language models for information retrieval. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in*

Information Retrieval, SIGIR '09, pages 299–306, New York, NY, USA, 2009a. ACM. ISBN 978-1-60558-483-6.

Y. Lv and C. Zhai. A comparative study of methods for estimating query language models with pseudo feedback. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 1895–1898, New York, NY, USA, 2009b. ACM. ISBN 978-1-60558-512-3.

Y. Lv and C. Zhai. Positional relevance model for pseudo-relevance feedback. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 579–586, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0153-4.

Y. Lv and C. Zhai. Lower-bounding term frequency normalization. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, pages 7–16. ACM, 2011. ISBN 978-1-4503-0717-8.

Y. Lv and C. Zhai. A log-logistic model-based interpretation of tf normalization of bm25. In *Proceedings of the 34th European Conference on Advances in Information Retrieval*, ECIR'12, pages 244–255, Berlin, Heidelberg, 2012. Springer-Verlag. ISBN 978-3-642-28996-5.

R. E. Madsen, D. Kauchak, and C. Elkan. Modeling word burstiness using the dirichlet distribution. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 545–552, 2005.

E. Meij. *Combining Concepts and Language Models for Information Access*. PhD thesis, University of Amsterdam, December 2010.

D. Metzler and W. B. Croft. A markov random field model for term dependencies. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 472–479, New York, NY, USA, 2005. ACM. ISBN 1-59593-034-5.

T. P. Minka. Estimating a dirichlet distribution. Technical report, Microsoft Research, 2000.

M. Mitzenmacher. A brief history of generative models for power law and lognormal distributions. *INTERNET MATHEMATICS*, 1:226–251, 2003.

S. Na, I. Kang, and J. Lee. Improving term frequency normalization for multi-topical documents and application to language modeling approaches. In *Advances in Information Retrieval, 30th European Conference on IR Research, ECIR 2008, Glasgow, UK, March 30-April 3, 2008. Proceedings*, pages 382–393, 2008.

S.-H. Na. Two-stage document length normalization for information retrieval. *ACM Transactions of Information Systems*, 33(2):8:1–8:40, Feb. 2015. ISSN 1046-8188.

J. H. Paik. A novel tf-idf weighting scheme for effective ranking. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 343–352, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2034-4.

R. Pemantle. A survey of random processes with reinforcement. *Probability Surveys*, 4:1–79, 2007.

J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 275–281, New York, NY, USA, 1998. ACM. ISBN 1-58113-015-5.

S. Robertson. Understanding inverse document frequency: On theoretical arguments for idf. *Journal of Documentation*, 60:2004, 2004.

S. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. In *Proceedings of the Third Text REtrieval Conference (TREC 1994)*, pages 109–126, 1994.

S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '94, pages 232–241, New York, NY, USA, 1994. Springer-Verlag New York, Inc. ISBN 0-387-19889-X.

T. Roelleke. IR models: foundations and relationships. In *The 35th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '12, Portland, OR, USA, August 12-16, 2012*, pages 1187–1188, 2012.

T. Roelleke and J. Wang. Tf-idf uncovered: A study of theories and probabilities. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 435–442, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-164-4.

H. A. Simon. On a class of skew distribution functions. *Biometrika*, 42(3–4):425–440, 1955.

A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '96, pages 21–29, New York, NY, USA, 1996. ACM. ISBN 0-89791-792-8.

K. Spärck-Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.

K. Spärck-Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: Development and comparative experiments. *Information Processing and Management*, 36(6):779–808, Nov. 2000. ISSN 0306-4573.

P. Sunehag. Emerge and spread models and word burstiness. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics, AISTATS*, pages 540–547, 2007.

M. Tsagkias, M. de Rijke, and W. Weerkamp. Hypergeometric language models for republished article finding. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 485–494, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0757-4.

X. Wei and W. B. Croft. Lda-based document models for ad-hoc retrieval. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 178–185, New York, NY, USA, 2006. ACM. ISBN 1-59593-369-7.

Z. Xu and R. Akella. A new probabilistic retrieval model based on the dirichlet compound multinomial distribution. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 427–434, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-164-4.

Z. Xu and R. Akella. Improving probabilistic information retrieval by modeling burstiness of words. *Information Processing and Management*, 46(2):143 – 158, 2010. ISSN 0306-4573.

C. Zhai. Statistical language models for information retrieval: A critical review. *Foundations and Trends in Information Retrieval*, 2(3):137–213, Mar. 2008. ISSN 1554-0669.

C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 334–342, New York, NY, USA, 2001a. ACM. ISBN 1-58113-331-6.

C. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the Tenth International Conference on Information and Knowledge Management*, CIKM '01, pages 403–410, New York, NY, USA, 2001b. ACM. ISBN 1-58113-436-3.

C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions of Information Systems*, 22:179–214, April 2004. ISSN 1046-8188.

J. Zhao and Y. Yun. A proximity language model for information retrieval. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 291–298, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-483-6.