A Novel TF-IDF Weighting Scheme for Effective Ranking

Jiaul H. Paik
Indian Statistical Institute, Kolkata, India
jia.paik@gmail.com

ABSTRACT

Term weighting schemes are central to the study of information retrieval systems. This article proposes a novel TF-IDF term weighting scheme that employs two different within document term frequency normalizations to capture two different aspects of term saliency. One component of the term frequency is effective for short queries, while the other performs better on long queries. The final weight is then measured by taking a weighted combination of these components, which is determined on the basis of the length of the corresponding query.

Experiments conducted on a large number of TREC news and web collections demonstrate that the proposed scheme almost always outperforms five state of the art retrieval models with remarkable significance and consistency. The experimental results also show that the proposed model achieves significantly better precision than the existing models.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Search and Retrieval : Retrieval Models

General Terms

Algorithm, Experimentation, Performance

Keywords

Document ranking, Retrieval model, Term weighting

1. INTRODUCTION

Term weighting schemes are the central part of an information retrieval system. Effectiveness of IR systems are thus crucially dependent on the underlying term weighting mechanism. Almost all retrieval models integrate three major variables to determine the degree of importance of a term for a document: (i) within document term frequency, (ii) document length and (iii) the specificity of the term in the collection. Term frequency and document length combination is used to infer the saliency of a term in a document, and when a query contains more than one term, term specificity is used to reward the documents that contain the terms rare in the collection.

Retrieval models can be broadly classified into two major families based on their term weight estimation principle. Vector space model casts queries and documents as finite dimensional vectors, where the weight of an individual component is computed using numerous variations of tf-idf scores. On the other hand, probabilistic models [16, 17] primarily focus on estimating the probabilities of the terms in the documents, and the estimation techniques differ from one approach to the other. But in essence all of them use the same basic principles that we have outlined before.

Most of the existing models (possibly all) employ a single term frequency normalization mechanism that does not take into account various aspects of a term's saliency in a document. For example, frequency of a term in a document relative to the frequency of the other terms in the same document gives us an important clue that can not be achieved by the commonly used document length based normalization scheme. On the contrary, length based normalization can restrict the likelihood of retrieval of extremely long documents which can not be taken care of by the relative frequency based term weighting.

Another major limitation of the present models is that they do not balance well in preferring short and long documents. Such limitation makes a system to retrieve low quality documents at the top of the ranked list when they face queries of varying length. For example, in pivoted document length normalization scheme, if the parameter is set to a smaller value, it performs better for shorter queries, and when the parameter value is larger, longer queries are benefited more than the shorter queries [10]. Similar observation can be made for other models such as BM25, language model or relatively recent divergence from randomness based models [13, 10].

The main reason is that when the parameter is set to a static value, most of the models prefer either short documents or long documents. If a weighting scheme prefers long documents, it pulls up extremely long documents when longer queries are encountered, since the longer documents have higher verbosity level it matches more query terms[28]. On the other direction, preference of short documents may degrade the overall retrieval performance, since it violates the likelihood of relevance versus retrieval pattern suggested by Singhal et al. [28].

SIGIR'13, July 28–August 1, 2013, Dublin, Ireland.

This article proposes a term weighting scheme that can address these weaknesses in an effective way. In particular, we argue that the two aspects of term frequencies, when combined appropriately, leads to significant performance benefit. In this article we make the following contributions.

- It introduces a two aspect term frequency normalization scheme, that combines relative tf weighting and the tf normalization based on document length. One component of the term frequency tends to prefer long documents, while the other component prefers short documents and therefore, it maintains a good balance in preferring short and long documents.
- It uses the query length information to emphasize the appropriate component. In particular, when the system faces a long query, it down-weights the part that prefers long documents in order to compensate the effect and vice versa.
- It modifies the usual term discrimination measure (idf) by integrating mean term frequency of a term in the set of documents the term is contained in.
- Finally, we use asymptotically bounded function (similar to Robertson and Walker [22]) to transform the tf factors that better handles the term coverage issues in the documents and also helps to combine the two tf factors more easily. As a bi-product of such transformation, the ranking function easily produces the similarity values in the range of [0-1].

In order to assess the effectiveness of the proposed model we carry out a set of experiments on a large number of standard test collections containing news and web data. The experimental results show that the proposed weighting function consistently and significantly outperforms five state of the art retrieval models (from vector space as well as probabilities families) measured in terms of the standard metrics such as MAP and NDCG. The experimental outcomes also attest that the model achieves significantly better precision than all the other models when measured in terms of a recently popularized metric, namely, expected reciprocal rank (ERR) [6].

The remainder of the article is organized as follows. In Section 2 we review the state of the art. Section 3 describes the proposed weighing scheme. Description about the test collections, evaluation metrics and the details of the baselines are given in Section 4. Experimental results are presented in Section 5, where we compare the performance of the proposed model with the state of the art vector space models, followed by the comparison with the probabilistic models. Finally, we conclude in Section 6.

2. STATE OF THE ART

Information retrieval systems, when encounter a query, tries to rank documents by their likelihood of relevance. Most IR systems assign a numeric score to the documents and then they are ranked based on these scores. Three widely used models in IR are the vector space model [26, 25], the probabilistic models [21] and the inference network based model [30]. In this section we review some of the state of the art models.

In vector space model, 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. Three main factors that come into play to compute 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 [29]) and (iii) the length of the document that contains the term. Fang et al. [10] 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 [8, 9].

Salton and Buckley [24] summarize a number of term weighting approaches which use various types of normalization. It is evident that document length is an important component in effective term weighting. Singhal et al. [28] 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 as follows [27]:

$$\sum_{t \in Q \cap D} \frac{1 + ln(1 + ln(TF(t,D)))}{1 - s + s\frac{len(D)}{ADL(\mathcal{C})}} \cdot TF(t,Q) \cdot ln\frac{N+1}{df(t)} \quad (1)$$

The parameter s controls the extent of normalization with respect to the document. Typically, the term weighting functions in vector space model are designed heuristically, which are based on the researchers experience. Several work tried to use the data to learn the patterns that satisfy the data. For example, Greiff [11] uses exploratory data analysis to uncover some important relationship between the document frequency and the relevance of a document.

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 model differs from one another. Binary independence model is perhaps the most widely accepted technique in the classical probabilistic model. A number of weighting formula have been developed and BM25 [20] has been the most effective among the formulae. The major differences between BM25 and the other commonly used TF-IDF models are the slightly variant IDF formulation and the use of the query term frequency. 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 [19, 14] is another effective ranking model that is widely used today. 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 BM25 model. Three major smoothing techniques (Dirichlet, Jelinek-Mercer and Two-stage) are commonly used in this model [31].

Relatively recent, another probabilistic model is proposed in [3] that computes the weight of a term by measuring the informative content of a term by computing the amount of divergence of the term frequency distribution from the distribution based on a random process. Like most of the well known models, they also use the same basic components. However, the integration of various component are derived theoretically. This family of formula also 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 inference network, document retrieval is modeled as an inference process [30]. 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 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.

3. PROPOSED WORK

3.1 Preliminaries

Given a query Q and a document D, the main task of a ranking function is to assign a score to D with respect to the query Q. The main objective of a term weighting scheme is to quantify the saliency of the query terms in the document. This section describes a novel TF-IDF term weighting scheme that serves above purpose. En-route to the development, we are guided by a number of hypotheses that are commonplace in quantifying the importance of a term. Thus, before we give the main motivation behind our work, let us first revisit the three key hypotheses.

1. **Term Frequency Hypothesis (TFH)**: The weight of term in a document should increase with the increase in term frequency (TF). However, it seems unlikely that the importance of a term grows linearly with the increase in TF. Therefore, researchers have used dampened TF instead of the raw TF for ranking. The most widely used damping function has been $log(TF)$ and the basis of this damping can be best captured by the following advanced hypothesis.

   Advanced TF Hypothesis (AD-TFH): The modified term frequency hypothesis captures the intuition that the rate of change of weight of a term should decrease with the larger TF. For example, the change in the weight caused by increasing TF from 2 to 3 should be higher than that caused by increasing TF from 25 to 26 [10]. Thus, the raw TF has to be transformed to fulfill the above goal. Formally, we hypothesize that, a function $F_t(TF)$, that maps the original TF to the resultant value (which will be used for final weighting), should possess the following two properties.

   (a) $F_t'(TF) > 0$

   (b) $F_t''(TF) < 0$

2. **Document Length Hypothesis (DLH)**: This hypothesis captures the relationship between the term frequency and the document length. Long documents tend to use a term repeatedly, thus term frequency can be higher in a long document. Therefore, if TF is considered in isolation (disregarding the document length), long documents are given undue preference. Thus, it is necessary to regulate the TF value in accordance with the document length. The general principle is that if two documents have different lengths and the same TF values for a term t, then the contribution of TF (of t) should be higher for the shorter document.

3. **Term Discrimination Hypothesis (TDH)**: If a query contains more than one term, then a good weighting scheme should prefer a document that contains the rare term (in the collection).

3.2 Two Aspects of TF

Most existing weighting schemes employ the above heuristics to quantify the term importance. However, they generally normalize the term frequency that captures a single aspect of the saliency of the terms and hence disregards another important aspect that we detail next. In particular, we consider the following two aspects:

1. **Relative Intra-document TF (RITF)** : In this factor, the importance of a term is measured by considering its frequency relative to the average TF of the document. Thus, a natural formulation for this could be

$$RITF(t, D) = \frac{TF(t, D)}{Avg.TF(D)} \tag{2}$$

   where $TF(t, D)$ and $Avg.TF(t, D)$ denote the frequency of the term t in D and average term frequency of D respectively. However, Equation 2 may too much prefer excessively long documents, since the denominator is close to 1 for a very long document [28]. Hence, a sublinear damping of TF seems to be a better choice over the raw TF and thus we use the following function.

$$RITF(t, D) = \frac{log_2(1 + TF(t, D))}{log_2(1 + Avg.TF(D))} \tag{3}$$

   Indeed, such a formula has been used by Singhal et al. [28] in the pivoted length normalization framework to normalize the tf values in accordance with the number of unique terms in the document.

2. **Length Regularized TF (LRTF)** : This factor normalizes the term frequency by considering the number of terms present in a document. Similar to Robertson's [22] notion, we assume that the appropriate length of a document should be the average document length of the collection and the frequency of the terms of an average length document should remain unchanged. Thus, a reasonable starting point could be $TF(t, D) \times \frac{ADL(\mathcal{C})}{len(D)}$, where $ADL(\mathcal{C})$ is the average document length of the collection and $len(D)$ is the length of the document D. But once again, it seems unlikely that the increase in term frequency follows a linear relationship with the document length, and thus the above formula over-penalizes the long documents. To overcome this

bias, we employ the following function (used in [3]) to achieve the length dependent normalization.

$$LRTF(t,D) = TF(t,D) \times log_2\left(1 + \frac{ADL(\mathcal{C})}{len(D)}\right) \quad (4)$$

Equation 4 still punishes the long documents but with diminishing effect.

However, we believe that any document length normalization can be used to achieve this purpose. Some of the possible alternatives might be the length normalization component of BM25 or that of the pivoted normalization scheme.

3.2.1 Motivation

In order to motivate the use of two TF factors, let us consider the following two somewhat toy examples. We use these examples just to introduce the basic idea.

Example 1 Let D_1 and D_2 be two documents of equal lengths, with the following statistics.

1. $len(D_1) = 20$, # distinct term of $D_1 = 5$, $TF(t, D_1)$=4
2. $len(D_2) = 20$, # distinct term of $D_2 = 15$, $TF(t, D_2)$=4

In both of the cases, $LRTF$ considers t equally important. A little thought will convince us that this is not appropriate, since the focus of the document D_1 seems to be divided equally among 5 terms and therefore t should not be considered salient, while t seems to be important for D_2. Thus, in the later case $RITF$ seems to be a better choice to $LRTF$.

Let us now turn to the other direction and consider the second example.

Example 2 Let D_1 and D_2 be two documents with the following statistics.

1. $len(D_1) = 20$, # distinct term of $D_1 = 15$, $TF(t, D_1)$=4
2. $len(D_2) = 200$, # distinct term of $D_2 = 150$, $TF(t, D_2)$=4

For this instance however, $RITF$ considers the term t equally important for both D_1 and D_2, which is not right, since D_2 contains more distinct terms and thus seems to cover many other topics (also possibly uses t repeatedly). Therefore, in this case, the use of $LRTF$ seems to be a potential choice over $RITF$.

Motivated by the above examples, our main goal now is to integrate the above two factors into our weighting scheme. However, we do not use the TF factors as defined in the Equations 3 and 4. We transform these TF values for our final use that in some sense makes use of the hypothesis **AD-TFH**. The next section details the transformation procedure and the underlying motivation.

3.2.2 Transforming TF Factors

Recall that the main motivation behind the advanced hypothesis on term frequency (AD-TFH) is that a good weighting function, while emphasizing on term frequencies and term discrimination factors, should also pay attention to the term coverage issue (i. e number of match). For example, if a document D_1 contains a query term 10 times and another document D_2 contains two query terms (of the same query) 5 times each, then the assigned score should be higher for D_2 (assuming that both the query terms have equal term discrimination values). That is probably the most important reason why raw TF does not work well in practice. Second, another common trait of many weighting schemes (for example, pivoted normalization) is that they use the TF functions that are not bounded above. We transform the TF factors using a function $f(x)$ that possesses the following properties: (i) vanishes at 0, (ii) satisfies the two conditions of **AD-TFH** hypothesis ($f'(x) > 0$ and $f''(x) < 0$), and (iii) asymptotically upper bounded to 1.

One of the simplest functions that satisfies the above properties is $f(x) = \frac{x}{1+x}$. Indeed, similar functions have been employed before in [22] and in [3]. Using this function, we now transform the two TF factors as follows:

$$BRITF(t,D) = \frac{RITF(t,D)}{1 + RITF(t,D)} \quad (5)$$

$$BLRTF(t,D) = \frac{LRTF(t,D)}{1 + LRTF(t,D)} \quad (6)$$

3.2.3 Combining Two TF Factors

Now the key question that we face: how should we combine $BRITF(t,D)$ and $BLRTF(t,D)$? A natural way to do this is as follows:

$$TFF(t,D) = w \times BRITF(t,D) + (1-w) \times BLRTF(t,D) \quad (7)$$

where $0 < w < 1$. The next important issues that arise out of Equation 7 are the following:

- Should we prefer $BRITF(t,D)$ $(w > 0.5)$?
- Should we prefer $BLRTF(t,D)$ $(w < 0.5)$?

In order to answer these questions, we now analyze the properties of the two TF components. From Equation 5, it is clear that $BRITF(t,D)$ has a tendency to prefer long documents, since for long documents the denominator part of $RITF(t,D)$ is close to 1, and TF is usually larger. On the other hand, $BLRTF(t,D)$ tends to prefer short documents, since $LRTF(t,D) \to 0$ as $len(D) \to \infty$. Therefore, when a query is long, $BRITF(t,D)$ heavily prefers extremely long documents, since the number of matches is more or less proportional to the length of the document [28]. On the contrary, since $BLRTF(t,D)$ prefers short documents it can penalize extremely long documents when it faces longer queries, and thus it is preferable when longer queries are encountered. Another interesting property of $BRITF(t,D)$ is that it emphasizes on the number of matches, since the main component of this formula $RITF(t,D)$ heavily punishes the term frequency, and thus important for the short queries. Hence, the foregoing discussion suggests that, for short queries $BRITF(t,D)$ should be preferred, while for longer queries, $BLRTF(t,D)$ should be given more weight

Based on the discussion given in the previous section, we now turn to incorporate the query length information into our weighting formula. The value of w should decrease with the increase in query length, while it must lie between [0-1]. Specifically, we characterize the query length factor $(QLF(Q))$ by the following variables. (i) $QLF(Q) = 1$ for $|Q| = 1$, (ii) $QLF'(Q) < 0$ and (iii) $0 < QLF(Q) < 1$. Numerous different functions can be constructed that satisfy the above conditions. We used the following three different

functions.

$$QLF_1(Q) = \frac{1}{log_2(1+|Q|)} \quad (8)$$

$$QLF_2(Q) = \frac{2}{1+log_2(1+|Q|)} \quad (9)$$

$$QLF_3(Q) = \frac{3}{2+log_2(1+|Q|)} \quad (10)$$

The first function descends more rapidly than the second function, while the second function descends more rapidly than the third function. Our experiments suggest that function 9 performs consistently better than the other two functions on all the collections. Hence, we set $w = QLF_2(Q)$. We leave this issue for further investigation.

3.3 Term Discrimination Factor

The goal of the term discrimination factor in weighting is to assign higher score to the documents that contain the terms which are rare in the collection. Inverse document frequency (IDF) is a well known measure that serves the above purpose. A number of IDF formulation are prevalent in the IR literature, all of which essentially quantify the above intuition. We use the following standard *idf* measure.

$$IDF(t,\mathcal{C}) = log\left(\frac{CS(\mathcal{C})+1}{DF(t,\mathcal{C})}\right) \quad (11)$$

The above IDF measure considers only the presence or absence of a term in a document and does not take into account the document specific term occurrence. We hypothesize that the term discrimination is a combination of the above two factors. In particular, we hypothesize that if two terms have equal document frequencies, then the term discrimination should increase with the increase in average elite set term frequency. The average elite set term frequency (AEF) is defined as $\frac{CTF(t,\mathcal{C})}{DF(t,\mathcal{C})}$, where $CTF(t,\mathcal{C})$ denotes the total occurrence of the term t in the entire collection. In fact, Kwok [18] used AEF for term weighting, but the purpose was different. However, the combination of raw AEF with IDF may disturb the overall term discrimination value, since the IDF values are obtained by dampening through *log* function. Hence, we employ a slowly increasing function to transform the AEF values for this combination. Once again, we use the function $f(x) = x/(1+x)$ to transform the AEF values for the final use. The final term discrimination value of term t is computed as

$$TDF(t,\mathcal{C}) = IDF(t,\mathcal{C}) \times \frac{AEF(t,\mathcal{C})}{1+AEF(t,\mathcal{C})} \quad (12)$$

Our experiments reveal that the use of the above term discrimination has not very significant effect on the overall performance. However, it is observed that the improvements, although are small, consistent across the collections.

3.4 Final Formula

Integrating the above factors we now obtain the following final scoring formula.

$$Sim(Q,D) = \sum_{i=1}^{|Q|} TFF(q_i,D) \times TDF(q_i,\mathcal{C}) \quad (13)$$

Again, since $TFF(q_i,D) < 1$, we obtain the following relationship.

$$Sim(Q,D) < \sum_{i=1}^{|Q|} TDF(q_i,\mathcal{C}) \quad (14)$$

Therefore, we can easily modify Equation 13 to get the normalized similarity scores ($0 < Sim(Q,D) < 1$) as follows:

$$Sim_{norm}(Q,D) = \frac{\sum_{i=1}^{|Q|} TFF(q_i,D) \times TDF(q_i,\mathcal{C})}{\sum_{i=1}^{|Q|} TDF(q_i,\mathcal{C})} \quad (15)$$

Equations 13 and 15 are equivalent in the sense that they produce the same ranked lists. However, an application that requires normalized scores, Equation 15 can be used as a suitable alternative.

4. EXPERIMENTAL SETUP

In this section we describe the details of our experimental setup. First, in Section 1 we give the details of the test collections used in our experiments. In Section 4.2 and Section 4.3 we describe the evaluation measures and the baseline retrieval models respectively.

4.1 Data

Table 1 summarizes the statistics on test collections used in our experiments. The experiments are conducted on a large number of standard test collections, that vary both by type, the size of the document collections and the number of queries.

TREC 6,7,8 and ROBUST are news collections containing 528,155 documents and supplemented by 150 (queries 301-450) and 100 (601-700) queries respectively. WT10G is a web collection of moderate size supplemented by 100 queries (451-550), while GOV2 is another web collection of larger size, which is crawled from .gov domain. There are 150 (queries 701-850) queries attached with GOV2 collection which were used in TREC terabyte [4] track for three years.

Table 1: Test Collection Statistics

Name	# of docs	# of queries
TREC 6,7,8	528,155	150
ROBUST	528,155	100
WT10G	1,692,096	100
GOV2	25,205,179	150
MQ-07	25,205,179	1778
MQ-08	25,205,179	784

The MQ-07 and MQ-08 set of queries are based on the Million Query Track 2007 [2] and 2008 [1] respectively. This track was designed to serve two purposes. First, it was an exploration of ad-hoc retrieval on a large collection of documents. Second, it investigated questions of system evaluation, particularly whether it is better to evaluate using many shallow judgments or fewer thorough judgments. Both million query track use GOV2 as document collection. Topics for this task were drawn from a large collection of queries that were collected by a large Internet search engine. The queries also vary by their length, with short (2-3 words) to long (6-10 words). Specifically, MQ-07 and MQ-08 collections contain 505 and 433 queries of length higher than 5

respectively. Therefore, the test collections provide us a diverse experimental setup for assessing the effectiveness of the proposed weighting method.

Except TREC-6,7,8, all the test collections have three scale graded relevance assessment. The grades are 0, 1 and 2- meaning non-relevant, relevant and highly relevant respectively. TREC-6,7,8 collection uses binary relevance assessment.

4.2 Evaluation Measures and IR System

All our experiments are carried out using TERRIER[1] retrieval system (version 3.5). Terrier is a flexible Information retrieval system which provides the implementation of many well known models. We use title field of the topics (note that two million query data contain more than 1000 queries that contain more than 5 terms). From all the collections we removed stopwords during indexing. Documents and queries are stemmed using Porter stemmer. Statistical significance tests are done using two sided paired t-test at 95% confidence level (i,e $p < 0.05$).

We use the following metrics to evaluate the systems.

- Mean Average Precision (MAP): This is a standard metric for binary relevance assessment.
- Normalized DCG at k (NDCG@k) [15]: Discounted cumulative gain (DCG) is an evaluation measure that can leverage the relevance judgment in terms of multiple grades, and has an explicit position-wise discount factor. NDCG is the normalized version of DCG.
- Expected Reciprocal Rank at k (ERR@k) [6]: To relax the additive nature and the underlying independent assumption in NDCG, another evaluation measure, namely, Expected Reciprocal Rank (ERR) is proposed in [6]. It discounts the documents which are shown below very relevant documents, and is defined as the expected reciprocal length of time that a user will take to find a relevant document. ERR@k is computed as follows:

$$ERR@k = \sum_{i=1}^{k} \frac{R(g_i)}{i} \prod_{j=1}^{i-1}(1 - R(g_i)) \qquad (16)$$

  where $R(g) = \frac{2^g - 1}{2^{hg}}$, hg is the highest grade and $g_1, g_2 \ldots g_k$ are the relevance grades associated with the top k documents. The value of mg is 2 for all the collections, except TREC 6,7&8 (for this $mg = 1$).

The first two metrics are used to reflect the overall performance of the systems, while the last evaluation measure reflects better the precision of search results, thereby making more important for the precision oriented systems. ERR has been chosen as one of the official metrics for recent TREC web tracks [7].

Note that, two million query collections (MQ-07 and MQ-08) have incomplete relevance assessment. Therefore, for the sake of more reliable conclusions, we evaluate the million query sets in two different ways. First, we skip the unjudged documents from the ranked lists to compute the values of well known metrics following the recommendation made in [23]. Additionally, we also present the statistical average precision[2] [5] which was one of the official metrics for the million query tracks [2].

[1] http://terrier.org/

4.3 Baselines

We have compared the performance of the proposed weighting scheme with five state of the art retrieval models. Since the proposed weighting function is a TF-IDF based formula, we have taken two well known state of the art TF-IDF models. We have also chosen BM25, language model with Dirichlet smoothing (LM), and relatively recent divergence from randomness based formula (PL2) as the other state of the art baselines. The choice of our baselines are primarily motivated by [10], which provides a thorough and detailed description of all the state of the art models along with the parameter sensitivity issues.

The performances of all the baseline models are dependent on the parameters they contain. Therefore, for the sake of more reliable comparisons with the baselines, we carry out two experiments by taking first 50 judged queries from MQ-07 and MQ-08 collections. We search parameters by optimizing NDCG@20. The parameters values are given in the description of the corresponding baselines. Our experiments mostly agree with the findings reported by Fang et al. [10]. In particular, we find that the performances of Pivoted TF-IDF and PL2 are very sensitive with the variation of the parameters. For example, the parameter value ($s = 0.2$) suggested for pivoted TF-IDF in the original paper [28] gives 12% poorer MAP than that we find here by training. Similarly, the default PL2 parameter ($c = 1$) is 14% poorer than the one we find. Therefore, for fair comparisons, we use these optimal parameter values for the baselines. The details of the baselines are given below.

1. Pivoted length normalized TF-IDF model: This model is one of the best performing TF-IDF formula in the vector space model framework. The value of the parameter s is set to 0.05.
2. Lemur TF-IDF model: This model is another TF-IDF model that uses Robertson's tf and the standard idf. The parameter of this model is set to its default value, 0.75.
3. Classical Probabilistic model (BM25): BM25 is chosen as a state of the art representative of the classical probabilistic model. The main differences with this model and the previous model are that BM25 uses query term frequency in a different way and the idf also differs with the standard one. The parameters of this model is set to $k_1 = 1.2$, $b = 0.6$ and $k_3 = 1000$. Note that we found (on training data) slightly better results for $b = 0.6$ than the default 0.75.
4. Dirichlet smooth language model (LM): Language model is another probabilistic model that performs very effectively. For this model we set the value of Dirichlet smoothing (μ) to 1700.
5. Divergence from Randomness model (PL2): Finally, PL2 [12] represents the recently proposed non-parametric probabilistic model from divergence from randomness

[2] the code available at TREC million query page is used to compute stat AP

(DFR) family. Similar to the previous models, its performance also depends on a parameter value (c in the formula). We conduct experiments for this model by setting $c = 13$.

5. RESULTS

In this section we present the experimental results of our proposed work and compare them with the state of the art retrieval models. In Section 5.1 we compare the performance of the proposed model (MATF for Multi Aspect TF) with the two TF-IDF models, followed by the comparison with three probabilistic models- BM25, language model (LM) and PL2. We use three evaluation measures to evaluate the performance of all the methods.

5.1 Comparison with TF-IDF Models

In this section we focus on to compare the performance of the proposed model (MATF) with the Lemur TF-IDF and and Pivoted TF-IDF models. Table 2 presents the experimental results for six test collections measured in terms MAP, NDCG@20 and ERR@20.

First, we describe the results in terms of MAP. Table 2 clearly shows that MATF gains significantly better MAP than both of the TF-IDF models on two news collections. MATF performs 12% and 15.7% better than Lemur TF-IDF model on TREC-678 and ROBUST respectively. MATF is also significantly surpasses the Pivoted TF-IDF model on these collections with a margin of 8.8% and 6.3% respectively.

The behavior of MATF is similar when we see the results for two web collections, namely, WT10G and GOV2. Once again, MATF outperforms Lemur TF-IDF model by a margin of more than 20% in both of the occasions, which is clearly highly significant as confirmed by the paired t test. Like the previous two news collections, MATF maintains its superior behavior over Pivoted TF-IDF in these web collections also. In particular MATF gains more than 8% and 19% average precision than Pivoted TF-IDF for both of the collections and paired t test once again attests the significance.

We now turn to describe the results on two million query data sets. These two collections are particularly interesting, since they contain real search queries collected from a commercial search engine and also because of their variations in length. Table 2 once again demonstrates that MATF unequivocally outperforms the two TF-IDF models with significantly large margin. The MAP achieved by MATF is nearly 11% and 7% better than that achieved by Lemur TF-IDF on MQ-07 and MQ-08 collections respectively. Similarly, MATF surpasses the Pivoted TF-IDF by more than 10% margin on both of the occasions. Significance tests show that the performance differences are always statistically significant.

Among the two TF-IDF models, Lemur TF-IDF often seems to perform poorer than Pivoted (except MQ-08 where Lemur TF-IDF is nearly 4% better than pivoted). One potentially interesting outcome that we can see from Table 2 is that, when the document collection is larger MATF outperforms Pivoted TF-IDF with larger margin. In particular, MATF gains a MAP on GOV2 collection which is almost 20% better than the pivoted TF-IDF, which is a clear sign of effectiveness of MATF over the state of the TF-IDF models.

So far our discussion of experimental outcomes primarily confined on the basis of the binary relevance assessment. Note that five out of six test collections used in our evaluation have graded assessment in three scales (0,1,2). Therefore, we now turn to describe the results measured in terms of NDCG, that leverages the graded assessment.

The middle segment of Table 2 presents the results in terms of NDCG@20. It is once again clear that the performances are more or less consistent with MAP. Specifically, MATF surpasses the Lemur TF-IDF models with consistently and significantly large margin on all six collections and often the differences are higher or close to 10%, which once again clearly demonstrates the effectiveness of MATF. Performance of pivoted TF-IDF is once again very similar under the graded assessment and it achieves larger NDCG than Lemur TF-IDF except in one occasion. MATF once again is significantly better than the pivoted TF-IDF on all the collections and the differences are larger for larger web collections.

Our final comparison between the proposed model and the TF-IDF models focus on precision enhancing capabilities measured in terms of a metric ERR, that consider three things simultaneously: rank of the document, quality conveyed by the assessor assigned grade (non-relevant, relevant and highly relevant) and the quality of the documents that have been seen before the document of our focus.

The last segment of Table 2 reports the ERR@20 values achieved by the competing models on six collections. We can easily infer that MATF once again unanimously beats the two TF-IDF models. Only on WT10G, pivoted TF-IDF performs slightly better than MATF. Consistent with the previous measures, ERR@20 results demonstrate that on larger web collections the performance differences between MATF and the two TF-IDF models are larger.

Table 3: Comparison with TF-IDF models (statAP). Lemur means Lemur TF-IDF. Superscripts have their usual meaning.

	Lemur	pivot	MATF (% improv)
MQ-07	29.0	29.7	$\mathbf{34.4}^{lp}$ (18.2, 15.8)
MQ-08	28.4	27.6	$\mathbf{32.5}^{lp}$ (14.9, 17.8)

The performances of MATF and the two TF-IDF models on two million query data, measured by statistical average precision, are shown in Table 3. MATF transcends Lemur TF-IDF by a margin of 18% and 15% on MQ-07 and MQ-08 respectively, while it is better than pivoted TF-IDF with more than 15% on both of the collections.

In summary, based on the results shown in Table 2 and Table 3 we can infer that MATF outperforms two state of the art TF-IDF models with remarkable significance and consistency, and the performance differences are often noticeably large. The performance measured by three evaluation metrics unequivocally demonstrate that MATF is highly effective in ranked retrieval. Moreover, the results also show that MATF is more effective for larger web collections.

5.2 Comparison with Probabilistic Models

In the last section we compare the performance of our model with two TF-IDF models. In this section we compare the performance of MATF with three well known state of the

Table 2: Comparison with the TF-IDF models measured in terms of MAP, NDCG@20 and ERR@20. MATF denotes the proposed model. The best results are boldfaced. Superscripts l and p denote that the performance difference is statistically significant ($p < 0.05$) compared to Lemur TF-IDF and Pivot TF-IDF respectively.

Metric	Method	TREC-678	ROBUST	WT10G	GOV2	MQ-07	MQ-08
MAP	Lemur.TF-IDF	20.9	26.1	18.4	24.8	39.6	42.8
	Pivot.TF-IDF	21.5	28.4	20.5	26.5	40.0	41.2
	MATF	**23.4**^{lp}	**30.2**^{lp}	**22.2**^{lp}	**31.7**^{lp}	**44.2**^{lp}	**45.7**^{lp}
% better than Lemur.TF-IDF		12.0	15.7	20.7	27.8	11.6	6.8
% better than Pivot.TF-IDF		8.8	6.3	8.3	19.6	10.5	10.9
NDCG@20	Lemur.TF-IDF	40.0	37.5	31.6	43.8	46.8	50.1
	Pivot.TF-IDF	41.5	40.2	33.4	46.8	48.3	48.7
	MATF	**44.6**^{lp}	**41.5**^{lp}	**34.6**^{l}	**51.0**^{lp}	**51.1**^{lp}	**52.6**^{lp}
% better than Lemur.TF-IDF		11.5	10.7	9.5	16.4	9.2	5.0
% better than Pivot.TF-IDF		7.5	3.2	3.6	9.0	5.8	8.0
ERR@20	Lemur.TF-IDF	40.7	45.7	34.7	48.3	40.6	44.5
	Pivot.TF-IDF	41.9	46.3	**37.9**	49.4	42.9	44.6
	MATF	**43.9**^{lp}	**48.5**^{lp}	37.1^{l}	**53.4**^{lp}	**44.9**^{lp}	**47.3**^{lp}

art probabilistic retrieval models. Our evaluation strategy is once again similar to the previous section. We compare the performances of the models under MAP, NDCG@20 and ERR@20.

First we compare the performance of MATF with the BM25 model. Table 4 shows the summary of the retrieval results on six test collections. It is clear from the table that MATF is superior to BM25 model. This result holds for all the collections and for all three evaluation measure. When the performance differences between them are measured in terms of MAP, we notice that MATF is significantly effective for news as well as web corpora compared to BM25. In fact, MATF is nearly 10% better than BM25 on two news data, while on two web collections (WT10G and GOV2), MATF achieves 17% and 12% more MAP than BM25. The differences on MQ-07 and MQ-08 are similarly significant with substantial margins. The performance differences between MATF and BM25 revealed by NDCG metric are consistent with that revealed by MAP and once again, all the differences are statistically significant. ERR@20 depicts that for all the collections, MATF remains consistently superior to BM25, which clearly confirms that MATF is very effective for precision oriented systems.

We now compare the effectiveness of MATF and language model with Dirichlet prior language model. From Table 4 we clearly see that the performance differences between MATF and LM are larger in three out of six cases than that we had observed when comparing the performance of MATF and BM25. Specifically, MATF achieves close to or more than 10% MAP than LM on four out out six instances (except WT10G). The performance measured on graded relevance assessment also demonstrates that MATF unequivocally beats the Dirichlet prior language model based approach, and the differences are substantially large. On GOV2, MQ-07 and MQ-08 data, MATF surpasses LM with a margin of 14%, 8% and nearly 9% respectively. The comparison of precision enhancing abilities of MATF and LM also clearly indicates that MATF is always better than LM, which is very concordant with the experimental findings captured by MAP and NDCG.

We now compare the performance of the proposed model with another probabilistic model from the divergence from randomness family, namely, PL2. This model is relatively recent compared to the previous two probabilistic models and was also found to be better than BM25 in the experiments reported in [3]. Table 4 reflects two major facts. First, it appears from the table that PL2 is most effective among the probabilistic models and in particular only on MQ-08 data it performs worse than BM25 as reflected by both MAP and NDCG. The second major observation that can be made from Table 4 is that MATF beats this model also with harmonious consistency and performance differences are statistically significant on TREC-678, GOV2, MQ-07 and MQ-08 data. Similar to the previous outcomes, on web collections the performance differences between MATF and PL2 are larger than that for the news collections. Lastly, ERR metric depicts that MATF is better than PL2 across all six collection.

Table 5: Comparison with probabilistic models (statAP).

	BM25	LM	PL2	MATF (% improvement)
MQ-07	30.6	29.7	30.4	**34.4**^{blp} (12.8, 15.8, 13.2)
MQ-08	29.6	27.6	27.4	**32.5**^{blp} (10.5, 17.8, 18.6)

Table 5 compares the performance of four models for million query collections measured in terms of statistical average precision. It is once again clearly evident that MATF is consistently better than all three models and all the differences are very large and it is very consistent with the performance measured in terms other metrics presented in Table 4.

Overall, the comparative analysis clearly shows that MATF is the most effective retrieval model, which unequivocally outperforms all three probabilistic models, when the performances are measured in terms of MAP, NDCG and a precision biased metric, namely, ERR. Also, the relative per-

Table 4: Comparison with probabilistic models measured in terms of MAP, NDCG@20 and ERR@20. MATF denotes the proposed model. The best results are boldfaced. Superscripts b, l and p denote that the performance differences are statistically significant compared to BM25, LM and PL2 respectively.

Metric	Method	TREC-678	ROBUST	WT10G	GOV2	MQ-07	MQ-08
MAP	BM25	21.3	27.7	18.9	28.3	41.2	43.6
	LM	21.3	28.4	21.3	29.1	40.1	41.0
	PL2	22.7	29.5	21.3	29.7	40.9	41.5
	MATF	**23.4**^{blp}	**30.2**^{bl}	**22.2**^{b}	**31.7**^{blp}	**44.2**^{blp}	**45.7**^{blp}
% better than BM25		9.9	9.0	17.5	12.1	7.3	4.8
% better than LM		9.9	6.9	4.2	8.9	10.2	11.5
% better than PL2		3.1	2.4	4.2	6.7	8.1	10.1
NDCG@20	BM25	41.2	39.3	32.4	45.2	48.1	50.9
	LM	40.2	39.3	32.5	44.6	47.2	48.3
	PL2	42.9	41.1	33.1	46.1	48.0	49.0
	MATF	**44.6**^{blp}	**41.5**^{bl}	**34.6**^{bl}	**51.0**^{blp}	**51.1**^{blp}	**52.6**^{blp}
% better than BM25		8.3	5.6	6.8	12.8	6.2	3.3
% better than LM		10.9	5.6	6.5	14.3	8.3	8.9
% better than PL2		4.0	1.0	4.5	10.6	6.5	7.3
ERR@20	BM25	41.1	45.6	34.7	48.2	41.3	44.8
	LM	41.2	46.5	35.4	47.6	39.9	42.7
	PL2	43.0	47.0	35.2	47.7	40.7	42.7
	MATF	**43.9**^{blp}	**48.5**^{b}	**37.1**^{b}	**53.4**^{blp}	**44.9**^{blp}	**47.3**^{blp}

Table 6: Performance of two TF factors on short and long query. The values are MAP.

	short		long	
	MQ-07	MQ-08	MQ-07	MQ-08
LRTF	43.5	43.3	39.9	44.3
RITF	45.4	45.0	37.7	41.8

formance differences are often substantially large and the differences are even larger for the web collections that contain large number of queries. Among the three probabilistic models, PL2 and Dirichlet prior language model perform almost equally, with PL2 having a marginal edge over LM.

5.3 Analysis

In this section we analyze the effect of the two TF factors on short and long queries. For this analysis we choose the two million query collections, primarily because the collections have large number of queries. We divide the queries in two sets. The queries having at least 5 terms are denoted as short, while the rest of the queries (longer than 5 words) are treated as long. The main goal of this section is to validate the hypothesis made in the proposed section that relative intra-document based TF (RITF) performs better on short queries, while length regularized TF (RLTF) performs better on long queries.

Table 6 presents the experimental results on two million query data. The results seem to confirm our aforesaid assumption. LRTF always performs better than RITF on both of the collections, while RITF does better for short queries. However, the performance differences between the methods on longer queries are noticeably better than that for shorter queries.

6. CONCLUSION

In this paper, we present a novel TF-IDF term weighting scheme. The proposed term weighting scheme employs two aspects of within document term frequency normalization to determine the importance of a term. One component of the term frequency tends to prefer short documents, while the other tends to prefer long documents. We then combine these two TF components using the query length information, that maintains a balanced trade-off in retrieving short and long documents, when the ranking function faces queries of varying lengths.

Experiments carried out on a set of news and web collections show that the proposed model outperforms two well known state of the art TF-IDF baselines with significantly large margin, when measured in terms of MAP and NDCG. The model also surpasses three state of the art probabilistic models with remarkable significance almost always. Moreover, the proposed model is also significantly better than all of the five baselines in improving precision.

Acknowledgments

I would like to thank Dipasree Pal, Mandar Mitra and Swapan Parui for their comments, suggestions and help.

7. REFERENCES

[1] J. Allan, B. Carterette, J. A. Aslam, V. Pavlu, and E. Kanoulas. Million query track 2008 overview. In E. M. Voorhees and L. P. Buckland, editors, *The Sixteenth Text REtrieval Conference Proceedings (TREC 2008)*. National Institute of Standards and Technology, December 2009.

[2] J. Allan, B. Carterette, B. Dachev, J. A. Aslam, V. Pavlu, and E. Kanoulas. Million query track 2007 overview. In *TREC*, 2007.

[3] G. Amati and C. J. Van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, 20(4):357–389, Oct. 2002.

[4] S. Büttcher, C. L. A. Clarke, and I. Soboroff. The trec 2006 terabyte track. In *TREC*, 2006.

[5] B. Carterette, V. Pavlu, E. Kanoulas, J. A. Aslam, and J. Allan. Evaluation over thousands of queries. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 651–658, New York, NY, USA, 2008. ACM.

[6] O. Chapelle, D. Metlzer, Y. Zhang, and P. Grinspan. Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 621–630, New York, NY, USA, 2009. ACM.

[7] C. L. A. Clarke, N. Craswell, I. Soboroff, and E. M. Voorhees. Overview of the trec 2011 web track. In *TREC*, 2011.

[8] S. Clinchant and E. Gaussier. Retrieval constraints and word frequency distributions a log-logistic model for ir. *Inf. Retr.*, 14(1):5–25, Feb. 2011.

[9] 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*, CIKM '12, pages 2443–2446, New York, NY, USA, 2012. ACM.

[10] H. Fang, T. Tao, and C. Zhai. Diagnostic evaluation of information retrieval models. *ACM Trans. Inf. Syst.*, 29(2):7:1–7:42, Apr. 2011.

[11] W. R. Greiff. 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, pages 11–19, New York, NY, USA, 1998. ACM.

[12] B. He and I. Ounis. 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, pages 465–471, New York, NY, USA, 2005. ACM.

[13] B. He and I. Ounis. On setting the hyper-parameters of term frequency normalization for information retrieval. *ACM Trans. Inf. Syst.*, 25(3), July 2007.

[14] D. Hiemstra, S. Robertson, and H. Zaragoza. 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, pages 178–185, New York, NY, USA, 2004. ACM.

[15] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, Oct. 2002.

[16] K. S. Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: development and comparative experiments - part 1. *Inf. Process. Manage.*, 36(6):779–808, 2000.

[17] K. S. Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: development and comparative experiments - part 2. *Inf. Process. Manage.*, 36(6):809–840, 2000.

[18] 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.

[19] 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.

[20] S. Robertson and H. Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, Apr. 2009.

[21] S. E. Robertson. Readings in information retrieval. chapter The probability ranking principle in IR, pages 281–286. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

[22] 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.

[23] T. Sakai. Alternatives to bpref. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 71–78, New York, NY, USA, 2007. ACM.

[24] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, 24(5):513–523, Aug. 1988.

[25] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986.

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

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

[28] 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.

[29] K. Sparck Jones. Document retrieval systems. chapter A statistical interpretation of term specificity and its application in retrieval, pages 132–142. Taylor Graham Publishing, London, UK, UK, 1988.

[30] H. Turtle and W. B. Croft. Evaluation of an inference network-based retrieval model. *ACM Trans. Inf. Syst.*, 9(3):187–222, July 1991.

[31] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22(2):179–214, Apr. 2004.