Tie Breaker: A Novel Way of Combining Retrieval Signals

Hao Wu
Department of Electrical and Computer Engineering
University of Delaware
Newark, DE USA
haow@udel.edu

Hui Fang
Department of Electrical and Computer Engineering
University of Delaware
Newark, DE USA
hfang@udel.edu

ABSTRACT

Empirical studies of information retrieval suggest that the effectiveness of a retrieval function is closely related to how it combines multiple retrieval signals including term frequency, inverse document frequency and document length. Although it is relatively easy to capture how each signal contributes to the relevance scores, it is more challenging to find the best way of combining these signals since they often interact with each other in a complicated way. As a result, when deriving a retrieval function from traditional retrieval models, the choice of one implementation over the others was often made based on empirical observations rather than sound theoretical derivations.

In this paper, we propose a novel way of combining retrieval signals to derive robust retrieval functions. Instead of seeking an integrated way of combining these signals into a complex mathematical retrieval function, our main idea is to prioritize the retrieval signals, apply the strongest signal first to rank documents, and then iteratively use the weaker signals to break the ties of the documents with the same scores. One unique advantage of our method is that it eliminates the need of having complicated implementation of the signals and enables a simple yet elegant way of combining the multiple signals for document ranking. Empirical results show that the proposed method can achieve comparable performance as the state of art retrieval functions over traditional TREC ad hoc retrieval collections, and can outperform them over TREC microblog retrieval collections.

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

General Terms: Algorithms, Experimentation

Keywords: tie-breaking, prediction

1. INTRODUCTION

Developing effective retrieval functions has been a long-standing challenge in the field of Information Retrieval (IR).

ICTIR '13, September 29 - October 02 2013, Copenhagen, Denmark

http://dx.doi.org/10.1145/2499178.2499190

Various models have been proposed and studied, such as vector space models [9, 11], classical probabilistic models [6, 7], and language models [5, 12]. Despite their differences, all the models leverage multiple retrieval signals including term frequency (TF), inverse document frequency (IDF) and document length (DL).

Previous studies have shown that the effectiveness of a retrieval function is closely related to how it implements and combines these signals [2]. Motivated by this observation, axiomatic approaches to IR was proposed to formalize the implementation of these signals through retrieval constraints and then derive retrieval functions that can satisfy all the constraints. The axiomatic approaches has been shown to be useful to develop more effective retrieval functions [1, 3].

Although it is relatively easy to define retrieval constraints based on desirable properties of a single retrieval signal, it is rather challenging to formalize constraints that balance the interactions among multiple ones. A few constraints (i.e., TF-LNC [2] and LBs [4]) have been defined to capture the desirable interaction between TF and DL, but no constraints have been defined to capture other interactions such as the one between TF and IDF. Moreover, due to the complexity of the interaction, the defined constraints are often based on very specific assumptions (e.g., a query only has a single term). As a result, these constraints might not generalize well to the scenarios violating the assumptions.

Moreover, existing studies on axiomatic approaches rely on traditional retrieval functions to derive more effective retrieval functions. The commonly used strategy is to identify constraints that an existing retrieval function fails to satify and then revise the function accordingly to satisfy more constraints [2–4]. It remains unclear how to search for a new retrieval function that satisfy all the constraints from the scratch.

In this paper, we propose a novel way of combining multiple retrieval signals based on *tie breaking*. The basic idea is to calculate the *strength* of individual retrieval signal, prioritize them based on their strength, and apply a multi-step tie-breaking method to rank documents. More specifically, we first use the strongest signal to rank documents. To break the ties of all the documents with the same scores, we then apply the next strongest signal to rank them. This process is repeated until all the signals are used or no more ties in the ranking list. For example, we could first use IDF to rank documents and then for all the documents with the same scores based on such a IDF weighting, we would break the tied documents by ranking them based on TF. Extensive experiments are conducted to evaluate the proposed

tie-breaking based retrieval strategy. Results show that the proposed strategy is as effective as the state of the art retrieval functions over TREC ad hoc search collections and are more effective over TREC Microblog collections. Moreover, our study reviews that it is not necessary to use the complicated implementation of the retrieval signals in the proposed method.

2. TIE BREAKING BASED DOCUMENT RANKING

2.1 Motivation

Almost all of the existing retrieval functions are implemented by combining multiple retrieval signals such as TF, IDF and DL, but their implementations vary a lot and some of them are quite complicated [8,11,12]. These retrieval signals are then often combined by taking the multiplication or summation.

The choice of the complicated implemenation is often based on empirical results rather than the justifications based on the sound theoretic framework. For example, why do the IDF implementations always have a log? The intuition of IDF weighting is well justified, i.e., we want to distinguish important terms from common ones based on the document frequency of the terms. But why not use $\frac{1}{df(t)}$ directly?

In a way, these complicated implementations of retrieval signals were caused by the requirement of having an integrated mathematical function to rank documents. Since the interaction among multiple signals could be quite complex [4], the combination requires the balances among these signals, which were often achieved by heuristic modifications of the implementations.

However, is it necessary to have a single mathematical function to rank documents? Are there other simple yet effective strategies that can combine multiple retrieval signals? These are the research questions we plan to address in the paper.

2.2 The Basic Idea

In this paper, we propose a novel multi-level tie-breaking based strategy to rank documents. The main idea is to combine multiple signals in a multi-step ranking process. Specifically, we will first apply a single retrieval signal to rank documents. In the cases when multiple documents have the same scores, the second signal will be used to break the ties. We will keep applying the next retrieval signal until there is no more ties or no more signals to be applied. It is clear that this process could be considered as a way of ranking documents.

Formally, we use $A \bigoplus B$ to denote the tie-breaking document ranking strategy, where we use B to break the ties of the results generated by using A.

2.3 A Working Example

Let us describe how a specific tie-breaking method (i.e., $IDF \bigoplus TF$) works through a working example.

Consider a document collection with 7 documents, i.e., $\{d_1, d_2^*, d_3^*, d_4^*, d_5, d_6, d_7\}$. Given a query, d_2^*, d_3^* and d_4^* are relevant while the other are not relevant. Suppose the relevance scores computed using an IDF weighting strategy are "1,3,2,5,2,3,4" respectively, which means that d_2^* and d_6 receive the same scores and so do d_3^* and d_5. The ranked list of documents can then be considered as a ranked list of the 5 blocks and documents within each block has the same score:

B_1	B_2	B_3	B_4	B_5
d_4^*	d_7	d_2^*,d_6	d_3^*,d_5	d_1

Note that B_i is ranked higher than B_{i+1}. It is clear that the performance of this search result is determined by how we break the ties within each block. There could be four possible ways of breaking the ties, and their performance measured with average precision (AP) is shown as follows:

$d_4^*, d_7, d_2^*, d_6, d_3^*, d_5, d_1$	$AP = \frac{1}{3} \times (\frac{1}{1} + \frac{2}{3} + \frac{3}{5}) = 0.756$
$d_4^*, d_7, d_6, d_2^*, d_3^*, d_5, d_1$	$AP = \frac{1}{3} \times (\frac{1}{1} + \frac{2}{4} + \frac{3}{5}) = 0.700$
$d_4^*, d_7, d_2^*, d_6, d_5, d_3^*, d_1$	$AP = \frac{1}{3} \times (\frac{1}{1} + \frac{2}{3} + \frac{3}{6}) = 0.722$
$d_4^*, d_7, d_6, d_2^*, d_5, d_3^*, d_1$	$AP = \frac{1}{3} \times (\frac{1}{1} + \frac{2}{4} + \frac{3}{6}) = 0.667$

Thus, it is clear that the performance range of IDF is from 0.667 to 0.756, and its expected performance is 0.711. Note that when evaluating the result with an existing IR evaluation script such as trec-eval, the performance could be either one of the four values since the ties would be broken randomly.

In the next step, we could try to use the TF weighting to break the ties within each block. For example, if the score of d_6 is higher than d_2^* based on TF and the scores of d_3^* and d_5 are the same, we would get the following ranked search results:

B_1	B_2	B_3	B_4	B_5	B_6
d_4^*	d_7	d_2^*	d_6	d_3^*,d_5	d_1

There is still a tie in block B_5. There are two ways of breaking the tie, and their performance is:

$d_4^*, d_7, d_2^*, d_6, d_3^*, d_5, d_1$	$AP = \frac{1}{3} \times (\frac{1}{1} + \frac{2}{3} + \frac{3}{5}) = 0.756$
$d_4^*, d_7, d_2^*, d_6, d_5, d_3^*, d_1$	$AP = \frac{1}{3} \times (\frac{1}{1} + \frac{2}{3} + \frac{3}{6}) = 0.722$

The performance range of $IDF \bigoplus TF$ is 0.722 to 0.756 with an expected performance 0.739. Comparing this performance with the one using only IDF, we can see that it has a smaller performance range and a better expected performance.

2.4 Properties

We now discuss a few properties of the proposed tie-breaking method.

- The performance of $A \bigoplus B$ is closely related to that of using only A for ranking. In particular, the performance range of $A \bigoplus B$ is a subset of the range when using only A. The best performance of $A \bigoplus B$ will not be better than the best performance of A, and the worst performance of $A \bigoplus B$ would not be worse than the worst performance of A.
- The expected performance of $A \bigoplus B$ could be better than that of A when B is a reasonable retrieval signal, and would be worse otherwise.
- The performance of $A \bigoplus B$ might be the same as that of A when B and A captures the similar retrieval signals.

Based on these properties, it is clear that the performance of the tie-breaking based ranking strategy is closely related to how to choose the signals and how to prioritize them.

3. EXPERIMENTS

3.1 Prioritizing Retrieval Signals

We leverage the most commonly used retrieval signals: term frequency (TF), inverse document frequency (IDF) and

Table 1: Performance Range of TF signals (MAP@1000)

	MAP_B	MAP_W	MAP_E	MAP_R
$c(t,D)$	0.0612	0.0455	0.0509	0.0504
$1+log(c(t,D))$	0.1332	0.1196	0.1242	0.1240
$1+log(1+log(c(t,D)))$	0.1769	0.1614	0.1668	0.1666
$\frac{c(t,D)}{c(t,D)+1}$	**0.1917**	**0.1733**	**0.1800**	**0.1799**

Table 2: Performance Range of IDF signals (MAP@1000)

	MAP_B	MAP_W	MAP_E	MAP_R
$\frac{1}{df(t)}$	0.4652	0.0820	0.1343	0.1406
$log(1+\frac{N}{df(t)})$	0.4646	0.0817	0.1336	0.1396
$log(1+\frac{maxdf}{df(t)})$	0.4646	0.0817	0.1336	0.1396
$log\frac{N-df(t)}{df(t)}$	0.4567	0.0817	0.1335	0.1394

document length (DL). To prioritize the signals, we propose to estimate their strength based on their performance range and the expected MAP value. In particular, signal A is stronger than B under the following two conditions: (1) the best performance of A is higher than that of B; and (2) the expected performance of A is better than that of B. The first condition is more important when we choose which signal should be applied first.

For each signal, we report its performance range as well as the expected MAP value. MAP_B denotes the best performance among all the possible document rankings, MAP_W denotes the worst performance, and MAP_E denotes the expected performance. We also report MAP_R, i.e., the performance of the ranking list generated by trec-eval, which arbitrarily breaks the tied documents. Moreover, for each signal, multiple implementations are considered. The experiments are conducted using the robust04 collection, which is the collection used for the TREC 2004 robust track.

The TF signal should be implemented in a way to satisfy TFCs constraints defined in the previous study [2]. In particular, it should favour documents that contain more occurrences of the query terms and should also favour documents that cover more distinct query terms. Table 1 shows the performance range of different implementations of the TF signal. It is clear that $\frac{c(t,D)}{c(t,D)+1}$ is the best choice since it has the highest MAP_B and MAP_E values. Note that the constraint analysis in the previous work can tell us the first TF implementation in the table is worse than the others because it violates more TF constraints, but it can not distinguish the last three implementations as what we can do in this paper.

The IDF signal should assign higher weights to the less frequent terms in the collection. Table 2 shows the performance range of different implementations of the IDF signal. It is clear that the performance of all these implementations are similar. Since $\frac{1}{df(t)}$ has the simplest form, we will choose this one over the other implementations for IDF signal. Moreover, if we compare the results with the ones in Table 1, we see that the IDF signals have larger performance range and better MAP_B values, which suggest that the IDF signal should be applied before the TF signal.

The DL signal should penalize long documents, and in the mean time avoid over-penalizing them. Table 3 shows the performance range of different representative implementations of the DL signal. It is clear that the performance of

Table 3: Performance Range of DL signals (MAP@1000)

	MAP_B	MAP_W	MAP_E	MAP_R
$\frac{1}{\|D\|}$	0.0219	0.0195	0.0196	0.0196
$\frac{1}{1-s+s\frac{\|D\|}{avdl}}(s=0.2)$	0.0219	0.0195	0.0196	0.0196
$log\frac{\mu}{\|D\|+\mu}(\mu=2000)$	0.0219	0.0195	0.0196	0.0196

Table 4: Performance of different combination of signals (MAP@1000)

	MAP_B	MAP_W	MAP_E	MAP_R
$IDF \bigoplus TF \bigoplus DL$	**0.2093**	**0.2070**	**0.2070**	**0.2071**
$IDF \bigoplus DL \bigoplus TF$	0.1700	0.1676	0.1677	0.1677
$TF \bigoplus IDF \bigoplus DL$	0.1840	0.1817	0.1818	0.1818
$TF \bigoplus DL \bigoplus IDF$	0.1836	0.1813	0.1813	0.1813
$DL \bigoplus IDF \bigoplus TF$	0.0219	0.0196	0.0196	0.0196
$DL \bigoplus TF \bigoplus IDF$	0.0219	0.0196	0.0196	0.0196

all these implementations are equally bad, which suggests that the DL signal is not as effective as the other two and might need to be applied after them. Since $\frac{1}{|D|}$ has the simplest implementation, we choose this one over others in our experiments.

3.2 Combining Retrieval Signals

According to the properties described in Section 2.4, it is unnecessary to combine different implementations of the same signal because such combinations would unlikely change the performance. For example, the performance of $\frac{1}{df(t)} \bigoplus log\frac{1}{df(t)}$ is similar to that of $\frac{1}{df(t)}$ since the second signal could not break the ties caused by the first signal. Thus, we will apply one implementation from each of the three signals and then combine them using the multi-level tie-breaking strategy. Based on our discussions in the previous subsection, our hypothesis is that $\frac{1}{df(t)} \bigoplus \frac{c(t,D)}{c(t,D)+1} \bigoplus \frac{1}{|D|}$ is the best choice. We now conduct experiments to verify the above hypothesis.

The first experiment is designed to test whether applying the signal with the highest MAP_B is the best choice. Specifically, given the implementation of the three signals, we will use different order to combine them and see whether the optimal order is consistent with the one ranked based on the decreasing order of MAP_B. We use the best strategies described earlier for each signal. The performance comparison is shown in Table 4. It is clear that $IDF \bigoplus TF \bigoplus DL$ gives the best performance, which means that it is reasonable to apply the signals with highest MAP_B first.

Moreover, it is interesting to mention that the performance of the level-by-level is definitely within the range of that generated by the first level signal alone. This observation matches what we described in Section 2.4. It also explains the reason why we should apply the signals with highest MAP_B first: high MAP_B simply leaves us enough space for following signals to further improve the performance.

The second experiment is designed to test whether MAP_E of an individual signal is a good indicator of the effectiveness after the tie-breaking. Table 5 shows the performance when using different TF signals. It is clear that the last implementation is the best, which is consistent with our observation based on Table 1. Moreover, Table 6 shows the performance when using different IDF signals. The performance differ-

Table 5: Performance comparison of different TF signals (MAP_E@1000)

combinations	MAP_E
$\frac{1}{df(t)} \bigoplus c(t,D)$	0.1841
$\frac{1}{df(t)} \bigoplus 1 + log(c(t,D))$	0.1988
$\frac{1}{df(t)} \bigoplus 1 + log(1 + log(c(t,D)))$	0.2038
$\frac{1}{df(t)} \bigoplus \frac{c(t,D)}{c(t,D)+1}$	**0.2052**

Table 6: Performance comparison of different IDF forms (MAP_E@1000)

combinations	MAP_E
$\frac{1}{df(t)} \bigoplus \frac{c(t,D)}{c(t,D)+1}$	**0.2052**
$log(1 + \frac{N}{df(t)}) \bigoplus \frac{c(t,D)}{c(t,D)+1}$	0.2047
$log(1 + \frac{maxdf}{df(t)}) \bigoplus \frac{c(t,D)}{c(t,D)+1}$	0.2047
$log \frac{N-df(t)}{df(t)} \bigoplus \frac{c(t,D)}{c(t,D)+1}$	0.2033

ent is small, which is also consistent with our observation from Table 2. It is clear that the complex implementation of IDF does not necessarily lead to better performance in the proposed tie-breaking ranking strategy.

3.3 Experiments on More Test Collections

To evaluate how well the derived tie-breaking methods can be generalized to other collections, we evaluate its performance on four TREC collections:

- *Robust05*: the collection used in TREC 2005 Robust track;
- *Wt2g*: the collection used in TREC 8 Web track;
- *MB11*: the collection used in TREC 2011 Miroblog track;
- *MB12*: the collection used in TREC 2012 Microblog track.

We compare the proposed method with a few baseline methods: (1) a simple retrieval function that multiplies all the selected signal implementation together; (2) a simple retrieval function that adds all the selected implementation together; (3) Pivoted, a state of art retrieval function derived from vector space model [11]; and (4) Okapi, a state of the art retrieval function derived from classical probabilistic model [8]. The parameters of the last two functions are set based on the default values suggested by the previous study [10].

Table 7 shows the results of the performance comparison. Clearly, the proposed tie-breaking method is more effective in combining these simple implementations of the signals than the commonly used methods, i.e., summation and multiplication. Moreover, on traditional TREC ad hoc retrieval collections, it can achieve comparable performance as the state of the art methods. Finally, the proposed method can outperform the state of the art methods on Microblog collections. Another advantage of the proposed method is that it does not have any parameter that needs to be tuned.

Table 7: Performance Comparison (MAP_E@1000)

	MB11	MB12	Robust05	Wt2g
$\frac{1}{df(t)} \bigoplus \frac{c(t,D)}{c(t,D)+1} \bigoplus \frac{1}{\|D\|}$	**0.364**	**0.230**	0.165	**0.276**
$\frac{1}{df(t)} \times \frac{c(t,D)}{c(t,D)+1} \times \frac{1}{\|D\|}$	0.145	0.082	0.033	0.072
$\frac{1}{df(t)} + \frac{c(t,D)}{c(t,D)+1} + \frac{1}{\|D\|}$	0.245	0.167	0.164	0.258
pivoted (s=0.2)	0.342	0.204	0.169	0.199
okapi (b=0.75)	0.309	0.172	**0.175**	0.254

4. CONCLUSIONS AND FUTURE WORK

The paper proposes a novel way of combining multiple retrieval signals. Instead of deriving a complicated mathematical function to combine the signals, we explore a tie-breaking based method to rank documents. In particular, we estimate the strength of retrieval signals based on the performance range, prioritize the signals based on their strengths, and then use the weaker signals to break the ties created by the stronger signals. Our experiment results are quite encouraging. First, it is clear that simple implementations of retrieval signals works as well as the complicated ones as long as they satisfy the retrieval constraints. Second, the proposed tie-breaking strategy is effective. It can achieve comparable performance as the state of the art retrieval functions on traditional TREC collections, and can outperform them on TREC microblog collections.

There are many interesting future directions that we plan to pursue. First, we plan to explore more retrieval signals such as term proximity. Second, it would be interesting to study how to design more retrieval constraints so that we can break the ties based on the constraints. Finally, since the proposed method enable us to use simple signals to achieve comparable effectiveness, it would be interesting to study how this would impact the efficiency of an IR system.

5. REFERENCES

[1] S. Clinchant and E. Gaussier. Information-based models for ad hoc ir. In *Proceedings of SIGIR'10*, 2010.

[2] H. Fang, T. Tao, and C. Zhai. Diagnostic evaluation of information retrieval models. *ACM Transactions on Information Systems*, 29:1–42, 2011.

[3] H. Fang and C. Zhai. An exploration of axiomatic approaches to information retrieval. In *Proceedings of SIGIR'05*, 2005.

[4] Y. Lv and C. Zhai. Lower-bounding term frequency normalization. In *Proceedings of CIKM'11*, 2011.

[5] J. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the ACM SIGIR'98*, 1998.

[6] S. Robertson and K. Sparck Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27:129–146, 1976.

[7] S. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of SIGIR'94*, pages 232–241, 1994.

[8] S. E. Robertson, S. Walker, S. Jones, M. M.Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In D. K. Harman, editor, *The Third Text REtrieval Conference (TREC-3)*, pages 109–126, 1995.

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

[10] A. Singhal. Modern information retrieval: A brief overview. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 24(4):35–43, 2001.

[11] A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *Proceedings of the 1996 ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 21–29, 1996.

[12] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of SIGIR'01*, 2001.