A Standard Document Score for Information Retrieval

Ronan Cummins
Department of Computing and Information Systems
School of Computing and Mathematical Sciences
University of Greenwich, UK
r.p.cummins@greenwich.ac.uk

ABSTRACT

In this paper we propose a standard document retrieval score based on term-frequencies. We model the within-document term-frequency aspect of each term as a random variable. The standard score is then used to transform each random variable to a regularised form so that they can be effectively combined for use as a standard document score. The standardisation used imposes no constraints on the choice of probability distribution for the term-frequencies.

We show that the standardisation automatically creates a measure of term-specificity. Analysis shows that this measure is highly correlated with the traditional *idf* measure, and furthermore suggests a novel interpretation and justification of *idf*-like measures. With experiments on a number of different TREC collections, we show that the standard document score model is comparable with BM25. However, we show that an advantage of the standard document score model is that the document scores output from the model are dimensionless quantities, and therefore are comparable across different queries and collections in certain circumstances.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Retrieval models, Search Process*

Keywords

Standard Score, Normalisation

1. INTRODUCTION

The frequency of term-occurrences has long been used as a measure of the degree to which a document is *about* a specific term [6]. This idea has been the basis for automatic approaches to document retrieval and is still prevalent in most models of information retrieval (IR) [8, 7, 11]. In the highly effective BM25 function [8], the weight of the term-frequency component saturates, approaches an asymptote, as the actual term-frequency approaches infinity. The term-frequency component for a term t in a document D in the BM25 function (written x_D^t) is as follows:

$$x_D^t = \frac{(k_1 + 1) \cdot tf_t^D}{tf_t^D + k_1} \tag{1}$$

This notion of measuring *aboutness* using term-frequencies, along with that of *idf* (inverse document frequency), has been the basis of a number of retrieval functions over the years. More recently language modelling approaches [11] have become prominent and while they incorporate term-frequencies, many do not explicitly contain the *idf* measure[1]. An in-depth review of various attempts of reconciling *idf* with various theoretical approaches can be found in [9].

In this paper, we return to the idea of modelling term-frequencies as random variables [9]. Using a linear combination of standardised random variables, we develop what we call a **Standard Document Score** for IR. The structure of the paper is as follows: In Section 2, we define the probability space used, the **Standard Document Score** model for document retrieval, and present an analysis of the standardisation method used in our model. In Section 3 we present the results of experiments which compare the **Standard Document Score** model to that of BM25. Finally, Section 4 concludes the paper with a discussion and an outline of future work.

2. A STANDARD DOCUMENT SCORE

We propose modelling the non-linear term-frequency aspect (x_D^t) of each term as a random variable (X^t) on a probability space (Ω, $\mathcal{F}$, P). When modelling the term-frequency for a particular term as a random variable, the probability distribution (whichever one we choose) defines a probability measure (P). Furthermore, it seems natural that each of the N documents (the unit of retrieval) can be seen as one independent event from the event space $\mathcal{F}$, and the outcomes (i.e. term-frequency aspect) are part of the sample space (Ω) from $[0 : k_1 + 1)$. It is important to note that 0 is a valid outcome from one of the N events (documents). By including 0 as a valid outcome, this essentially means that each

ICTIR '13, September 29 - October 02 2013, Copenhagen, Denmark
Copyright 2013 ACM 978-1-4503-2107-5/13/09 ...$15.00.
http://dx.doi.org/10.1145/2499178.2499183

[1] We define *idf* as $log(N/df_t)$ where N is the number of documents in the collection and df_t is the number of documents in which term t appears. While the *idf* function in the original BM25 function is slightly different, studies have shown that there is little difference in effectiveness for many collections. In fact when stopword removal is not used, the *idf* in the original BM25 function performs poorly [4].

document can be thought of as a vector of length T, where T is the number of distinct terms in the collection. However, a problem arises from the fact that each document is actually of variable length and it is known that term-frequencies are affected by document length. In this paper we do not suggest a theoretically sound solution to this problem. We simply note that a useful heuristic is to modify the actual term-frequency using a document length normalisation[2] approach similar to that used in BM25. Furthermore, we do not suggest, or impose, any particular form for the probability measure P for the random variables X^t (suffice it to say that there are some restrictions on the particular form that they can take [1]).

2.1 Standardised Random Variables

Most retrieval functions aim to return a set of documents that are most about the query. This is typically achieved by aggregating some transformation of the term-frequency of the query-terms for each document in the collection, and then ranking those documents accordingly. For the following discussion we assume equal length documents and term-independence. We use the standard score (or z-score) which can be applied to any random variable, to render the random variables comparable. Given that these standard scores are comparable they are also easily combined in a composite score assuming term-independence. Therefore we use a composite of these standard scores to achieve a **Standard Document Score (SDS)** as follows:

$$SDS(Q, D) = \frac{1}{\sqrt{|Q|}} \sum_{t}^{Q} \frac{(x_D^t - E[X^t])}{\sigma(X^t)} \tag{2}$$

where $E[X^t]$ and $\sigma(X^t)$ are the expected value and the standard deviation of the random variable X^t respectively, and $|Q|$ is the query length. It should be noted that the random variables X^t for each term are linearly transformed under such a standardisation and do not lose their original shape (whatever it may be). After standardisation each random variable will have a mean of 0 and a standard deviation of 1 (and also a unit variance). The summation of the standardised random variables implies that the mean **Standard Document Score (SDS)** $SDS(Q, D)$ will also be 0. The standard deviation of the final score (**SDS**) is also 1 because of the $\sqrt{|Q|}$ normalisation that is applied. This follows as the variance of the sum of any $|Q|$ random variables is equal to the sum of the individual $|Q|$ variances when each random variable is independent. Due to the standardisation process, we have unit variances and therefore, the sum of the individual variances is $|Q|$. The standard deviation is therefore $\sqrt{|Q|}$, which is our query normalisation factor. Although the query normalisation of $\sqrt{|Q|}$ does not affect the ranking of documents, it renders the **Standard Document Score (SDS)** a dimensionless quantity and comparable across queries and collections. Another interesting observation of this retrieval function is that the average score for the documents in a collection is zero. The **Standard Document Score** can be interpreted as the number of standard deviations a document is from the average document score for a specific query.

As we do not know the underlying distribution for the term-frequencies, technically we can only estimate $E[X^t]$ and $\sigma(X^t)$. They can be estimated using the mean and standard deviation of X^t over the N documents for each term[3]. For the $\sum_t^Q$ in (2) we adopt a *bag semantic* view and therefore if a term appears multiple times in a query, its successive occurrences are treated as distinct terms and they are aggregated in the usual manner (as if it was a new term). This is equivalent to using a linear within-query term-frequency function as the query term-weights.

2.2 Analysis and Binary Simplification

For the analysis that follows, we assume that all documents are of equal length and therefore we do not use any notion of document length normalisation in our random variables X (i.e. $b = 0$). Therefore, the random variable X^t models only the non-linear term-frequency of each term t. From equation 2, we can see that the weight of a term t appearing in a document D with a term-frequency of 1, will be $(1 - E[X^t])/\sigma(X^t)$. Similarly, in the BM25 function the weight of the same term t, appearing in a document D will be $idf(t)$. Therefore, when there is a single term occurrence of t in a document D (i.e. $x_D^t = 1$), the expression for the new measure of term-specificity can be written in terms of the variance of the random variable X^t as follows:

$$\frac{1 - E[X^t]}{\sigma(X^t)} = (1 - E[X^t]) \cdot \sqrt{\frac{1}{E[(X^t)^2] - (E[X^t])^2}} \tag{3}$$

Furthermore, if $k_1 = 0.0$, then X^t models a binary random variable for term-frequency. As a result of this simplified binary representation, we can estimate $E[(X^t)^2] = df_t/N$ and $E[X^t]^2 = df_t^2/N^2$. Therefore in this binary case the measure of term-specificity simplifies to:

$$\frac{1 - E[X^t]}{\sigma(X^t)} = (1 - \frac{df_t}{N}) \cdot \sqrt{\frac{N}{df_t - (df_t^2/N)}} \tag{4}$$

for the single occurrence of a term t in a document D. Figure 1 shows the plot of $(1 - E[X^t])/\sigma(X^t)$ (in green), $(0 - E[X^t])/\sigma(X^t)$ (in blue), and *idf* (in red). We can see that new measure of term-specificity introduced by the standardisation of X^t is very similar in shape to *idf*. Furthermore we note that when a term t does not occur in document D there is a penalisation of $E[X^t]/\sigma(X^t)$ (the blue curve).

Table 1 shows the correlation of *idf* and the term-weight of $(1 - E[X^t])/\sigma(X^t)$ in the **Standard Document Score** for actual terms in a number of test collections. We report the linear correlation for two values of k_1. When $k_1 = 0.0$ the term-frequency aspect becomes a binary indication of whether a term occurs in a document, and when $k_1 = 1.2$ the actual term-frequency is taken into consideration. We can see that there is a strong linear correlation between the two measures of term-specificity. As per our analysis, when a binary term-frequency aspect is used (ie. $k_1 = 0.0$), there is a Kendall-Tau correlation of 1.0 between the two types of term-specificity. This is because they are monotonically related as per Figure 1.

[2] For the experiment reported in Section 3, we normalised the actual term-frequency tf_t^D in (1) similarly to BM25.

[3] If we were to forego our view of the term-frequency as a random variable, we could then view our N documents as the entire population. The expected value and variance of the population are then known quantities.

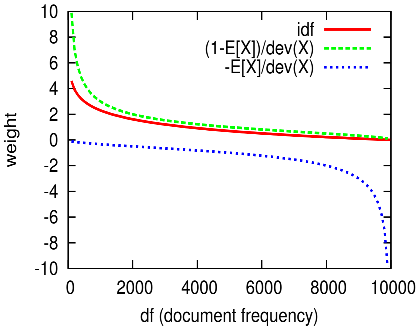

Figure 1: weight of *idf*, $(1 - E[X^t])/\sigma(X^t)$, and $(0 - E[X^t])/\sigma(X^t)$ when $k_1 = 0$ for different document frequencies where $N = 10,000$

Table 1: Linear Correlation (Kendall-Tau in parenthesis) of $(1 - E[X^t])/\sigma(X^t)$ and *idf* for terms in the title fields of topics for two different values of k_1

Collection	Topics	# Terms	$k_1 = 0.0$	$k_1 = 1.2$
FT	251-450	469	0.752 (1.0)	0.730 (0.961)
LA	301-450	334	0.794 (1.0)	0.782 (0.967)
WT2G	401-450	118	0.777 (1.0)	0.733 (0.957)
WT10G	451-500	116	0.765 (1.0)	0.756 (0.969)

We end this section by outlining a simplified version of the **Standard Document Score (SDS)** that uses a binary weighting only for the measure of term-specificity. The simplified binary version (BSDS) of the **Standard Document Score** retrieval function is as follows:

$$BSDS(Q, D) = \frac{1}{\sqrt{|Q|}} \sum_{t}^{Q} \frac{(x_D^t - \frac{df_t}{N})}{\sqrt{(df_t - (df_t^2/N))/N}} \qquad (5)$$

3. EXPERIMENTS

We undertake two experiments that evaluate SDS and BSDS. Firstly, we evaluate the retrieval effectiveness of SDS and BSDS against BM25 in terms of mean average precision (MAP) and precision-at-10 (P@10). We set $k_1 = 1.2$ in all three retrieval functions. We compare SDS and BSDS against BM25 when no normalisation is applied (i.e. $b = 0$) and compare these retrieval functions when incorporating document length normalisation (i.e. $b > 0.0$)[4]. Details of the TREC documents and query sets used in this evaluation are presented in Table 2. Stemming and stop-word removal was performed on all collections and queries. For the second experiment, we evaluate the comparability of the scores output by SDS and BM25 across queries and collections using measure of linear correlation.

3.1 BM25 vs SDS

Table 2 shows the effectiveness of the retrieval approaches in terms of MAP and P@10 for a number of scenarios. Firstly, when ignoring document length normalisation (i.e. $b = 0$), both SDS and BSDS significantly outperform BM25 in terms of MAP for medium and long queries on many of the collections. The results for short queries on all the collections for all retrieval methods are quite similar. When documents are of similar length (i.e. low deviation of document length in the collection), there tends to be a performance increase when using the SDS.

Table 2 shows also shows the MAP and P@10 for the new retrieval functions (SDS and BSDS with $b = 0.4$) and for the tuned BM25 function on the test collections. When document length normalisation is introduced (i.e. $b \neq 0$), we can see that in general BM25 outperforms both SDS and BSDS in terms of MAP for short queries, although the results are not significant. For both medium and long queries, both SDS and BSDS tend to outperform BM25 in terms of MAP, although again the results tend not to be significant.

3.2 Evaluating the Comparability of Scores

To measure the comparability of scores across queries and collections, we measure the correlation between the average precision of the query and the top document score returned for both BM25 and SDS for the query. The document scores returned for a particular query have been used to predict the effectiveness of the ranking returned for a particular query in a number of previous studies [10, 2]. Table 3 shows the results of these experiments. In general we can see that the correlation coefficient is higher for SDS than it is for BM25. This suggests that the document scores of SDS are more comparable across different queries. We also measured the correlation of average precision and the top document score across varying query lengths on the same collection, and across different collections for similar length queries. All these scenarios are outlined in Table 3. However, on some of the Web collections the BM25 function has a higher correlation coefficient. More research is needed to ascertain the reason for this. One possible reason is that the lengths of documents are more varied in these Web collections, and this may distort the SDS standardisation method unduly.

4. DISCUSSION AND CONCLUSION

These results and analysis would seem to suggest that *idf* has been playing a standardisation role in $tf \cdot idf$ type schemes. It would seem in fact that *idf* is inversely related to the uncertainty (in terms of variance) of *aboutness* when modelled as a random variable. Furthermore, it is interesting that a number of performance prediction approaches [12, 3] use the square-root of the query length to normalise document scores for use in query performance prediction. In the SDS model proposed here, it is shown that $\sqrt{|Q|}$ arises naturally. Given that each document score in our model can be viewed as the number of standard deviations from an average document score for a query, the scoring process can also be thought of as a method of *outlier detection* and ultimately related to work in score distributions [1].

It would be interesting to apply this retrieval function to collections of items that do not vary much in length, such as microblog retrieval [5]. Furthermore, future work could explore other non-linear term-frequency transformations.

[4] We use the default value of $b = 0.75$ and also tuned the BM25 (labelled BM25$_b$) normalisation parameter (the optimum of $b = \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$) for each set of queries on each collection. For the SDS and BSDS functions, we included BM25's document length normalisation prior to calculating the collection statistics for all experiments. We set $b = 0.4$ for both SDS and BSDS. We found this to be a somewhat robust setting across different query lengths on our test collections.

Table 2: MAP (P@10 in parenthesis) on test collections for all retrieval functions. Best MAP is in bold. Statistical significance of MAP is measured against the tuned $BM25_b$ function at the $p < 0.05$ level using a paired two-sided t-test and denoted †.

	Gov.	Newswire			Web		
Collections	FR	LA	FT	FBIS	TREC8	TREC9	TREC01
# Documents	55630	131896	210158	130471	221066	1692096	1692096
avg. len	333	223	190	240	623	263	263
stdev len	508	116	173	459	1472	772	772
Topic Range	251-450	301-450	251-450	301-450	401-450	451-500	551-550
# Topics	91	144	188	116	50	50	50
short queries (title only)							
$BM25_{b=0}$	**0.266** (0.17)	**0.211** (0.26)	**0.242** (0.26)	**0.234** (0.27)	**0.250** (0.38)	**0.148** (0.20)	0.117 (0.17)
$SDS_{b=0}$	0.262 (0.18)	0.209 (0.25)	0.235 (0.25)	0.220 (0.26)	0.242 (0.35)	0.145 (0.17)	**0.135** (0.20)
BM25	0.286 (0.17)	0.207 (0.26)	0.231 (0.30)	0.229 (0.26)	0.226 (0.37)	0.157 (0.22)	0.157 (0.29)
$BM25_b$	**0.287** (0.19)	**0.222** (0.27)	**0.249** (0.30)	**0.245** (0.29)	**0.279** (0.44)	**0.181** (0.25)	0.171 (0.31)
SDS	0.283 (0.20)	0.218 (0.25)	0.241 (0.30)	0.228 (0.27)	0.255 (0.42)	0.178 (0.22)	0.174 (0.31)
BSDS	0.284 (0.20)	0.218 (0.26)	0.244 (0.29)	0.229 (0.27)	0.250 (0.44)	0.177 (0.22)	**0.176** (0.31)
medium queries (description only)							
$BM25_{b=0}$	0.187 (0.12)	0.145 (0.20)	0.194 (0.23)	0.153 (0.15)	0.183 (0.30)	0.061 (0.13)	0.061 (0.13)
$SDS_{b=0}$	**0.233** (0.16)	**0.187** (0.21)	**0.219** (0.24)	**0.209**† (0.20)	**0.222** (0.31)	**0.095** (0.14)	**0.082** (0.11)
BM25	0.268 (0.17)	0.184 (0.24)	0.204 (0.27)	0.220 (0.27)	0.225 (0.39)	0.168 (0.28)	0.135 (0.32)
$BM25_b$	**0.274** (0.17)	0.191 (0.25)	0.219 (0.25)	0.223 (0.28)	0.252 (0.41)	0.182 (0.34)	**0.159** (0.33)
SDS	0.268 (0.18)	0.206 (0.24)	0.235 (0.26)	0.233 (0.26)	**0.255** (0.37)	0.180 (0.28)	0.132 (0.24)
BSDS	0.270 (0.18)	**0.211** (0.24)	**0.244** (0.27)	**0.238** (0.27)	0.250 (0.37)	**0.183** (0.29)	0.138 (0.24)
long queries (title, description, and narrative fields)							
$BM25_{b=0}$	0.169 (0.12)	0.165 (0.24)	0.202 (0.25)	0.113 (0.12)	0.126 (0.20)	0.063 (0.12)	0.057 (0.12)
$SDS_{b=0}$	**0.256**† (0.18)	**0.234**† (0.26)	**0.254**† (0.27)	**0.208**† (0.21)	**0.197**† (0.27)	**0.112**† (0.16)	**0.104** (0.14)
BM25	0.326 (0.21)	0.255 (0.31)	0.258 (0.33)	0.277 (0.33)	0.267 (0.44)	0.220 (0.37)	0.208 (0.39)
$BM25_b$	0.328 (0.22)	0.256 (0.31)	0.257 (0.32)	0.282 (0.34)	0.275 (0.42)	**0.225** (0.38)	**0.209** (0.40)
SDS	0.322 (0.23)	0.272 (0.30)	0.281 (0.30)	0.281 (0.30)	0.279 (0.38)	0.189 (0.26)	0.192 (0.33)
BSDS	**0.334** (0.23)	**0.276** (0.31)	**0.291** (0.32)	**0.286** (0.32)	**0.281** (0.38)	0.196 (0.28)	0.201 (0.33)

Table 3: Linear correlation (+/- 95% confidence interval) between average precision and the top document score for BM25 and SDS on a number of test collections.

	Gov.	Newswire			Web			All
Cols	FR	LA	FT	FBIS	TREC8	TREC9	TREC01	Pooled
short queries (title field)								
BM25	0.175 (0.167)	0.291 (0.158)	0.330 (0.112)	0.264 (0.175)	**0.359** (0.284)	**0.367** (0.281)	**0.544** (0.202)	0.260 (0.081)
SDS	**0.507** (0.139)	**0.471** (0.117)	**0.425** (0.110)	**0.477** (0.130)	0.218 (0.281)	-0.054 (0.271)	0.516 (0.178)	**0.295** (0.060)
medium queries (description field)								
BM25	0.085 (0.213)	0.097 (0.160)	0.299 (0.114)	0.182 (0.179)	0.240 (0.281)	**0.225** (0.275)	**0.435** (0.213)	0.182 (0.072)
SDS	**0.593** (0.119)	**0.438** (0.126)	**0.303** (0.124)	**0.385** (0.150)	**0.269** (0.241)	0.017 (0.282)	-0.093 (0.270)	**0.188** (0.080)
long queries (title, description, and narrative fields)								
BM25	0.065 (0.201)	0.380 (0.142)	0.318 (0.120)	0.233 (0.181)	0.175 (0.276)	0.313 (0.281)	0.450 (0.291)	0.258 (0.071)
SDS	**0.560** (0.127)	**0.470** (0.119)	**0.404** (0.114)	**0.551** (0.121)	**0.275** (0.241)	0.037 (0.278)	-0.054 (0.270)	**0.321** (0.068)
All queries lengths pooled								
BM25	0.056 (0.121)	0.143 (0.097)	0.157 (0.082)	0.077 (0.099)	0.002 (0.150)	0.193 (0.160)	0.252 (0.160)	0.113 (0.043)
SDS	**0.535** (0.079)	**0.462** (0.075)	**0.384** (0.068)	**0.477** (0.065)	**0.265** (0.159)	0.038 (0.280)	0.025 (0.156)	**0.320** (0.060)

5. REFERENCES

[1] Ronan Cummins. Measuring the ability of score distributions to model relevance. In *AIRS*, pages 25–36, 2011.

[2] Ronan Cummins. Predicting query performance directly from score distributions. In *AIRS*, pages 315–326, 2011.

[3] Ronan Cummins, Joemon Jose, and Colm O'Riordan. Improved query performance prediction using standard deviation. In *SIGIR 2011*, pages 1089–1090, New York, NY, USA, 2011. ACM.

[4] Hui Fang and ChengXiang Zhai. An exploration of axiomatic approaches to information retrieval. In *SIGIR '05*, pages 480–487. ACM Press, 2005.

[5] Paul Ferguson, Neil O'Hare, James Lanagan, Owen Phelan, and Kevin McCarthy. An investigation of term weighting approaches for microblog retrieval. In *ECIR'12*, pages 552–555, Berlin, Heidelberg, 2012. Springer-Verlag.

[6] H. P. Luhn. The automatic creation of literature abstracts. *IBM J. Res. Dev.*, 2(2):159–165, April 1958.

[7] Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *SIGIR '98*, pages 275–281, New York, NY, USA, 1998. ACM.

[8] S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR 1994*, SIGIR '94, pages 232–241, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[9] Stephen Robertson. Understanding inverse document frequency: On theoretical arguments for idf. *Journal of Documentation*, 60:2004, 2004.

[10] Anna Shtok, Oren Kurland, and David Carmel. Predicting query performance by query-drift estimation. In *ICTIR*, pages 305–312, 2009.

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

[12] Yun Zhou and W. Bruce Croft. Query performance prediction in web search environments. In *SIGIR 2007*, pages 543–550, New York, NY, USA, 2007. ACM.