Parameterized Decay Model for Information Retrieval

JIAUL H. PAIK, University of Maryland, College Park

This article proposes a term weighting scheme for measuring query-document similarity that attempts to explicitly model the dependency between separate occurrences of a term in a document. The assumption is that, if a term appears once in a document, it is more likely to appear again in the same document. Thus, as the term appears again and again, the information content of the subsequent occurrences decreases gradually, since they are more predictable. We introduce a parameterized decay function to model this assumption, where the initial contribution of the term can be determined using any reasonable term discrimination factor. The effectiveness of the proposed model is evaluated on a number of recent web test collections of varying nature. The experimental results show that the proposed model significantly outperforms a number of well known retrieval models including a recently proposed strong Term Frequency and Inverse Document Frequency (TF-IDF) model.

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

General Terms: Algorithm, Experimentation, Performance

Additional Key Words and Phrases: Document ranking, retrieval model, term weighting

ACM Reference Format:
Jiaul H. Paik. 2016. Parameterized decay model for information retrieval. ACM Trans. Intell. Syst. Technol. 7, 3, Article 27 (February 2016), 21 pages.
DOI: http://dx.doi.org/10.1145/2800794

1. INTRODUCTION

Given a natural language query, a document is considered potentially relevant if it contains the query terms (one or more). In the case of large document collections, the resulting number of documents matching the query terms can far exceed the number a human user could possibly sift through. Accordingly, it is essential for a search engine to rank-order the documents matching a query. To do this, the search engine computes, for each matching document, a score with respect to the query at hand. Thus, the success of a text retrieval system, in satisfying the user's information need, crucially depends on the underlying term weighting function.

Term frequency and inverse document frequency are the two key variables that play a pivotal role in determining a term's importance in a document. Moreover, term frequency is not a fully independent variable, since it is often positively corelated with the length of a document. Thus, term frequencies are regularized with respect to the document length to reduce the advantage the longer documents may get over the shorter documents. The main goal of a weighting model is to combine these variables to produce a score of a term in a document. Different models attempt to combine these

This research was supported in part by DARPA contract HR0011-12-C-0015 and NSF award 1065250.
Authors' addresses: J. H. Paik, UMIACS, University of Maryland, College Park, 20740; email: jia.paik@gmail.com.

DOI: http://dx.doi.org/10.1145/2800794

variables differently. For example, tf.idf models take the product of the *tf* factor and the *idf* factor, where the *tf* factor is some combination of *tf* and the document length. BM25 [Robertson and Walker 1994], the most successful classical probabilistic model, measures term weight by approximating the two Poisson models, while the Language Models (LM) [Ponte and Croft 1998] rank documents based on their query likelihood scores. On the other hand, the Divergence From Randomness (DFR) models [Amati and Van Rijsbergen 2002] compute term weight by measuring the difference between a term distribution produced by a random process and the actual term distribution.

In the probabilistic models (LM, DFR), the structure of the term weighting functions are determined theoretically. However, this is not the case for most of the state of the art *TF.IDF* functions. Often, the construction of the functions is guided by observations. Although the *idf* function does not vary much from one model to another, there are a large number of variations in the *tf* functions. These variations can mostly be attributed to the degree of influence the term frequency factor can make in determining relevance of a document. The initial *TF.IDF* model started with a linear tf function. It was later realized that the degree of relevance is very unlikely to be directly proportional to the frequency of a term. As a consequence, linear *tf* functions tend to favor the documents that contain a single term many times. However, it is observed that the number of distinct matches is more important than the frequent occurrence of a subset of the query words. Thus, to overcome the limitations of linear *tf* functions, subsequent *TF.IDF* models empirically choose logarithmic damping functions, such as $1 + \log(tf)$ [Salton and Buckley 1988] and $1 + \log(1 + (\log(tf)))$ [Singhal 2001]. BM25, the classical probabilistic model, uses an asymptotically upper bounded increasing *tf* function to achieve the same effect, however, unlike *TF.IDF* models, BM25's *tf* function is not entirely empirical, since it is an approximation of the 2-Poisson model. Very recently, in Paik [2013], a similar empirical damping function ($x/(1 + x)$, x is normalized *tf*) is used to transform the length normalized *tf* values.

The principal objective of this article is to introduce a weighting model that goes beyond the empirical way of choosing a function, unlike existing *TF.IDF* models [Singhal et al. 1996; Paik 2013]. The proposed approach attempts to factor in the phenomenon that words in a document tend to appear in bursts: if a word appears once, it is more likely to appear again [Madsen et al. 2005]. The key hypothesis is that the weight of a term at a particular instance of term frequency decreases at a rate that depends on the weight at that particular point, where the initial weight of the term can be determined using any reasonable term discrimination function (such as standard *idf*). We model the hypothesis by a parameterized decay function. Intuitively, the hypothesis models an instance-based term weighting, where the contribution of an instance of a term is punished depending upon the extent of the previous occurrence of the term in the document. The resultant weight of the term in a document is then measured by taking the cumulative weight of each instances. Term frequency is normalized using the two-aspect *tf* normalization scheme proposed in Paik [2013].

The effectiveness of the proposed model is evaluated on a number of recent test collections containing homogeneous (GOV2) as well as heterogeneous (ClueWeb) web data. We compare the performance of the proposed method with the state-of-the-art representative baselines from the *TF.IDF* model, classical probabilistic model, language model, and divergence from randomness model. Our primary experimental results show that the proposed model almost always outperforms the preceding baselines by a significantly large margin. Moreover, the results demonstrate that the proposed model is more precise than the state-of-the-art models, thereby making it a potential choice for web search.

We organize the article as follows. Section 2 presents the motivation, followed by the derivation of the model, the term frequency normalization schemes, and the term

discrimination functions. Section 3 reviews the state of the art. The Test collections, the baselines, and the other experimental setup is detailed in Section 4. In Section 5, we present the experimental results. The impact of a particular term discrimination function is explored in Section 5.5. Finally, we conclude in Section 6.

2. PROPOSED WORK

2.1. Motivation

Although *TF.IDF* weighting function is known to be an empirical term weighting model, Amati and Van Rijsbergen [2002] show that the design of the function can be grounded in information theoretic principle. To illustrate this point, consider the following setup. Let the document frequency of a term t be $df(t)$ and a document d contains the term t, $c(t, d)$ times. Assuming that the observed event (t occurs $c(t, d)$ times in document d) has been governed by a random process, Amati and Van Rijsbergen [2002] justify that the information gain of the term t in d can be measured by

$$-\log(P(c(t, d)|d)),$$

where $P(c(t, d)|d)$ denotes the probability that the term t appears $c(t, d)$ times in d. We can estimate $P(c(t, d)|d)$ in two different ways. In one approach, we can estimate $P(c(t, d)|d)$ by modeling the separate occurrences of the term t as a sequence of random events, rather than treating the frequency as a single atomic quantity. Specifically, if we assume that the separate instances of the term t are independent, we can estimate $P(c(t, d)|d)$ as follows:

$$P(c(t, d)|d) = P(t \in d)^{c(t,d)} = \prod_{i=1}^{c(t,d)} P(t \in d). \tag{1}$$

The quantity $P(t \in d)$ is the probability that a randomly chosen document will contain the term t and its estimation is the central issue in the preceding weighting framework. In principle, $P(t \in d)$ can be estimated using any standard distribution function, such as Binomial or Poisson distribution (see Section 2.4 for details), however, not all functions are found to be equally effective (see Section 5). Amati and Van Rijsbergen [2002] describe a mechanism to estimate $P(t \in d)$ using a Bayesian approach, which is measured as

$$P(t \in d) = \frac{df(t) + 0.5}{N + 1}, \tag{2}$$

where N and $df(t)$ denote the number of documents in the collection and the document frequency of the term t, respectively. Consequently, $P(c(t, d)|d)$ is estimated as

$$P(c(t, d)|d) = \left(\frac{df(t) + 0.5}{N + 1}\right)^{c(t,d)}. \tag{3}$$

Therefore, the information content of the $c(t, d)$ occurrences of t in d is computed as

$$-\log(P(c(t, d)|d)) = c(t, d) \cdot \log\left(\frac{N + 1}{df(t) + 0.5}\right). \tag{4}$$

Clearly, the preceding function turns out to be the classical *TF.IDF* weighting scheme, where the *tf* factor is a linear function of within document term frequency. In the other approach, one can directly model the probability of the observed term frequency ($c(t, d)$). Amati and Van Rijsbergen [2002] introduce a number of weighting functions that explicitly model this quantity using various well known distributions (such as

Binomial, Poisson, Bose-Einstein statistics). As a concrete example, $P(c(t,d)|d)$ can be estimated using the Poisson distribution as

$$P(c(t,d)|d) = e^{-\mu} \cdot \frac{\mu^{c(t,d)}}{c(t,d)!}. \tag{5}$$

Consequently, the information content, $-\log(P(c(t,d)|d))$ is measured as

$$-\log(P(c(t,d)|d)) = \log(c(t,d)!) + \mu - c(t,d) \cdot \log(\mu). \tag{6}$$

The mean of the Poisson distribution, μ is defined as $\frac{ctf(t)}{N}$, where $ctf(t)$ is the collection frequency of t.

One potential deficiency with the preceding models is that the separate occurrences of the same term in a document are considered independent and thus they are allowed to contribute equally. Amati and Van Rijsbergen [2002] argued that once a term appears in a document, it increases the likelihood of further occurrence of the same term in the same document (this phenomenon is known as the "apparent aftereffect of future sampling" [Amati and Van Rijsbergen 2002; Feller 2008]). Since a more probable event has less information content (measured by $-\log(p)$), as the term appears repeatedly, the information content of the subsequent occurrences of the terms should decrease gradually [Cummins et al. 2010].

Several attempts have been made to address this potential limitation. The most standard approach measures the weight by taking the product of the length normalized *tf* factor and the *idf* factor (which measures the information content of single occurrence). The *idf* factor does not vary much from one model to another. However, a large number of *tf* functions have been employed. The main focus of these *tf* functions has been to downplay the contribution of *tf* that in effect takes into account the dependency assumption. In the earlier *TF.IDF* models, the logarithmic *tf* function $(1 + \log(tf))$ was used [Salton and Buckley 1988]. Singhal [2001] observed that the logarithmic *tf* function is not effective enough and recommended the use of an even more slow growing function $(1 + \log(1 + \log(tf)))$. However, the choice of most of the functions is very empirical in nature. In that respect, Robertson and Walker [1994] mark a significant departure from the empirical ways of choosing the *tf* function and propose an asymptotically bounded *tf* function that approximates the 2-Poisson probabilistic model. Recently, Paik [2013] proposes the multiaspect *TF.IDF* model where the function $x/(1 + x)$ is used to transform the normalized *tf* values.

All in all, most well known *TF.IDF* models are based on the product of the term frequency and the term discrimination factors. As we discussed, the occurrence of a particular instance of a term in a document tends to be influenced by its prior occurrence in the same document. In order to address the phenomenon, they primarily focus on the design of the *tf* function. However, none of the weighting schemes models this dependency explicitly. This article makes an attempt to explicitly model this hypothesis using a parameterized decay function. Unlike existing *TF.IDF* models, the proposed model does not integrate the *tf* and the *idf* factors separately by assuming that the final weight is a product of these two factors. The resultant weighting function is a direct consequence of the underlying decay hypothesis.

2.2. Model Derivation

What is evident from the preceding discussion is that if $t^1, t^2, t^3, \ldots, t^n$ are the instances of the term t in the document d, then the following condition holds.

$$P(t^1) < P(t^2|t^1) < \cdots < P(t^n|t^{n-1}). \tag{7}$$

Consequently, if $f_t(i)$ is our desired function that measures the contribution of the i-th instance of term t, it should satisfy the following condition (Equation (8)).

$$f_t(i+1) < f_t(i), \text{ for } 1 \leq i \leq n-1. \tag{8}$$

Clearly, Equation (8) suggests that the function $f_t(i)$ should decrease monotonically. Thus, our next major goal is to figure out the rate at which $f_t(x)$ drops off. The crudest assumption might be that the function $f_t(x)$ decreases at a constant rate. However, this assumption ignores the hidden connection between the probability of occurrence of a term and its specificity in the collection. Recent work [Lv and Zhai 2011a] has shown that a *tf* function that factors in the distribution of a term in the collection is a better choice as opposed to the distribution-independent functions and thus the preceding assumption has potential limitations. To illustrate the point, consider the following example. Let t_1 and t_2 be two terms such that $df(t_1) > df(t_2)$ (i.e., t_1 is more general than t_2 in the retrieval corpus). If each of t_1 and t_2 occur in a document two times (let the instances of t_1 be t_1^1, t_1^2 and the instances of t_2 be t_2^1, t_2^2), then the probability that t_1^2 is coming from the same topic as t_1^1, is smaller than the probability that t_2^2 is coming from the same topic as t_2^1. The reason is that the more general terms are used in many different contexts and therefore, they are part of many uncorrelated topics. Thus, the probability of topical independence of the two separate instances of a more general term should be higher than the same for the rare terms. Therefore, the function $f_t(i)$ should decrease slower (faster) for the more general terms (rarer terms) (henceforth we call it *decay hypothesis*). We model this hypothesis by the following equation:

$$\frac{\partial f_t(x)}{\partial x} = -\lambda \cdot (f_t(x))^m, \tag{9}$$

where $\lambda(> 0)$ is the parameter that controls the rate of the decay, and m is the shape parameter that determines the nature of the decay. Equation (9) is not completely satisfactory in the sense that it only represents an instance of an innumerable number of possible functions that can be constructed from the *decay hypothesis*. The author makes no attempt to give fully rigorous justification for the choice of Equation (9). In fact, the author believes, determining the precise nature of the decay is beyond his abilities. Hence, Equation (9) is an attempt to approximate the *decay hypothesis* where the parameterized function, $f_t(x)^m$, plays the key role in achieving that goal. In particular, the parameter m plays the pivotal role in modeling the nature and the degree of the decay (and hence, dependency). A near zero value of m implies that $f_t(x)$ decreases at a constant rate (λ). For a large positive value of m, $f_t(x)$ drops very sharply. In other words, the rate of descent of $f_t(x)$ is higher for larger value of m. On the other hand, if $m \approx 0$, then λ close to zero means that the descent of $f_t(x)$ is very marginal, which in turn gives a linear *tf*-like effect (all instances are equal contributors). Thus, by varying the values of the parameters we can get various types of decay function.

In order to get the functional form of $f_t(x)$, we now need to solve the differential Equation (9). However, the differential equation does not produce a unique functional form of $f_t(x)$ for all values of m. The power rule (i.e., x^n) of integration is applicable only if $m \neq 1$, while we need to apply the reciprocal rule for $m = 1$. Next, we handle these two cases separately.

Case 1: ($m \neq 1$) From Equation (9), we have

$$\frac{1}{(f_t(x))^m} df_t(x) = -\lambda \cdot dx. \tag{10}$$

The differential Equation (10) has the following solution:

$$\frac{1}{m-1} f_t(x)^{1-m} = -\lambda x + c, \tag{11}$$

where c is a constant associated with each term. The value of c can be determined from the initial condition as follows. When for the first time a term t is seen in a document, the amount of the contribution is the information gain due to the occurrence of term t (such as *idf*). Therefore, a natural initial condition could be

$$f_t(x) = f_t^0 \text{ for } x = 1. \tag{12}$$

However, once term frequency is normalized in accordance with the document length, we do not have the liberty to use condition (12), since normalized term frequency may be less than 1 even if a document contains the term more than once. Therefore, we need to set an initial value that should be consistent with the possible range of the term frequency values. Since we are dealing with the normalized term frequency values (i.e., *tf*'s are regularized under the same document length scale), we assume a nonzero common initial term frequency value (for each term in the collection).

Thus, we use the following initial condition:

$$f_t(x) = f_t^0 \text{ for } x = x_0, \text{ where } x_0 \to 0+. \tag{13}$$

Putting this condition into Equation (11), we have

$$c = \lim_{x_0 \to 0+} \lambda x_0 + \frac{1}{1-m} (f_t^0)^{1-m} \tag{14}$$

$$= \frac{1}{1-m} (f_t^0)^{1-m}. \tag{15}$$

Putting the value of c into Equation (11) we have

$$(f_t(x))^{1-m} = -\lambda(1-m)x + (f_t^0)^{1-m}. \tag{16}$$

Hence, from Equation (16) we have

$$f_t(x) = (-\lambda(1-m)x + (f_t^0)^{1-m})^{\frac{1}{1-m}}. \tag{17}$$

The last equation measures the amount of contribution of the term t at frequency x. We now use the formula to compute the total contribution of the term t in document d having frequency x.

If we consider a somewhat simplistic scenario where all the documents in a collection have the same length, we can simply compute the total contribution of a term in a document by

$$F_t(x) = \sum_{i=1}^{x} f_t(i). \tag{18}$$

However, any real test collection contains documents of varying lengths and hence, the term frequency needs to be regularized with respect to the document length for ultimate retrieval effectiveness. Clearly, the normalized term frequencies do not lie on the discrete natural number space; instead they take continuous real values. Thus, Equation (18) cannot be applied straightaway. Instead, the total contribution of a term is computed by the following integral.

$$F_t(x) = \int_a^x f_t(x)\, dx, \tag{19}$$

where a is the lower bound of the term frequency. Furthermore, since the normalized term frequencies are continuous in $(0, x]$ (that is, the term frequency values are always positive), we can rewrite the preceding equation as

$$F_t(x) = \lim_{a \to 0+} \int_a^x f_t(x)\, dx \tag{20}$$

$$= \lim_{a \to 0+} \int_a^x (-\lambda(1-m)x + (f_t^0)^{1-m})^{\frac{1}{1-m}}\, dx. \tag{21}$$

Once again, Equation (21) can be solved using the algebraic rule of integration if $\frac{1}{1-m} \neq -1 \Rightarrow m \neq 2$, while we apply a reciprocal rule for $m = 2$. Hence, Equation (21) has the following solution (see Appendix for detailed derivation):

$$F_t(x) = \begin{cases} \frac{1}{\lambda(2-m)}((f_t^0)^{2-m}) - z^{\frac{2-m}{1-m}}), & \text{if } m \neq 2 \\ \frac{1}{\lambda}(\log(f_t^0) - \frac{1}{(1-m)}\log(z)), & \text{otherwise,} \end{cases} \tag{22}$$

where $z = -\lambda(1-m)x + (f_t^0)^{1-m}$.

Case 2: ($m = 1$) Now, setting $m = 1$ in Equation (9) we have

$$\frac{\partial f_t(x)}{\partial x} = -\lambda \cdot f_t(x). \tag{23}$$

The differential Equation (23) has the following solution:

$$ln(f_t(x)) = -\lambda x + k. \tag{24}$$

Reorganizing the preceding equation we have

$$f_t(x) = c \cdot e^{-\lambda x}, \tag{25}$$

where c is a constant associated with each term. We use the same initial condition (Equation (13)) to compute the value of c as follows:

$$c = \lim_{x_0 \to 0+} f_t^0 \cdot e^{\lambda x_0} = f_t^0. \tag{26}$$

Putting the value of c into Equation (25) we have

$$f_t(x) = f_t^0 \cdot e^{-\lambda x}. \tag{27}$$

The total contribution of a term t is thus

$$F_t(x) = \lim_{a \to 0+} \int_a^x f_t(x)\, dx \tag{28}$$

$$= \lim_{a \to 0+} f_t^0 \int_a^x e^{-\lambda x}\, dx \tag{29}$$

$$= \frac{1}{\lambda} f_t^0 (1 - e^{-\lambda x}). \tag{30}$$

Summary. In summary, if x is the normalized term frequency of a term t in document d, with f_t^0 being the information content of its single occurrence, the weight of the term in d is computed by the function given next, where $z = -\lambda(1-m)x + (f_t^0)^{1-m}$.

$$F_t(x) = \begin{cases} \frac{1}{\lambda(2-m)}((f_t^0)^{2-m}) - z^{\frac{2-m}{1-m}}), & \text{if } m \neq 1 \text{ and } m \neq 2 \\ \frac{1}{\lambda}(\log(f_t^0) - \frac{1}{(1-m)}\log(z)), & \text{if } m \neq 1 \text{ and } m = 2 \\ \frac{1}{\lambda} f_t^0 (1 - e^{-\lambda x}) & \text{if } m = 1. \end{cases} \tag{31}$$

In Section 2.3, we give the details on the term frequency normalization and how to combine the two term frequency normalizations for final weight. The schemes for measuring the information content of a term (f_t^0) are discussed in Section 2.4.

The exponential decay model (that is a special case of our model) has been extensively used in many branches of natural science (fluid dynamics, radioactivity), social science (finance), and computer science (routing protocol). Indeed, in information retrieval, Zhang et al. [2009] use exponential decay kernel to predict what term is going to recur next in an online stream, given the term occurrence history. They propose a time sensitive language modeling approach that uses the term frequency and the term recency information. However, the model does not address the issue of document length and is thus difficult to apply in a full text retrieval system. To the best of our knowledge, we are not aware of any other work that uses a parameterized decay model to derive term weighting function for full text retrieval.

2.3. Term Frequency Normalization

Raw term frequencies are known to be less effective because of its correlation with the document length. Thus, a long document enjoys preference over a short document if the term frequency is used as is. Such phenomenon enforces the need for term frequency normalization in accordance with the document length. There are two major reasons behind the necessity of term frequency normalization, namely, the scope and the verbosity hypothesis [Robertson and Walker 1994].

(1) Scope Hypothesis: A document may get longer if it contains many unrelated contents together. Therefore, although the frequency of a term may not increase in this case, the document uses many distinct terms. Since the chance of a random match of a term between a query and a document is approximately proportional to the number of distinct terms in the document, the longer documents get an additional advantage over the shorter documents.
(2) Verbosity Hypothesis: This is the opposite case of scope hypothesis, where the documents get longer because of their repeated use of the same content. Thus, the term frequencies become higher without giving any additional useful information.

One of the well known approaches to address this issue is to regularize the term frequency in accordance with the document length. A successful approach for doing this is to compare the length of the concerned document to the length of an ideal document (pivotal document). Both pivoted TF.IDF and BM25 effectively use this strategy where the length of the pivotal document is the average document length of the retrieval collection. Thus, the *tf*'s of an average length document remain unchanged, while the *tf*'s of the documents longer (shorter) than the average length document are punished (rewarded).

Recently, Paik [2013] argued that the traditional length based normalization alone is not sufficient to capture the different aspects of term importance and proposed two

normalization formulas: one is based on the within document average term frequency, while the other makes use of the traditional length based approach. These two normalized *tf*'s are then combined. We use the same normalization schemes as described in Paik [2013], since it gives state-of-the-art results. For convenience, the normalization factors are called $nf1(t, d)$ (normalized frequency 1 of term t in the document d) and $nf2(t, d)$ (normalized frequency 2 of term t in the document d). The following equations formally define the normalization schemes.

$$nf1(t, d) = \frac{\log(1 + tf(d))}{\log(\delta + mtf(d))}, \tag{32}$$

$$nf2(t, d) = tf(t, d) \log\left(1 + \frac{adl}{l(d)}\right). \tag{33}$$

The terms mtf, adl, and $l(d)$ denote the mean term frequency of the document that contains t, the average document length of the collection, and the length of the document d. We introduce a smoothing parameter δ in Equation (32).

The weight of the term t is finally computed by the following equation where we give equal weight to each of the *tf* components.

$$W(t) = 0.5 \cdot F_t(nf1(t, d)) + 0.5 \cdot F_t(nf2(t, d)). \tag{34}$$

2.4. Determining the Contribution of a Single Term (f_t^0)

We reiterate that the quantity f_0 gives us the amount of contribution made by a single occurrence of a term t, and hence it is reasonable to assume that the quantity can be any effective term discrimination measure. Various term discrimination functions are used in Information Retrieval (IR) experiments. We integrate the following term discrimination functions into our retrieval model. Our main objective is to investigate the impact of various term discrimination functions and to determine which one fits best into the proposed model.

(1) Standard *idf*: The usual *idf* used widely in most retrieval models. This *idf* can be correlated with the information content of the event that a randomly chosen document contains a term t having document frequency df. It is defined as

$$-\log(P(t \in d)) = \log\left(\frac{N}{df}\right). \tag{35}$$

(2) BM25 idf [Robertson and Zaragoza 2009]: $\log(\frac{N-df+0.5}{df+0.5})$.

(3) Poisson idf [Church and Gale 1995]: This function is estimated using the assumption that the terms in a collection are distributed according to a Poisson process. Thus, Poisson distribution can be used to estimate the probability of at least one occurrence of a term in a document. Formally, if a document d contains a term with ctf collection frequency, then the probability of k occurrences of t in d is

$$P(k) = \frac{e^{-\mu}\mu^k}{k!}, \tag{36}$$

where μ is $\frac{ctf}{N}$. Hence, the probability of at least one occurrence of the term in the document is

$$1 - P(0) = 1 - e^{-\mu}. \tag{37}$$

Consequently, the information content of the term is $-\log(1 - e^{-\mu})$.

3. RELATED WORK

Modeling term weight is the central issue in an information retrieval system. Three widely used models in IR are the probabilistic models [Robertson 1997], the vector space model [Salton et al. 1975; Salton and McGill 1986], and the inference network based model [Turtle and Croft 1991]. A large number of instances of these models exist in the literature. In this section, we mainly focus on the review of some of the state-of-the-art representatives from different families of models.

Probabilistic models can be broadly classified into three groups, namely, the classical probabilistic model, language model, and a nonparametric divergence from randomness model. Three main factors that determine the weight of a term are (i) frequency of the term in the document, (ii) document frequency of the term in the collection (first proposed in Sparck Jones [1988]), and (iii) the length of the document that contains the term.

The key part of the probabilistic models is to estimate the probability of relevance of the documents for a query. This is where most probabilistic models differ from one another. A number of weighting formulas have been developed in this category and BM25 [Robertson and Zaragoza 2009] seems to be the most effective weighting function from among them. The BM25 model approximates the two Poisson model of relevance. The approximation is done using an increasing asymptotic *tf* function. Although, structurally, BM25 and TF.IDF functions are very similar (in the sense that they both use *tf* and *idf* factors), they differ in many respects. First, BM25 has a well grounded theory, while most of the TF.IDF models have an empirical background. Second, anatomically, the *IDF* factor of BM25 discounts the collection size by the document frequency of the term, which is different from the standard IDF factor. Third, BM25 uses a different query term frequency function, unlike TF.IDF models where that function is linear. The length normalization factor uses the average document length and a parameter has been introduced to control the relative length effect.

Probabilistic language modeling technique [Ponte and Croft 1998; Hiemstra et al. 2004] follows a different principal in estimating the relevance of a document, unlike classical probabilistic models. Typically, language modeling approaches compute the probability of generating a query from a document, assuming that the query terms are chosen independently. Unlike TF-IDF models, language modeling approaches do not explicitly use document length factor and the *idf* component. It seems that the length of the document is an integral part of this formula and that automatically takes care of the length normalization issue. However, smoothing is crucial and it has very similar effect as the parameter that controls the length normalization components in pivoted normalization or the BM25 model. Three major smoothing techniques (Dirichlet, Jelinek-Mercer, and two-stage) are commonly used in this model [Zhai and Lafferty 2004].

Amati and Van Rijsbergen [2002] proposed a nonparametric probabilistic model (called DFR) where term weights are computed by measuring the divergence between a term distribution produced by a random process and the actual term distribution. The weight computed by the DFR model is a product of two factors, defined as follows:

$$w(t, d) = -\log_2(Prob_1) \cdot (1 - Prob_2). \tag{38}$$

The left factor measures the information content of the term in a document based on its distribution in the entire collection, while the right factor measures the information gain of the term with respect to its occurrence in the elite set (set of documents that contains the term). Various standard models (such as Poisson distribution, Bose-Einstein statistics) are used to measure $Prob_1$, while Laplace's law of succession and ratio of two Bernoulli process are used to compute $Prob_2$. Like most of the well known

models, they also use the same basic components. However, the integration of various components is derived theoretically. This family of formula uses the average document length as an ideal length of the documents and the term frequencies are normalized with respect to the average document length.

In the vector space model, the search problem is viewed in a different way. Queries and documents are represented as the vector of terms. To compute a score between a document and a query the model measures the similarity between the query and document vector using cosine function. The central part of the vector space model is to determine the weight of the terms that are present in the query and the documents. Fang et al. [2011] give a comprehensive analysis of four retrieval models by defining a set of constraints that needs to be satisfied for effective retrieval. Using these constraints the strengths and weaknesses of some well known models are analyzed and some of the models are modified. There are also a number of recent works that focus on the constraint based analysis of the retrieval models [Clinchant and Gaussier 2011; Cummins and O'Riordan 2012]. Proper term frequency normalization is very crucial for the performance of $tf.idf$ models. Lv and Zhai [2011b] discover that a significant limitation of tf normalization based on document length is that they overly penalize long documents, since the well known term frequency normalization schemes are not bounded below. They propose two constraints to alleviate the weakness of the models.

Salton and Buckley [1988] summarize a number of term weighting approaches that use various types of normalization. It is evident that document length is an important component in effective term weighting. Singhal et al. [1996] identify a number of weaknesses of cosine and maximum *tf* normalization and they observe that a weighting formula that retrieves documents with chances similar to their probability of relevance performs better. Following this observation, they propose a pivoted normalization scheme that acts as a correction factor of old normalization and is one of the most effective term weighting schemes in the vector space framework. The pivoted length normalization scheme computes the term weight using the following formula. The parameter s controls the extent of normalization with respect to the document length. Typically, the term weighting functions in the vector space model are constructed empirically. Roelleke and Wang [2008] give probabilistic interpretation of TF.IDF models. Several works tried to use the data to learn the patterns that satisfy the data. For example, Greiff [1998] uses exploratory data analysis to uncover some important relationships between the document frequency and the relevance of a document.

Most of the earlier work on the vector space model normalizes the term frequency in accordance with the length of the documents. Paik [2013] argued that the length based normalization alone is not sufficient to capture the different aspects of term salience and that within document distribution of the terms plays an important role. He then proposed a two-aspect normalization scheme. An asymptotic bounded increasing function is then used to transform the normalized *tf* values. Two *tf* components are then combined using query length information. However, the main criticism is that the model is highly empirical and that is where the model proposed in this article differs with Paik [2013].

In inference network, document retrieval is modeled as an inference process [Turtle and Croft 1991]. A document instantiates a term with a certain strength and given a query the credit from multiple terms is accumulated to compute a relevance that is very equivalent to the similarity score of the vector space model. From an operational angle, the strength of instantiation of a term for a document can be considered as weight of the term in a document. The strength of instantiation of a term can be computed using any reasonable formula.

A number of works use parameterized retrieval models to estimate the weight of a term as a function of presumably effective hand constructed features from both the

Table I. Summary of the Test Collections and Topics Used in Our Experiments

Task	Collection	# Docs	Topic numbers	# Queries
Terabyte	Gov2	25,205,179	751–850	100
Web-09,10	ClueWeb-B	50,220,423	1–100	100
Web-11,12	ClueWeb-B	50,220,423	101–200	100
MQ-09	ClueWeb-B	50,220,423	20,001–57,118	684

retrieval corpus and various external information sources (such as Wikipedia, querylog). Lease et al. [2009] introduce a supervised model uniquely parameterized for each query term, where the parameters are estimated using regression rank and a set of corelated features. Zhao and Callan [2010] revisit the problem of predicting term necessity in a trainable supervised framework. They use training data from past queries to predict the necessity of new terms in previously unseen queries. Parameterized query expansion method [Bendersky et al. 2011] takes a somewhat similar approach where the weight of each query concept is determined using a parameterized combination of diverse importance features. Unlike the existing supervised ranking methods, the model learns importance weights not only for the explicit query concepts, but also for the latent concepts that are associated with the query through pseudorelevance feedback. However, all these models are supervised in nature and operate on feature rich environment. Moreover, they either focus on verbose queries or on query expansion. On the other hand, the proposed model is a general purpose unsupervised method and does not use any feature other than the basic term specific statistics (*tf* and *idf*).

4. EXPERIMENT SETUP

In this section, we describe the experiment setup for evaluating the proposed work. The objectives of our experiments are the following.

(1) To compare against a recent effective *TF.IDF* weighting scheme based on multiaspect *tf* normalization [Paik 2013]. Note that the proposed model uses the same multiaspect *tf* normalization proposed in Paik [2013]. Hence, these experiments are specifically intended to validate the merit of the proposed decay model based term weighting (Section 5.1).
(2) The effectiveness of the BM25 retrieval model can be primarily attributed to its nonlinear asymptotically bounded *tf* function. Hence, we compare the performance of the proposed scheme to the model that is based on multiaspect *tf* normalization and BM25's *tf* function (Section 5.2).
(3) To compare the performance of the proposed model with the state-of-the-art probabilistic models from different families, namely, classical probabilistic model, language model, and divergence from randomness model (Section 5.3).
(4) Finally, we investigate the impact of the three term discrimination functions (Section 5.5).

4.1. Data

We provide a summary of the test collections used for our experiments in Table I. The test collections are taken from the TREC tasks of seven recent years (2006–2012) that contain web documents. We note that the test collections vary both by size and nature. The documents of the GOV2 collection are crawled from .gov domain and thus, homogeneous in nature. On the other hand, the ClueWeb-B collection contains web pages from diverse domains. The GOV2 collection contains nearly 25 million documents, while the ClueWeb collection contains approximately 50 million web pages. There are 150 topics associated with the GOV2 collection, which were part of TREC terabyte tracks from 2006 to 2008. The ClueWeb collection was used in four TREC web tracks from 2009 to

2012. MQ-09 (million query 2009) contains 684 judged queries taken from the query log of a commercial search engine and clueweb-B data is used as the document collection. Unlike the other test collections, MQ-09 has sparse relevance judgments. Hence, for the sake of reliable conclusions the unjudged documents are removed from the ranked lists of all the models following previous work [Sakai 2007; Paik 2013]. We use 50 queries (701–750) of terabyte track as a training set to determine the most effective *idf*.

The documents and the queries are stemmed via Porter stemmer. Stopwords are removed from the documents and the queries. Statistically significant performance differences are determined using a paired t-test at 95% confidence level ($p < 0.05$). All our experiments are carried out using the *title* field of the topics.

4.2. Baselines

The performance of the proposed model is compared to a number of state-of-the-art retrieval models from different families. BM25 [Robertson and Walker 1994] is chosen as the representative baseline from the classical probabilistic model. From the language model, we choose Dirichlet smooth version [Zhai and Lafferty 2004], since it is known to be the most effective among the language models [Fang et al. 2011]. From the divergence from randomness family, we choose PL2 [Fang et al. 2011; He and Ounis 2005] as the baseline. MATF [Paik 2013] is chosen as the state-of -the-art TF.IDF model.

4.3. Free Parameters and Evaluation Metrics

It is important to note that all the baseline models (except MATF) and the proposed model contain one or more free parameters. These parameters may influence the performance of the models to a statistically significant degree. Hence, for the sake of reliable and competitive comparison, the parameters are optimized using fivefold cross validation, where the corresponding metric (NDCG@10, NDCG@20, and NDCG@50) is optimized. Folds are created based on the query numbers. Specifically, a query having number n is assigned to the fold i ($0 \leq i \leq 4$), where $i = n\%5$.

Parameter selection by cross validation suggests that although the optimal parameter values for all the folds are not the same, the range of the variation is not wide. For BM25, the optimal value of k_1 lies between 0.5 and 0.8 and the optimal value of b lies between 0.3 and 0.5. The optimal range of the smoothing parameter of language model (μ) seems to be 750–1000, while 4–7 appears to be the optimal range of the parameter of PL2 (c). The decay function of the proposed model contains two parameters (λ and m) that control the shape and the rate of decay. Our experiments reveal that the proposed model achieves optimal performance for $\lambda \in$ [0.3–0.5], while that of m lies in [0.8–1.0]. We note that the proposed model is relatively more sensitive on m compared to λ. However, within the aforementioned range of m, the proposed model always gives optimal performance on the test collections we have used. Specifically, we recommend $m = 0.9$ and $\lambda = 0.4$ for the proposed model.

We choose NDCG@10 [Järvelin and Kekäläinen 2002], NDCG@20, and NDCG@50 as the evaluation measures. NDCG@k leverages graded relevance with a position-wise discounting. Thus, NDCG@k is more suitable for web queries. NDCG@10 is chosen to measure the precision at top 10, while NDCG@20 is chosen since it has been widely used in the recent TREC web tracks.

5. RESULTS

In this section, we present the experimental results of the proposed model and compare them to the state-of-the-art retrieval models. In Section 5.3, we compare the performance of the proposed model (PDM) to the three baselines. In Sections 5.1 and 5.2, we

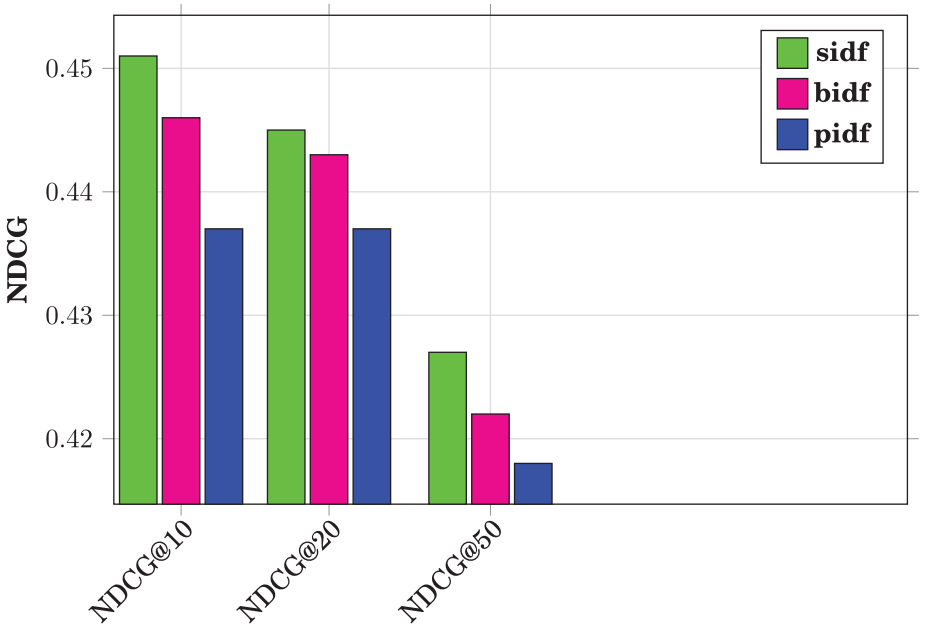

Fig. 1. Performance of the *idf* measures on the training data.

compare the performance of the proposed model to the multiaspect TF.IDF model and the BM25 multiaspect *tf* model.

In Section 2.4, we described the three *idf* measures that can be used in our model. In order to find the most effective *idf*, we carry out experiments on 50 training queries (701–750) from the TERABYTE track by integrating all three *idf* measures into the proposed model. Figure 1 shows the retrieval effectiveness of the three *idf* functions measured in terms of the three evaluation measures. The acronyms $sidf$, $bidf$, and $pidf$ denote the standard *idf*, BM25 *idf*, and Poisson based *idf*, respectively. As shown in the figure, standard *idf* is the most effective measure, while $pidf$ is the least effective, irrespective of whichever evaluation metric is chosen. BM25 *idf* is slightly worse than the standard *idf*. Since the standard *idf* is always better than the other two, all subsequent experiments are conducted using the standard *idf* measure. In Section 5.5, we carry out an additional set of experiments on the four test collections to validate the performance consistency of the *idf* measures.

5.1. Comparison to Multiaspect TF.IDF Model (MATF)

Note that the proposed model uses the two *tf* normalizations (relative intradocument *tf* and length regularized *tf*) introduced in MATF [Paik 2013] model. MATF is a TF.IDF model that uses $x/(1+x)$ as the *tf* function to transform the normalized *tf* values. Thus, the purpose of these experiments is to compare the performance of the proposed decay based term weighting model with MATF that is based on explicit TF.IDF assumption.

Table II compares the performance of PDM to MATF on the four test collections. NDCG@10 numbers clearly suggest that PDM is always superior to MATF. PDM is significantly better than MATF on three out of the four collections. PDM gives 1.3%, 40.4%, 18.7%, and 8.8% NDCG@10 improvements over MATF. Very similar conclusions can be drawn from the performance measured in terms of NDCG@20. Once again, PDM is consistently better than MATF and in all the occasions the differences are statistically significant.

Table II. Retrieval Effectiveness of the Proposed Model (PDM) Compared to the State-of-the-Art TF.IDF Model MATF. Best Result per Column is Marked by Boldface. The Numbers in Parentheses Indicate % Improvement over MATF. The Superscript *m* Denotes PDM is Statistically Significantly Better than MATF

Metric	Model	Terabyte	Web-09,10	Web-11,12	MQ-09
NDCG@10	MATF	0.553	0.183	0.193	0.365
	PDM	**0.560**	**0.257**^{m}	**0.229**^{m}	**0.397**^{m}
		(1.3)	(40.4)	(18.7)	(8.8)
NDCG@20	MATF	0.523	0.201	0.208	0.444
	PDM	**0.536**^{m}	**0.271**^{m}	**0.245**^{m}	**0.465**^{m}
		(2.5)	(34.8)	(17.8)	(4.7)
NDCG@50	MATF	**0.500**	0.254	0.252	0.490
	PDM	0.496	**0.278**^{m}	**0.279**^{m}	**0.504**^{m}
		(−0.8)	(9.4)	(10.5)	(2.8)

Experimental outcomes depict a slightly different picture when the performance is measured in terms of NDCG@50. Unlike the previous two cases, MATF is very marginally better (less than 1%) than PDM on Terabyte collection, which is clearly very insignificant. However, PDM is significantly better than MATF on the other three collections.

We note that, although MATF performs very well on the GOV2 collection, its performance is noticeably worse than PDM on the Web-09,10 and Web-11,12 collections. Since the GOV2 collection is built from a .gov domain, it is expected that the content quality adheres to a minimum standard, and also it is highly likely that the documents are free from excessive content repetitions. On the other hand, documents of the Web-09,10 and Web-11,12 collections come from unrestricted domains and therefore, spam, term repetition (by the selfish users to influence the system), and an abundance of web tables and lists (where the same term appears repeatedly) are very prevalent. We believe PDM's ability to address term repetitions better makes it more effective than MATF on the open domain web collections.

5.2. Comparison to Multiaspect Log-Logistic TF Function

In the last section, we compare the performance of PDM to MATF, both of which use two-aspect *tf*. We argued that one of the crucial factors that determines the performance of TF.IDF like weighting scheme is their choice of *tf* function. MATF uses $x/(1+x)$ as the *tf* function, which is static in nature. Although PDM uses the two-aspect *tf* normalization, its weight estimation mechanism is very different from that of MATF. The natural question then arises—can we use a different *tf* function that is not static unlike MATF's *tf* function as well as would be able to take into consideration the important nonlinear *tf* damping issue? As is well known, BM25's effectiveness depends on its asymptotically upper bounded nonlinear *tf* function [Lv and Zhai 2012]. Hence, our natural choice is to use that function along with the two-aspect normalized *tf*'s and compare its performance with PDM. We use log-logistic function ($\frac{x^k}{c+x^k}$, c and k are the parameters) as the *tf* function since BM25's *tf* function $\left(\frac{x}{c+x}\right)$ is a special case of log-logistic function (when $k=1$) and hence it has more expressive power (Lv and Zhai [2012] give a probabilistic justification on BM25's *tf* function). In this experiment, we use the same (as is used in PDM) two-aspect *tf* normalization with BM25's *tf* function and compare the performance to PDM. Once again our objective is to understand the merit of the decay model.

Table III presents the experimental results. The results reveal several facts. It is once again clear that PDM is almost always the best performer. NDCG@10 shows that PDM

Table III. Retrieval Effectiveness of the Proposed Model (PDM) Compared to Multiaspect BM25 *tf* Function. Best Result per Column is Marked by Boldface. The Numbers in Parentheses Indicate % Improvement Over LL-MATF. The Superscript *l* Denotes PDM is Statistically Significantly Better than MATF

Metric	Model	Terabyte	Web-09,10	Web-11,12	MQ-09
NDCG@10	LL-MATF	0.552	0.214	0.184	0.372
	PDM	**0.560**	**0.257**^{l}	**0.229**^{l}	**0.397**^{l}
		(1.4)	(20.1)	(24.5)	(6.7)
NDCG@20	LL-MATF	0.528	0.235	0.210	0.447
	PDM	**0.536**	**0.271**^{l}	**0.245**^{l}	**0.465**^{l}
		(1.5)	(15.3)	(16.7)	(4.0)
NDCG@50	LL-MATF	**0.502**	0.257	0.258	0.496
	PDM	0.496	**0.278**^{l}	**0.278**^{l}	**0.504**
		(−1.2)	(8.2)	(8.1)	(1.7)

Table IV. Retrieval Effectiveness of the Proposed Model (PDM) Compared to the State-of-the-Art Probabilistic Models. Best Result per Column is Marked by Boldface. The Numbers in Parentheses (from Left to Right) Indicate % Improvement Achieved by PDM over BM25, LM, and PL2, Respectively. The Superscripts *b*, *l*, and *p* Denote the Statistically Significant Improvement over BM25, LM, and PL2, Respectively

Metric	Model	Terabyte	Web-09,10	Web-11,12	MQ-09
NDCG@10	BM25	0.523	0.214	0.177	0.342
	LM	0.495	0.178	0.159	0.294
	PL2	0.506	0.179	0.158	0.296
	PDM	**0.560**^{blp}	**0.257**^{blp}	**0.229**^{blp}	**0.397**^{blp}
		(7.1, 13.1, 10.7)	(20.1, 44.4, 43.6)	(29.4, 44.0, 44.9)	(16.1, 35.0, 34.1)
NDCG@20	BM25	0.506	0.234	0.199	0.423
	LM	0.486	0.207	0.182	0.394
	PL2	0.500	0.211	0.182	0.396
	PDM	**0.536**^{blp}	**0.271**^{blp}	**0.245**^{blp}	**0.465**^{blp}
		(5.9, 10.3, 7.2)	(15.8, 30.9, 28.4)	(23.1, 34.6, 34.6)	(9.9, 18.0, 17.4)
NDCG@50	BM25	0.478	0.260	0.236	0.469
	LM	0.468	0.244	0.221	0.449
	PL2	0.477	0.250	0.224	0.450
	PDM	**0.496**^{lp}	**0.278**^{blp}	**0.279**^{blp}	**0.504**^{blp}
		(3.7, 6.0, 4.0)	(7.1, 14.0, 11.2)	(17.9, 26.4, 24.3)	(7.5, 12.3, 12.0)

is always better than LL-MATF and often the differences are significant. PDM maintains the same trend if we consider the NDCG@20 measure. Finally, NDCG@50 reveals somewhat the same picture as we had observed in the last section—LL-MATF is slightly better in GOV2 and poorer in the other cases. One notable outcome demonstrated by the experiments is that, log-logistic *tf* is slightly better than MATF's static *tf* function. We believe the presence of the parameters in log-logistic function more accurately determines the contribution of the *tf* component.

5.3. Comparison to Probabilistic Models

We now turn to analyze the performance of the proposed model (PDM) and compare the same with BM25, LM, and PL2. Table IV shows the performance of all the models measured in terms of NDCG@10, NDCG@20, and NDCG@50. We can see from Table IV that on Terabyte collection, the proposed model is always significantly better than BM25, LM, and PL2 when the performance is measured using both NDCG@10 and NDCG@20. PDM surpasses BM25, LM, and PL2 by a NDCG margin of more than

5% and the performance difference is maximum with LM. BM25 is the most effective baseline on Gov2. LM appears to be the least precise model among the baselines. NDCG@50 values give a somewhat different picture. Although PDM is numerically better than BM25, LM, and PL2, PDM is not significantly better than BM25.

On Web-09,10 collection, PDM is unequivocally superior to all three baselines. This holds for all three NDCG measures. The table clearly shows that the performance difference of PDM compared to the baselines is very large. PDM beats (in terms of NDCG@10) BM25, LM, and PL2 by 20.1%, 44.4%, and 43.6%, respectively. The behavior of PDM is more or less very consistent in terms of NDCG@20 and once again it outperforms the most competitive baseline by a more than 15% margin. Once again, BM25 is the best baseline on Web-09,10 measured in terms of all three metrics.

Our next set of experiments evaluates the effectiveness of PDM on Web 2011 and 2012 collections containing 100 topics. We can see again that the performance of PDM is significantly superior compared to the three baselines. PDM beats the strongest baseline BM25 by 29.4% and 23.1% margins, measured using NDCG@10 and NDCG@20, respectively. As in the previous cases, the performance differences with LM and PL2 are even larger (more than 30%). A very similar conclusion can be drawn from NDCG@50 figures where PDM significantly surpasses all the baselines with a more than 17% margin.

We now analyze the results of the methods on the MQ-09 collection that contains more than 600 queries taken from a commercial search engine log. The performance of PDM remains very consistent as in the previous cases. PDM outperforms BM25, LM, and PL2 by a NDCG@10 margin of 16.1%, 35%, and 34.1%, respectively, and the differences are always statistically significant. NDCG@20 and NDCG@50 reveal a very similar trend, where PDM is always significantly better than the baselines.

In summary, the experimental results clearly demonstrate that the performance of the proposed model is always better than BM25, LM, and PL2 irrespective of what evaluation measure is chosen. More importantly, PDM surpasses the baselines with a highly significant margin on the ClueWeb collections. Moreover, the results also attest that PDM is significantly more precise than the baselines. BM25 appears to be the most effective and precise baseline probabilistic model.

5.4. Effect of Spam Filtering

The previous experiments demonstrate that the proposed model performs better than the state-of-the-art models in many cases, especially for early precision metrics (e.g., NDCG@10). This demonstrates that the model deals better with repeated term occurrence, which is prevalent on the web (e.g., spam pages, pages with duplicate or near-duplicate repeated text, web tables and lists). Thus, the outcomes validate our intuition that the proposed model is able to address the problem of term burstiness [Cummins et al. 2015] better than prior work. This is especially helpful for a larger heterogeneous web corpus such as ClueWeb where many of the documents are spammy with many term repetitions. The experiments conducted so far did not answer whether PDM's ability to tackle spam due to term repetitions is the major reason behind the significant effectiveness of PDM over the other baselines. In order to answer this question, we remove spam documents from the ClueWeb collection. Specifically, documents assigned by Waterloo's spam classifier [Cormack et al. 2011] with a score below 50 were filtered out from the initial corpus. The models are then run on the residual corpus to produce final ranked lists. In this section, we are interested to see how PDM performs (early precision measured in terms of NDCG@10) compared to the state-of-the-art models when a spam filter is applied. Since spam is not an issue for the GOV2 colletion, we conduct experiments on the ClueWeb collection. Also, we do not apply spam filtering on MQ-09 queries, since these queries are evaluated only based on the judged documents

Table V. Performance of the Models When Spam Filtering is Applied. The Evaluation Measure is NDCG@10. Best Result per Column is Marked by Boldface. The Superscripts *b*, *l*, *p*, *m*, *L* Denote the Performance is Significantly Better than BM25, LM, PL2, MATF, and LL-MATF, Respectively

Collection	BM25	LM	PL2	MATF	LL-MATF	PDM
Web-09,10	0.280	0.278	0.283	0.275	0.284	0.312^{blpmL}
Web-11,12	0.224	0.225	0.230	0.224	0.228	0.263^{blpmL}

Table VI. Effectiveness of Various Term Discrimination Functions. The Best Results are Boldfaced. sidf, bidf, and pidf Denote Standard idf, BM25 idf, and Poisson Idf, Respectively

		Terabyte	Web-09,10	Web-11,12	MQ-09
NDCG@10	sidf	**0.560**	**0.257**	**0.229**	**0.397**
	bidf	0.558	**0.257**	0.221	0.392
	pidf	0.547	0.251	0.212	0.383
NDCG@20	sidf	**0.536**	**0.271**	**0.245**	0.465
	bidf	0.530	0.270	0.240	**0.468**
	pidf	0.517	0.266	0.229	0.453

(which are unlikely to be spammy). Thus, spam documents are not part of the ranked lists that have been evaluated.

Table V compares the performance of the models on spam filtered residual ClueWeb corpus. The results clearly show that PDM is significantly better than all the baselines on both the collections.

5.5. Effect of Term Discrimination Function

We reiterate that, to measure the weight of a term, the proposed model starts with an initial weight of the term. We mentioned that the initial weight can be any reasonable term discrimination function that measures the contribution made by a single occurrence of the term. We already evaluated the term discrimination functions on training data and we conclude that the standard *idf* has been the most effective measure. In this experiment, we investigate the performance consistency of the term discrimination functions. The experiments are carried out for the three term discrimination functions on all four test collections.

Table VI compares the performance of the three term discrimination functions. It is clearly evident that the standard *idf* ($\log(N/df)$) consistently outperforms the other two functions on four datasets and measured under both evaluation metrics. BM25 *idf* seems to be performing slightly worse than the standard *idf* and slightly better than *pidf*. Poisson *idf* is the poorest performer. We note that, although the standard *idf* is better than the BM25 *idf*, the differences are never statistically significant. This holds unequivocally for all the evaluation measures. Overall, the results suggest that either the standard *idf* or the BM25 *idf* can be used with our model without sacrificing the performance to a statistically significant degree. However, since standard *idf* is always numerically better than BM25 *idf*, we recommend standard *idf*.

6. CONCLUSION

In this article, we introduce a novel term weighting model. The major point that distinguishes our model from many well known models is that the proposed model estimates term weight using a parameterized decay function that explicitly models the dependence of separate occurrences of a term in a document. The model starts with the assumption that the weight of a term at a particular moment decays at a rate that is a

function (possibly nonlinear) of the weight at that point. This hypothesis is then used to derive the resulting weighting function.

A set of experiments on a number of recent test collections containing homogeneous (GOV2) as well as heterogeneous (ClueWeb) web data shows that the model often outperforms BM25, language model (Dirichlet prior), a probabilistic DFR model (PL2), and a recently proposed effective TF.IDF model (MATF) with a significantly large margin. An additional set of experiments reveals that the performance of the proposed model is also better than a model that is based on log-logistic *tf* function and the two-aspect *tf* normalization. The results also demonstrate that our model is significantly more precise compared to the state-of-the-art models.

APPENDIX

Derivation of F_t (Equation (22))

$$F_t(x) = \lim_{a \to 0+} \int_a^x (-\lambda(1-m)x + (f_t^0)^{1-m})^{\frac{1}{1-m}}\, dx. \tag{39}$$

Let

$$z = -\lambda(1-m)x + (f_t^0)^{1-m}. \tag{40}$$

From Equation (40), taking differential we have

$$dx = \frac{-1}{\lambda(1-m)} dz. \tag{41}$$

When $x = a$, Equation (40) gives the following.

$$z = -\lambda(1-m)a + (f_t^0)^{1-m}. \tag{42}$$

Again, when $a \to 0+$, we have from Equation (42),

$$z = (f_t^0)^{1-m}. \tag{43}$$

Hence, the lower limit of z is $(f_t^0)^{1-m}$ and it is denoted as l (i.e., $l = (f_t^0)^{1-m}$).

Putting dx (from Equation (41)) into Equation (39) we have

$$F_t(x) = \int_l^z \frac{1}{-\lambda(1-m)} z^{\frac{1}{1-m}}\, dz. \tag{44}$$

Case I: ($\frac{1}{1-m} \neq -1$) From Equation (44) we have

$$F_t(x) = \int_l^z \frac{1}{-\lambda(1-m)} z^{\frac{1}{1-m}}\, dz \tag{45}$$

$$= \frac{1}{-\lambda(1-m)} \left(\frac{1-m}{2-m}\right) (z^{1+\frac{1}{1-m}} - l^{1+\frac{1}{1-m}}) \tag{46}$$

$$= \frac{1}{-\lambda(1-m)} \left(\frac{1-m}{2-m}\right) (z^{\frac{2-m}{1-m}} - l^{\frac{2-m}{1-m}}) \tag{47}$$

$$= \frac{1}{-\lambda(1-m)} \left(\frac{1-m}{2-m}\right) (z^{\frac{2-m}{1-m}} - (f_t^0)^{2-m}) \text{(replacing l)} \tag{48}$$

$$= \frac{1}{\lambda(2-m)} ((f_t^0)^{2-m} - z^{\frac{2-m}{1-m}}). \tag{49}$$

Case II: ($\frac{1}{1-m} = -1$) From Equation (44) we have

$$F_t(x) = \int_l^z \frac{1}{-\lambda(1-m)} \frac{1}{z}\, dz \tag{50}$$

$$= \frac{1}{-\lambda(1-m)} (\log(z) - \log(l)) \tag{51}$$

$$= \frac{1}{-\lambda(1-m)} (\log(z) - (1-m)\log(f_t^0)) \tag{52}$$

$$= \frac{1}{\lambda} (\log(f_t^0) - \frac{1}{1-m}\log(z)). \tag{53}$$

ACKNOWLEDGMENTS

I thank the anonymous reviewers whose comments have greatly helped to improve the article.

REFERENCES

Gianni Amati and Cornelis Joost Van Rijsbergen. 2002. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.* 20, 4 (Oct. 2002).

Michael Bendersky, Donald Metzler, and W. Bruce Croft. 2011. Parameterized concept weighting in verbose queries. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 605–614.

Kenneth Church and William A. Gale. 1995. Inverse document frequency (IDF): A measure of deviations from Poisson. In *Proceedings of the 3rd Workshop on Very Large Corpora*. 121–130.

Stéphane Clinchant and Eric Gaussier. 2011. Retrieval constraints and word frequency distributions: A log-logistic model for IR. *Inf. Retr.* 14, 1 (Feb. 2011), 5–25.

Gordon V. Cormack, Mark D. Smucker, and Charles L. A. Clarke. 2011. Efficient and effective spam filtering and re-ranking for large web datasets. *Inf. Retr.* 14, 5 (2011), 441–465.

Ronan Cummins, Mounia Lalmas, and Colm O'Riordan. 2010. Examining the information retrieval process from an inductive perspective. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM'10)*. 89–98.

Ronan Cummins and Colm O'Riordan. 2012. A constraint to automatically regulate document-length normalisation. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM'12)*. 2443–2446.

Ronan Cummins, Jiaul H. Paik, and Yuanhua Lv. 2015. A Pólya urn document language model for improved information retrieval. *ACM Trans. Inf. Syst.* 33, 4 (May 2015), 21:1–21:34.

Hui Fang, Tao Tao, and Chengxiang Zhai. 2011. Diagnostic evaluation of information retrieval models. *ACM Trans. Inf. Syst.* 29, 2 (April 2011).

Willliam Feller. 2008. *An Introduction to Probability Theory and its Applications*. Vol. 2. John Wiley & Sons.

Warren R. Greiff. 1998. A theory of term weighting based on exploratory data analysis. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98)*. 11–19.

Ben He and Iadh Ounis. 2005. A study of the Dirichlet priors for term frequency normalisation. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'05)*. 465–471.

Djoerd Hiemstra, Stephen Robertson, and Hugo Zaragoza. 2004. Parsimonious language models for information retrieval. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'04)*. 178–185.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.* 20, 4 (Oct. 2002), 422–446.

Matthew Lease, James Allan, and W. Bruce Croft. 2009. Regression rank: Learning to meet the opportunity of descriptive queries. In *Advances in Information Retrieval*. 90–101.

Yuanhua Lv and ChengXiang Zhai. 2011a. Adaptive term frequency normalization for BM25. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM'11)*.

Yuanhua Lv and ChengXiang Zhai. 2011b. Lower-bounding term frequency normalization. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM'11)*. 7–16.

Yuanhua Lv and ChengXiang Zhai. 2012. 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)*. 244–255.

Rasmus E. Madsen, David Kauchak, and Charles Elkan. 2005. Modeling word burstiness using the Dirichlet distribution. In *Proceedings of the 22nd International Conference on Machine Learning (ICML'05)*. 545–552.

Jiaul H. Paik. 2013. 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)*.

Jay M. Ponte and W. Bruce Croft. 1998. 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)*.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.* 3, 4 (April 2009).

S. E. Robertson. 1997. Readings in information retrieval. Morgan Kaufmann Publishers Inc., San Francisco, CA, 281–286.

S. E. Robertson and S. Walker. 1994. 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)*.

Thomas Roelleke and Jun Wang. 2008. 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)*. 435–442.

Tetsuya Sakai. 2007. Alternatives to Bpref. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'07)*. 71–78.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.* 24, 5 (Aug. 1988).

Gerard Salton and Michael J. McGill. 1986. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc.

G. Salton, A. Wong, and C. S. Yang. 1975. A vector space model for automatic indexing. *Commun. ACM* 18, 11 (Nov. 1975), 613–620.

Amit Singhal. 2001. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.* 24, 4 (2001), 35–43.

Amit Singhal, Chris Buckley, and Mandar Mitra. 1996. Pivoted document length normalization. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'96)*.

Karen Sparck Jones. 1988. Document retrieval systems. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation* 28, 1 (1972), 132–142.

Howard Turtle and W. Bruce Croft. 1991. Evaluation of an inference network-based retrieval model. *ACM Trans. Inf. Syst.* 9, 3 (July 1991), 187–222.

Chengxiang Zhai and John Lafferty. 2004. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.* 22, 2 (April 2004), 179–214.

Dell Zhang, Jinsong Lu, Robert Mao, and Jian-Yun Nie. 2009. Time-sensitive language modelling for online term recurrence prediction. In *ICTIR*. 128–138.

Le Zhao and Jamie Callan. 2010. Term necessity prediction. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. ACM, 259–268.

Received July 2014; revised June 2015; accepted July 2015