# Examining the Information Retrieval Process from an Inductive Perspective

Ronan Cummins
School of Computing Science
University of Glasgow
Scotland, UK
ronanc@dcs.gla.ac.uk

Mounia Ialmas
School of Computing Science
University of Glasgow
Scotland, UK
mounia@acm.org

Colm O'Riordan
Dept. of Information Technology
NUI Galway, Ireland
colmor@it.nuigalway.ie

## ABSTRACT

Term-weighting functions derived from various models of retrieval aim to model human notions of relevance more accurately. However, there is a lack of analysis of the sources of evidence from which important features of these term weighting schemes originate. In general, features pertaining to these term-weighting schemes can be collected from (1) the document, (2) the entire collection and (3) the query. In this work, we perform an empirical analysis to determine the increase in effectiveness as information from these three different sources becomes more accurate.

First, we determine the number of documents to be indexed to accurately estimate collection-wide features to obtain near optimal effectiveness for a range of a term-weighting functions. Similarly, we determine the amount of a document and query that must be sampled to achieve near-peak effectiveness. This analysis also allows us to determine the factors that contribute most to the performance of a term-weighting function (i.e. the document, the collection or the query).

We use our framework to construct a new model of weighting where we discard the 'bag of words' model and aim to retrieve documents based on the initial physical representation of a document using some basic axioms of retrieval. We show that this is a good first step towards incorporating some more interesting features into a term-weighting function.

## Categories and Subject Descriptors

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

## General Terms

Experimentation, Performance

## Keywords

Information Retrieval, Models, Term-Weighting


*CIKM'10,* October 26–30, 2010, Toronto, Ontario, Canada.


## 1. INTRODUCTION

One main focus in research into information retrieval (IR) is the modelling and estimation of the human notion of relevance given an information need (query) and a large unstructured collection of information items (e.g. document collection).

Numerous models have been proposed to try and correctly model the notions underlying relevance and subsequent retrieval. These models have been adopted with the aim of gaining new insights into retrieval and ultimately, to improve the effectiveness of the retrieval process. Most models place both documents and queries into a framework in which they can be operated upon by operations inherent within the algebra of the model itself. Whether a specific model actually uncovers real truths regarding retrieval and relevance is open to question. In reality, the ranking functions produced from these models combine similar features in a similar manner to construct a term-weighting function. In general, these features can be collected from three different sources (namely the document, the collection and the query itself). Indeed, several works describe term-weighting functions solely by the different weights applied to each of these sources using a triple representation [18].

Users typically determine relevance by reading a piece of text (i.e. document) with an information need in mind (i.e. query), equipped with a good knowledge of the language in which they are searching (i.e. collection-wide information). Some research [9, 10, 11] has aimed to more accurately model this process of retrieval by developing a number of axioms for retrieval. These axioms are deemed valid in an inductive framework that supposes a human linearly reading a document with an information need in mind. The degree of relevance (i.e. a numeric score of some type) changes as the human reads the document, depending on whether a human encounters words/phrases that are on-topic or off-topic. This process of linearly scanning a document is useful in developing of a set of intuitive and useful axioms for term weighting schemes [10].

However, it is possible to adopt this incremental view to the other sources of evidence mentioned. One can equally view other sources of evidence (e.g. collection and query) in a similar manner. In this paper, we adopt this incremental approach and iteratively increase the information available from the three sources of evidence (the document, the collection and the query) to ascertain which of these sources is more sensitive to a lack of information. We apply this method of analysing the retrieval process to a number of state of the art term-weighting functions and present sev-

eral interesting findings. In general, we find that very little of the query and collection need to be sampled to achieve near-peak performance. Therefore, we concentrate on the process of scoring a document within the same framework. Finally, we present a first step towards creating a new type of term-weighting scheme that relaxes the 'bag of words' approach and instead attempts to inherently model the linear process that is undertaken when a human assesses relevance.

In summary, the contributions of this paper are three-fold:

- We outline a method of determining the effect that the various sources of evidence have on the performance of a term-weighting scheme.
- We present results from experiments that incrementally increase the amount of information available from three sources of evidence.
- We create a new type of term-weighting scheme that scores a document similarly to how a reader might linearly scan a document.

The remainder of the paper is organised as follows: Section 2 outlines related work. Section 3 details a number of state of the art term-weighting functions and presents results regarding their performance. Section 4 outlines our incremental approach and studies the effectiveness of different sources of evidence in the retrieval process. Section 5 outlines how new term-weighting functions can be constructed by adopting the inductive process [10]. Section 6 outlines our conclusions and future work.

## 2. RELATED RESEARCH

As previously mentioned there have been many different models proposed for IR. These include the Boolean model, the vector space model [13], classical probabilistic models [12], language models [17], divergence from randomness models [1] and others. Furthermore, attempts have been made to learn term-weighting functions explicitly using an evolutionary model that artificially induces a 'survival of the fittest' paradigm to find suitable term-weighting schemes [6].

Most term weighting schemes assume perfect knowledge of the entire document collection. However, there has been research in the domain of distributed information retrieval where these assumptions do not hold. To deal with the problems of source selection and results merging, attempts are made to estimate the term frequency distributions in text collections. In these scenarios, one must sample the collections to generate suitable estimates which can be used to guide result fusion. Query-based sampling approaches [4] involve generating a number of queries, submitting these queries to the collections, retrieving the top $N$ documents and then updating the term distribution estimates. These queries should be sampled in an appropriate manner [5] and at appropriate times given a dynamic collection. Related work in resource selection uses evidence gleaned from previous queries to build a suitable sample [15]. Some work [16] has also studied the problem of estimating global features (e.g. *idf*) for distributed IR.

However, the work outlined here is different from previous work as our aim is to show how the estimation of information from a number of sources of evidence affects the performance of a number of state of the art term-weighting functions. This process informs us about the sources of evidence and is also a study into term-weighting behaviour. A study of sources of evidence for vertical selection has also been conducted recently [2]. However, the task studied therein (vertical selection) and the data sources used are quite different to those studied in this paper.

A recent approach [10] to modelling the retrieval problem has been to aim to develop a number of axioms and to build up a retrieval foundation from which we can develop new term-weighting schemes. This model supposes a reader encountering terms as he/she reads a document. This process at least more accurately reflects the process of how a human may determine relevance[1]. A mathematical description of these axioms (constraints) is contained in [10]. The first constraint (C1) states that adding a new query term to a document must *always* increase the score of that document. The second constraint (C2) states that adding a non-query term to a document must *always* decrease the score of that document. The third constraint (C3) states that adding successive query terms to a document should increase the score of the document less with each successive addition. Furthermore, the axioms developed have been to shown by empirical studies to be useful estimators of term-weighting performance [10, 8]. A fourth constraint (C4) states that adding more non-query terms to a document should decrease the score of a document less with each occurrence. Furthermore, a proximity constraint (C5) regarding within-document term proximity has also been developed [14].

However, as yet there has been no attempt to create actual term-weighting schemes by directly modelling the inductive process previously outlined [10]. This work attempts to remedy this situation.

## 3. MODELS

In this section we introduce term-weighting formulas that are derived from different models of retrieval. We perform some preliminary experiments and present the performance of these schemes on TREC data.

### 3.1 Term-Weighting Functions

One of the best performing term-weighting functions, $BM25$ [10], is derived from the probabilistic model of retrieval and is defined as follows

$$BM25(Q,D) = \sum_{t \in Q \cap D} \left( \frac{tf_t^D \cdot log(\frac{N - df_t + 0.5}{df_t + 0.5})}{tf_t^D + k_1 \cdot ((1-b) + b \cdot \frac{dl}{dl_{avg}})} \cdot tf_t^Q \right) \quad (1)$$

where $tf_t^D$ is the frequency of a term $t$ in $D$ and $tf_t^Q$ is the frequency of the term in the query $Q$. $dl$ and $dl_{avg}$ are the length and average length of the documents respectively. $N$ is the number of documents in the collection and $df_t$ is the number of documents in which term $t$ appears. $k_1$ and $b$ are tuning parameters set to 1.2 and 0.75 by default.

We study a number of other state of the art term-weighting schemes developed from different models of retrieval. We also use the pivoted document length normalisation ($PIV$) [17] from the vector space model, the $I(n)L2$ function from the divergence from randomness approach ($DFR$) [1], a ranking function ($ES$) developed using a evolutionary learning model [7] and a language modelling approach using dirichlet priors ($LM$) [17]. These five term-weighting functions

[1]Although, it may be noted that a more realistic model might be constructed by incorporating eye-tracking.

cover a wide range of models and are all state of the art in terms of performance. Furthermore, all of these term-weighting schemes use features that are calculated from the three different sources of evidence (i.e. collection, document and query).

## 3.2 Performance of Functions

We outline here the data used in this work and we measure the performance of the five term-weighting functions on that data. The performance is presented so that the reader gets a general view of the performance of the schemes and can refer back to these absolute values at a later stage.

**Table 1: Test Collections**

| | Documents | | | Topics | | | | |
|---|---|---|---|---|---|---|---|---|
| | # | Avg. | Dev. | Range | # | short | med | long |
| | | | | | | Avg. | | |
| FT | 210,158 | 191 | 174 | 251-450 | 188 | 2.6 | 10.5 | 32.3 |
| WSJ | 130,837 | 206 | 219 | 051-200 | 150 | 3.6 | 21.2 | 81.6 |
| FBIS | 130,471 | 241 | 461 | 301-450 | 116 | 2.4 | 10.4 | 33.3 |
| AP | 242,918 | 221 | 114 | 051-200 | 149 | 3.6 | 21.2 | 81.6 |
| LATIMES | 131,896 | 225 | 231 | 301-450 | 143 | 2.4 | 10.4 | 31.1 |

For our analysis and subsequent experiments, we use the FT, FBIS, WSJ and AP collections from TREC disks 1 to 6 as test collections and topics. The LATIMES collection is used later in this work to tune certain parameters in the results section (Section 5). For each set of topics, we create a short query set (title field of the topics), a medium length query set (title and description fields) and a long query set (title, description, narrative and concept fields, where available). Table 1 shows some of the characteristics of the collections used in this research. We stemmed the collections using Porter's algorithm and removed standard stop-words.

Table 2 shows the $MAP$ and $P@10$ for the collections used in this research. The $PIV$ scheme is the poorest performing scheme. In terms of MAP, the best performing function tends to be the $ES$ function, but suffers from a lack of high precision ($P@10$) for medium and long queries on some collections. As mentioned these schemes use features from the three sources of evidence mentioned. We can confirm that many of the differences between the schemes are statistically significant[2]. The next section outlines how we can measure the influence of each source of evidence for this set of term-weighting schemes.

# 4. INCREASING THE ACCURACY OF DIFFERENT INFORMATION SOURCES

The term-weighting approaches outlined in the previous section use features from each of the three different sources of information (i.e. collection, document and query). In this section, we determine the change in effectiveness of each of these term-weighting functions as the information from these sources becomes more accurate. We wish to determine the percentage of a collection (global information) that must be indexed to achieve a near optimal level of performance (i.e. MAP) for a term-weighting approach. This type of information is similar to the information that humans possess regarding the semantic value of a term. Therefore, using this process we can determine the number of documents a human needs to read to gain accurate knowledge about the semantic value of terms in the language. In general, adult reader possess a good knowledge of their language, but for

[2]We do not include a full breakdown of statistical comparisons between all combinations of term-weighting scheme on different datasets for this preliminary analysis.

**Table 2: MAP (P@10) on test collections**

| Functions | FT | WSJ | FBIS | AP |
|---|---|---|---|---|
| Topics | 251-450 | 051-200 | 251-450 | 051-200 |
| *short queries* | | | | |
| $PIV$ | 0.2211 (0.26) | 0.1686 (0.35) | 0.2165 (0.28) | 0.1639 (0.26) |
| $BM25$ | 0.2281 (0.26) | 0.1685 (0.35) | 0.2298 (0.29) | 0.1632 (0.25) |
| $DFR$ | 0.2321 (0.27) | 0.1701 (0.35) | 0.2344 (0.30) | 0.1632 (0.26) |
| $ES$ | 0.2402 (0.28) | 0.1786 (0.37) | 0.2663 (0.31) | 0.1644 (0.27) |
| $LM$ | 0.2326 (0.27) | 0.1784 (0.37) | 0.2497 (0.29) | 0.1626 (0.27) |
| *medium queries* | | | | |
| $PIV$ | 0.2534 (0.30) | 0.1926 (0.41) | 0.2309 (0.31) | 0.1891 (0.31) |
| $BM25$ | 0.2540 (0.30) | 0.1972 (0.42) | 0.2475 (0.32) | 0.1888 (0.30) |
| $DFR$ | 0.2581 (0.30) | 0.1993 (0.42) | 0.2529 (0.32) | 0.1868 (0.30) |
| $ES$ | 0.2652 (0.29) | 0.2010 (0.39) | 0.2920 (0.31) | 0.1780 (0.25) |
| $LM$ | 0.2630 (0.30) | 0.1914 (0.40) | 0.2869 (0.31) | 0.1814 (0.28) |
| *long queries* | | | | |
| $PIV$ | 0.2688 (0.33) | 0.2840 (0.54) | 0.2434 (0.32) | 0.2639 (0.40) |
| $BM25$ | 0.2786 (0.33) | 0.2865 (0.55) | 0.2597 (0.33) | 0.2653 (0.40) |
| $DFR$ | 0.2813 (0.33) | 0.2887 (0.55) | 0.2588 (0.33) | 0.2638 (0.40) |
| $ES$ | 0.2959 (0.32) | 0.2595 (0.48) | 0.2847 (0.31) | 0.2361 (0.33) |
| $LM$ | 0.2675 (0.32) | 0.2672 (0.52) | 0.2834 (0.28) | 0.2504 (0.37) |

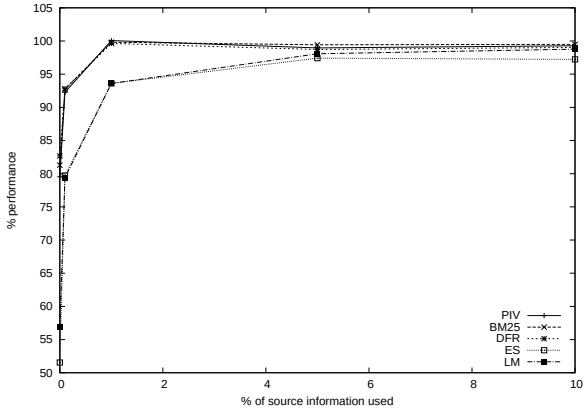


**Figure 1: % MAP increase on FT collection for short queries as global information becomes more accurate**

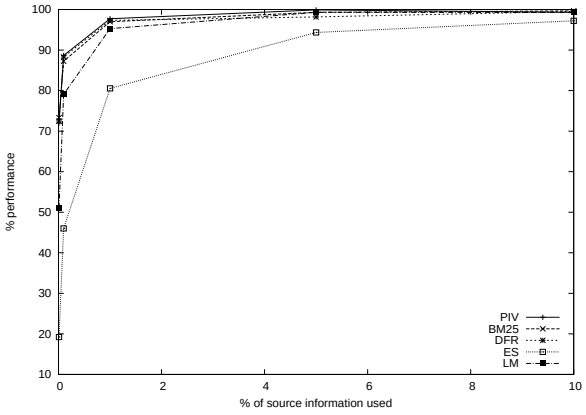


**Figure 2: % MAP increase on FT collection for long queries as global information becomes more accurate**

specialised collections this general information may not be as useful.

To this end, we perform a number of experiments that measures the change in performance as varying amounts of the collection are indexed (sampled). We sample the following percentages of the collection: (0.01, 0.1, 1, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90 and 100). For the graphs that follow in this section, one can determine how quickly the performance reaches 100% (i.e. if the graph rises quickly and remains flat at around 100%, it means we only need to sample a small amount of the source to achieve 100% performance).

We also perform a similar experiment on the document source. We measure the change in performance of a term-weighting function as the information in the document source becomes more accurate. This will inform us whether a human can ignore much of the latter part of the document or whether one must read the complete document to determine relevance. Finally, we perform a similar experiment with the query source, where we increase the number of unique query terms used in the representation and measure the performance as more keywords are added to it. For the global based experiments, we index documents as they are ordered within the TREC collection. For the document and query experiments, we increase the number of terms (information) in the order in which the document and query is written/read in natural language.

## 4.1 Global Information

Global information has an effect on the calculation of *idf* type features, average document length ($dl_{avg}$) features and collection size ($|C|$) features in the term-weighting schemes. Global information is completely accurate once the entire collection has been indexed and therefore, we measure the effectiveness of the term-weighting schemes as a percentage of the performance when all global features are entirely accurate.

Figures 1 and 2 show the percentage effectiveness achieved (in terms of $MAP$) for all of the term-weighting functions as the number of documents indexed in a collection is increased. For the FT collection, we can see that once 10% of the collection is indexed all the functions have achieved above 95% of the effectiveness that could be achieved for a term-weighting scheme, when the entire collection is indexed. This is true for all lengths of queries (short, medium and long). The results from the AP, FBIS and WSJ collection (not shown) are very similar to those on the FT collection. While global information is important in term-weighting schemes, it does not take much information to obtain near optimal global estimates. The results of these experiments when using $P$@10 as a measure of effectiveness are almost identical.

Furthermore, we can see that the $LM$ and $ES$ term-weighting functions perform very poorly when there is very little of the collection known (i.e. less than 1% of the collection). However, the $PIV$, $BM25$ and $DFR$ schemes perform very close to their maximum performance with less global information. In general, only a small sample of the collection (language base) is needed to achieve a high level of performance. This is quite an interesting finding as it indicates that only a small number of documents need to be sampled to achieve a good performance (possibly useful in a filtering scenario). Furthermore, from the results of this experiment, we can also determine the usefulness of global information to each term-weighting function. For example, if we look at the short queries on the FT collection (Figure 1), we can see that for the $ES$ and $LM$ schemes over 40% of the performance comes from global information (as if we ignore global information, the performance of those schemes drops to about 60%). We can also see that the contribution of global information to the performance of the $PIV$, $BM25$ and $DRF$ schemes is much less that those of the other schemes.

## 4.2 Within-Document Information

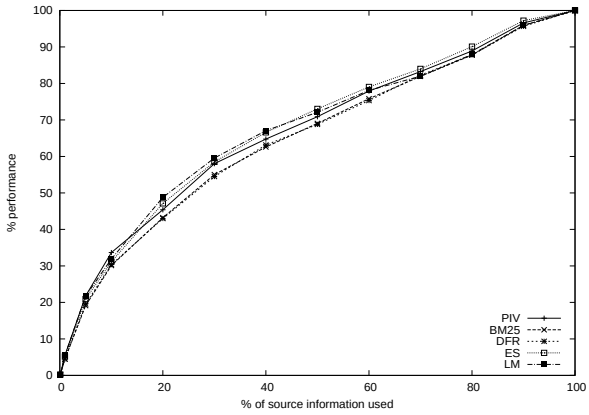


**Figure 3: % MAP increase on FT collection for short queries as local information becomes more accurate**

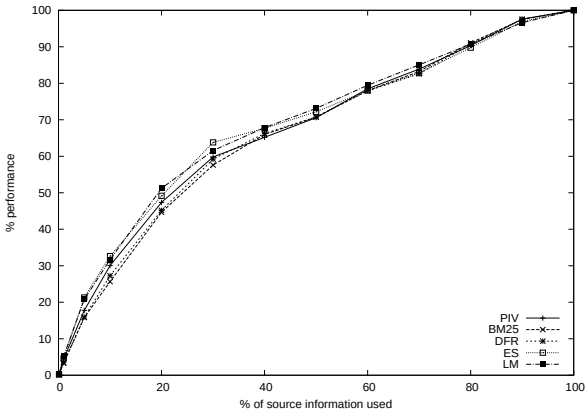


**Figure 4: % MAP increase on FT collection for long queries as local information becomes more accurate**

Having studied the effect that the estimation of global information has on the performance of term-weighting schemes, we can turn our attention to local information. Local information is the information within the information item that is currently being assessed (or read). Local information usually consists of length information and term-frequency information (although this is not all encompassing).

Figures 3 and 4 show the effectiveness as we encounter larger samples of the document for the FT collection. We can see that to achieve at least 90% of the peak effectiveness, we need to read 80% of each document. This indicates that there is important relevant information in much of the document. The results for the FBIS, AP and WSJ collection (not

shown) are again very similar to that of the FT collection. In general, to achieve anywhere near the peak effectiveness we need to use the entire document. Again, the results of these experiment are almost identical when using $P$@10 as a measure of effectiveness.

## 4.3 Within-Query Information

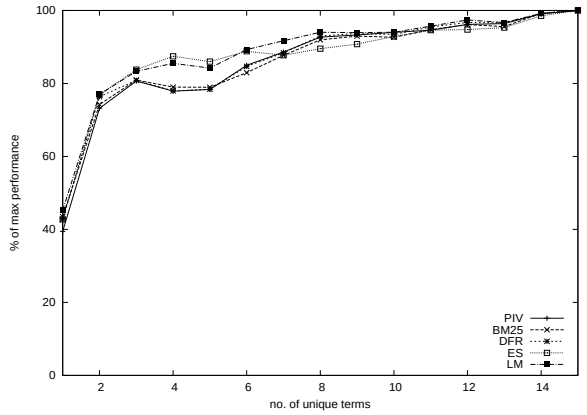


**Figure 5: % MAP increase when the query length increases on the FT collection**

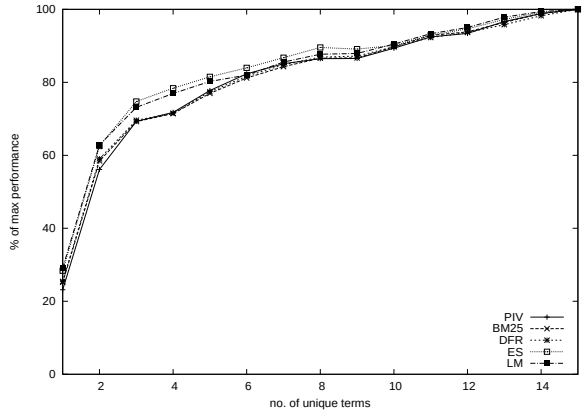


**Figure 6: % MAP increase when the query length increases on the AP collection**

We now analyse the contribution that the information in the query has on performance by measuring the performance each time we encounter a new (unseen) term in the query. We consider queries in this incremental manner measuring the performance as each new term is encountered for all queries until a query of length (measured by unique terms) 15 is reached. For the collections in this research, a query of 15 unique terms is about 25 terms on average (i.e. a sizeable query). We can see from Figure 5 and Figure 6 that most of the performance (about 80%) can be achieved using a 4 term query (i.e 4 unique terms). For the FBIS collection (not shown), 90% of the performance can be achieved using the first 3 terms. The results on the WSJ collection are similar to those on the AP collection. This result confirms previous findings that suggests queries of length 2 to 5 are most effective [3] when balancing effectiveness and effort.

## 4.4 Breakdown by Source

Using the results from the previous three sections, we can determine the percentage of performance that comes from various sources. We can calculate the percentage of performance that comes from the collection by indexing very little or no documents. For each query type (short, medium and long), we can calculate the percentage of information that comes from the query source by preventing a scheme from using query term features (i.e. within-query term-frequencies). Therefore, the remainder of the performance can be deemed as coming from the document source. It is true that information is not strictly mutually exclusive to each source. For example, term occurrences in each document affects global information. Furthermore, in large documents the distribution of terms may be some way representative of the distribution across the entire collection. Regardless, the method chosen in this work informs us as to the effectiveness of each source in comparison with other term-weighting functions, which has not been shown before.

Figures 7 and 8 show the breakdown of the effectiveness by source on the FT and AP collections. The results on the other collections are very similar. We can see that for the $ES$ and $LM$ schemes a lot of the effectiveness comes from better use of the global information. We can see for short queries that there is little or no information in the query (other than term occurrence/absence). This is because the lengths and within-query term-frequencies are all small (i.e. limited information). Short queries (i.e. common web type queries) have very little extra information (i.e. other than term occurrence). The results also indicate that the $ES$ scheme uses less query information than the other schemes to achieve its performance (this can be seen for medium and long queries). Another interesting point is that the effectiveness of the $PIV$, $BM25$ and $DFR$ schemes is distributed similarly from the sources of information (although, the performance of these schemes is different). Not surprisingly, global information is less useful for shorter queries (as if we consider the case of a query of length 1, we can deduce that global information is not useful at all). This global information becomes more important for longer queries. These results are also useful for one's choice of term-weighting scheme for a particular task. For example, in situations where global information is unavailable (i.e. a cold start in a filtering system), $DFR$ or $BM25$ would be a good choice of term-weighting.

## 4.5 Summary

We have shown that very little global information is needed to achieve good performance for a centralised index. We have also shown that for the collections used in this work, all of the document must be read to achieve close to maximum performance. Our query-based experiments have shown that there seems to be diminishing returns when using queries longer than four unique terms. Furthermore, we have shown that when no (or very little) global information is available, the learned function ($ES$) and the language modelling approach ($LM$) perform poorly. We have determined that the $ES$ and $LM$ scheme make better use of global information and that there is a limited amount of information in the query (especially for shorter queries). Therefore, we turn our attention to developing a better representation for exploiting information within the document itself. The next section deals with this process.

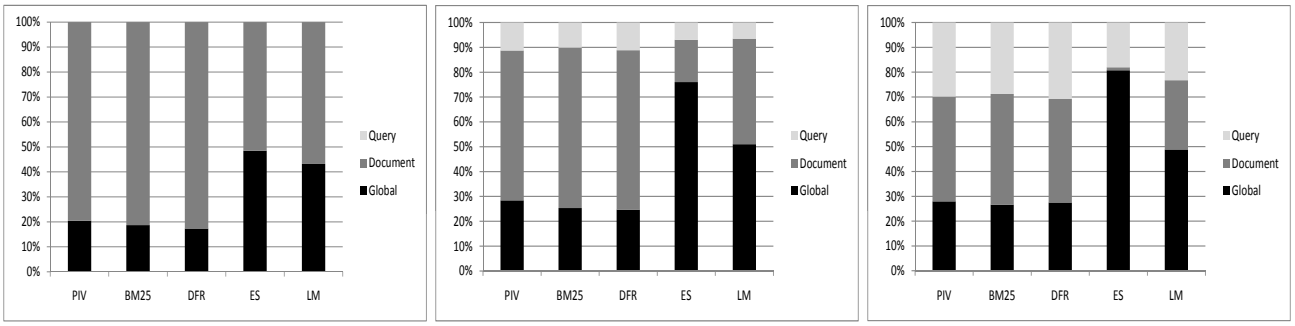


**Figure 7: % of performance from different sources on FT collection for short, medium and long queries**

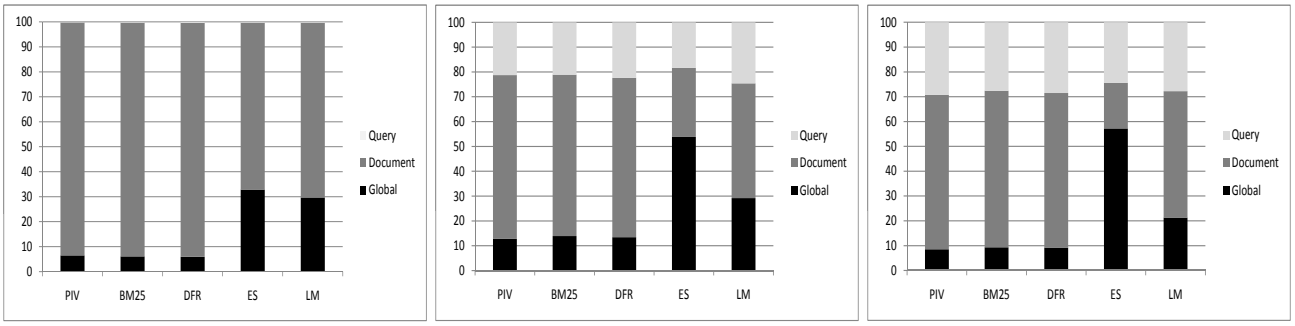


**Figure 8: % of performance from different sources on AP collection for short, medium and long queries**

# 5. NEW TERM-WEIGHTING APPROACH

We have shown that different schemes achieve their performance using information from different sources. Typically a short query (less than five terms) is suitable for specifying an information need. Therefore, there is very little extra information that can be gathered from such a short query. Similarly, very little frequency information needs to be collected (in a global context) to inform a user about the discrimination value (or resolving power) of a term. Because of this and because ultimately, it is the document representation that determines the performance of a particular ranking strategy, we will turn our attention to the document representation and in particular, we look more closely at the linear traversal of a document (similar to how a person might read a document and determine relevance). In this section, we aim to construct a ranking function based on linearly scanning a document using the natural ordering of terms. To aid us in developing a term-weighting strategy we will make use of a number of constraints.

## 5.1 Inductive Approach

As mentioned earlier in the related work section, a number of axioms have already been constructed assuming the inductive approach [10]. We now wish to construct a term-weighting scheme that can score a document using this approach. Therefore our term-weighting scheme will scan through an entire document in a linear manner as the imaginary user would. The estimation of relevance as this process occurs is governed by the axioms. Furthermore, one can notice that the document remains in the same representation as it is in reality. We do not propose that this representation is indeed the true (correct) view of language or meaning for humans. However, we do submit that it is through the construction of correct axioms that apply to the original representation of a document that we will be able to infer a greater understanding of retrieval (and possibly a truer view of relevance). Indeed, we as yet do not fully understand axioms for meaning, relevance and the relatedness between terms, although as we have noted there has been some attempts to do this [10, 11]. Nonetheless, this representation does allow us easier access to a number of interesting document features (e.g. position, proximity, etc)[3].

### 5.1.1 Initial Weight for Terms

As a basis for construction of this new weighting scheme, we will assume perfect knowledge of the collection. The initial weight ($w(t)$) of a term is usually some type of term discrimination value or measure of resolving power. When reading a document, most adult readers already possess a good estimate of the expected frequency of terms in a global context (i.e. the semantic content of a term). Therefore, we can model the initial weight ($w(t)$) of a term as one of the following functions (keeping in mind that we do not need a large sample of a collection to get accurate values):

$$gw(t) = \sqrt{\frac{cf_t^3 \cdot N}{df_t^4}} \tag{2}$$

$$idf(t) = log(\frac{N}{df_t}) \tag{3}$$

where $gw(t)$ is the global part of the $ES$ scheme and $idf(t)$

[3]It can be noted that the computational complexity of such an approach will be $O(D)$ instead of $O(Q)$. The aim of this work it is to seek performance improvements, or at least create a more intuitive framework that more readily allows access to, and intuitive incorporation of, more features (i.e. position, proximity etc).

is a simplified version of the *idf* factor from the $PIV$, $BM25$ and $DFR$ schemes.

### 5.1.2 Term-Frequency Aspect

We know from the first constraint ($C1$) that as on-topic terms (query terms) appear the estimation of relevance of a document (or score) must increase. From the second constraint ($C3$) we also know that the increase in weight for a term must decrease with successive occurrences. Therefore, to model this aspect in our inductive term-weighting model we use the following damping factor $d(n_t)$:

$$d(n_t) = w(t) \cdot \frac{1}{n_t^x} \tag{4}$$

where $n_t$ in the $n^{th}$ occurrence of term $t$ in the document. We can now score a document as we linearly traverse a document $D$ as follows:

$$L(Q,D) = \sum_{t \in D} \left\{ \; d(n_t) \cdot tf_t^Q \quad \forall t \in Q \; \right\}$$

The damping function, $d(n_t)$, acts as a term-frequency aspect. The first time the reader encounters term $t$, the score of the document is increased with the full initial weight ($w(t)$) of term $t$ (e.g. $idf(t)$). When the reader next encounters term $t$, the score is increased with a smaller weight. Figure 9 shows the fraction of the initial weight to add to the score of a document as a reader re-encounters a term (for different values of $x$). We can see that the first time a term is encountered (i.e. $n = 1$ ), the full weight of a term is added to the score. On successive occurrences a fraction of the initial weight is added. We used the LATIMES collection to tune this parameter (i.e. $x$). We found a value of 1.6 (see Figure 10) to be suitable for queries of different length using both $idf(t)$ and $gw(t)$ as an initial weight ($w(t)$). From Figure 9 we can see that a value of $x = 1.6$ means that the initial weight of the term drops to about 1/3 of its initial weight when it occurs for the second time.

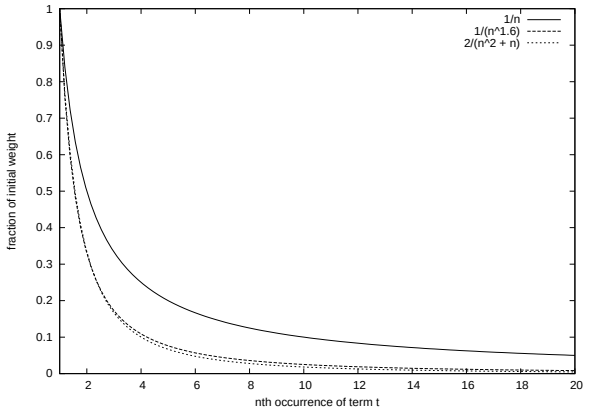


**Figure 9: The fraction of the initial weight added to the score as a query term is re-encountered**

A brief analysis of this damping factor shows that it is similar to the term-weighting factor of other weighting schemes. The term-frequency parts of both $BM25$ (when $k_1 = 1$) and the $DFR$ schemes can be written as $TF(n) = \frac{n}{n+1}$ for an average length document where $n = tf_t^D$. The additional

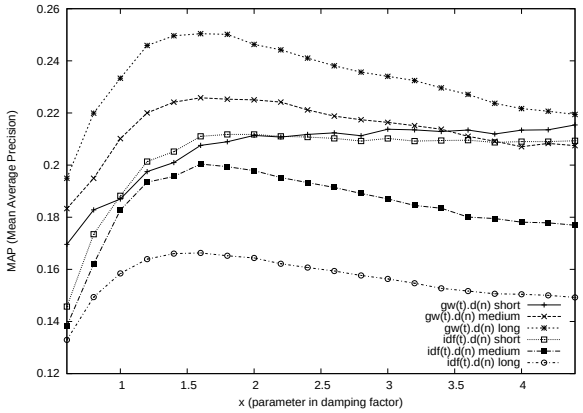


**Figure 10: Tuning the damping parameter on LATIMES**

weight for each of the $n^{th}$ successive term occurrence can then be calculated as follows:

$$d(n_t) = TF(n) - TF(n-1) = \frac{n}{n+1} - \frac{n-1}{n} = \frac{1}{n^2+n} \tag{5}$$

which can be multiplied by a constant in a ranking situation to $\frac{2}{n^2+n}$. In Figure 9 it can be seen that this function is a very close fit to the tuned function that we use as a damping factor.

### 5.1.3 Normalisation Aspect I

Now we attempt to satisfy constraint 2 (and constraint 4) so that non-query (off-topic) terms are penalised. We can incorporate this as follows :

$$L(Q,D) = \sum_{t \in D} \left\{ \begin{array}{ll} d(n_t) \cdot tf_t^Q & \forall t \in Q \\ -\lambda \cdot d(n_t) & \forall t \notin Q \end{array} \right\}$$

where $-\lambda$ is a penalising factor. This subtracts a weight every time a non-query term is encountered. We can see from Figure 11 that this normalisation acts similarly to other types of normalisation (i.e. short queries require little normalisation, while longer queries require more normalisation). We found that using a value of $\frac{30}{10000}$ and $\frac{5}{10000}$ were the best, on average, for different query lengths on the LATIMES collection when using $idf(t)$ and $gw(t)$ as initial weights respectively. Table 3 shows the results of applying this sort of normalisation to both $idf(t)$ and $gw(t)$ and is labelled '$nd_1$' to denote this type of normalisation. Unfortunately, this simplistic normalisation approach does not work well for medium or long queries. Therefore, we try another approach.

### 5.1.4 Normalisation Aspect II

For our second approach to normalisation, we will incorporate the document length ($dl$) directly into the function. We set $\lambda$ to zero and normalise the initial weight of a query term as follows:

$$nd_2(n_t) = w(t) \cdot \frac{1}{n_t^{1.6} + N()} \tag{6}$$

where $N()$ is some normalisation function. If $N() = 0$ there will be no normalisation, while an increase in $N()$ will de-

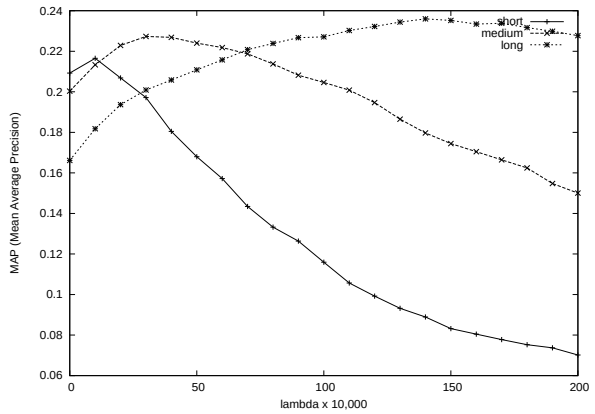


**Figure 11: Tuning the normalisation parameter (lambda) on LATIMES when using *idf(t)***

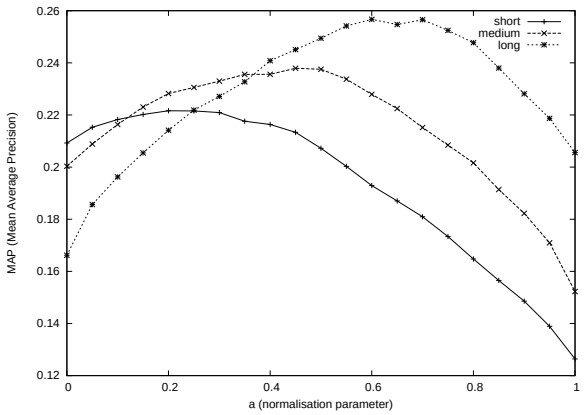


**Figure 12: Tuning the normalisation parameter (a) on LATIMES when using *idf(t)***

crease the $w(t)$ weight more. Using this method with our damping factor ($nd_2()$), we would expect the normalisation function to be zero (i.e. $N() = 0$) for an average length document and to decrease the score for documents longer than the average document. Therefore, we use the following formula:

$$N() = a \cdot \frac{sqrt(dl) - sqrt(dl_{avg})}{sqrt(dl_{avg})} \quad (7)$$

where $a$ is a tuning factor that is similar to normalisation tuning factors in other term-weighting schemes. We take the square root of the lengths and average length of the document to adhere to constraint 4 (C4). Figure 12 shows that the parameter $a$ varies per query length as for other weighting schemes. For a fair comparison we took a single value of $a$ for all query lengths. We set $a = 0.5$ when using the $idf(t)$ as the initial weight and $a = 0.25$ when using $gw(t)$. The results from this approach to normalisation can be seen in Table 3. We can see that adding this type of normalisation (labelled '$nd_2$') to our term-weighting schemes is comparable to $BM25$ and $ES$ on some collections and outperforms them on others. In general, we see a slight improvement over $BM25$ and $ES$ when using our new weighting scheme with this type of normalisation ($nd_2$).

### 5.1.5 Proximity Aspect

Now that we have defined a basic term-weighting approach that is at least comparable to our baseline functions in terms of performance, we now attempt some improvements. As one is scanning through a document in such a manner, it is very simple to incorporate other information (such as term position and term proximity). To show this, we incorporate one of these heuristics (i.e. proximity) and show that we can achieve performance gains using this representation. It is intuitive that a reader has some estimate of when an on-topic term last appeared. We incorporate proximity by remembering the position of the last occurring query term as we scan through a document. For example, if a query term appears in a particular position (e.g. position 7) and we encounter another query term very soon afterward (e.g. position 9), we increase the weight of that document. In this way, the complexity of the approach does not change (as we only iterate through the document once). The scoring function can be written as:

$$L(Q, D) = \sum_{t \in D} \{ \; nd_2(n_t) \cdot tf_t^D) + \max(p(t, t')) \quad \forall t, t' \in Q \; \}$$

where $p(t, t')$ is a function which measures the distance between the current query term $t$ and the most recent previous query term $t'$ when $t \neq t'$. It is intuitive that the weight to increase the document by is proportional to the initial weight (i.e. discrimination value) of each term involved and inversely proportional to the distance between the two terms within the segment of text. We use the proximity constraint [14] (C5), which says that the proximity curve should be convex, to instantiate $p(t, t')$. Therefore, we define $p(t, t')$ as:

$$p(t, t') = \sqrt{w(t) \cdot w(t')} \cdot \frac{1}{dist(t, t')^y} \quad (8)$$

where $t$ is the current query term, $t'$ is the query term last encountered, $dist(t, t')$ is the difference between the two query term positions (i.e. linear distance between both terms) and $y$ is a tuning factor. We have found that $y = 1.6$ is a good choice for this parameter. We use the geometric mean of the initial term weights to weight the proximity score. For example, if the distance between the terms is 1 (i.e. they are a bi-gram), we add a score of $\sqrt{w(t) \cdot w(t')}$ to the document. We take the maximum value of the function $p(t, t')$ as the document is traversed. We also experimented with the average value of this function. Table 4 shows the results of the schemes that include proximity (both the maximum and average for our proximity approach are included). For a fairer comparison, we compared our schemes to $BM25$ and $ES$ when a baseline proximity function [14] is incorporated. We can see that the newly developed term weighting approach outperforms the proximity based versions of $BM25$ and $ES$ on most collections and query lengths and is significant on some data sets.

## 6. CONCLUSION

In this paper we have presented a method of applying an inductive view to the sources of evidence available. By varying the amount of evidence available in these sources we gain an insight into how much of a collection one needs

**Table 3: MAP (P@10) on test collections for non-proximity based functions. Schemes labelled $idf(t)$ should be compared to $BM25$, while those labelled $gw(t)$ should be compared to the $ES$ scheme. Significant increase or decrease (↑ and ↓) compared to respective baseline functions using a Wilcoxon signed-rank test at the 0.05 level.**

| Collections | FT | WSJ | FBIS | AP | AVG |
|---|---|---|---|---|---|
| Topic Range | 251-450 | 051-200 | 301-450 | 051-200 | |
| | | | short queries | | |
| **BM25** | 0.2281 (0.265) | 0.1685 (0.358) | 0.2298 (0.297) | 0.1632 (0.258) | **0.1977 (0.292)** |
| $idf(t).nd_1$ | 0.2320 (0.270) | 0.1750 (0.370) | 0.2481 (0.253) | 0.1636 (0.277) | |
| $idf(t).nd_2$ | 0.2293 (0.260) | 0.1746↑ (0.361) | 0.2384 (0.290) | 0.1638 (0.259) | **0.2015 (0.290)** |
| **ES** | 0.2402 (0.282) | 0.1786 (0.378) | 0.2663 (0.314) | 0.1644 (0.271) | **0.2113 (0.311)** |
| $gw(t).nd_1$ | 0.2228↓ (0.268) | 0.1803 (0.375) | 0.2480↓ (0.263) | 0.1675 (0.285) | |
| $gw(t).nd_2$ | 0.2318↓ (0.273) | 0.1855↑ (0.385) | 0.2573 (0.304) | 0.1685↑ (0.270) | **0.2096 (0.307)** |
| | | | medium queries | | |
| **BM25** | 0.2540 (0.307) | 0.1972 (0.424) | 0.2475 (0.320) | 0.1885 (0.305) | **0.2227 (0.338)** |
| $idf(t).nd_1$ | 0.2413↓ (0.272) | 0.1745↓ (0.359) | 0.1974↓ (0.200) | 0.1789↓ (0.300) | |
| $idf(t).nd_2$ | 0.2686↑ (0.306) | 0.2021↑ (0.419) | 0.2652↑ (0.324) | 0.1910 (0.308) | **0.2324 (0.338)** |
| **ES** | 0.2652 (0.290) | 0.2010 (0.396) | 0.2920 (0.309) | 0.1780 (0.259) | **0.2333 (0.312)** |
| $gw(t).nd_1$ | 0.2510↓ (0.280) | 0.1956 (0.392) | 0.2561↓ (0.260) | 0.1830 (0.270) | |
| $gw(t).nd_2$ | 0.2604 (0.298) | 0.2108↑ (0.430) | 0.2900 (0.307) | 0.1890↑ (0.280) | **0.2362 (0.329)** |
| | | | long queries | | |
| **BM25** | 0.2786 (0.339) | 0.2865 (0.552) | 0.2597 (0.332) | 0.2653 (0.402) | **0.2737 (0.406)** |
| $idf(t).nd_1$ | 0.2069↓ (0.252) | 0.2217↓ (0.444) | 0.1322↓ (0.127) | 0.2411↓ (0.377) | |
| $idf(t).nd_2$ | 0.2838 (0.338) | 0.2869 (0.544) | 0.2718 (0.318) | 0.2627 (0.396) | **0.2771 (0.400)** |
| **ES** | 0.2959 (0.321) | 0.2595 (0.484) | 0.2847 (0.312) | 0.2361 (0.334) | **0.2700 (0.362)** |
| $gw(t).nd_1$ | 0.2690↓ (0.303) | 0.2432↓ (0.456) | 0.2344↓ (0.233) | 0.2367 (0.353) | |
| $gw(t).nd_2$ | 0.2884 (0.332) | 0.2692↑ (0.504) | 0.2990↑ (0.324) | 0.2450↑ (0.357) | **0.2754 (0.379)** |

to index in order to estimate global collection wide information to achieve a particular level of performance for a number of weighting schemes. We show that for many well known schemes, suitably precise estimates can be calculated while indexing only a relatively small amount of the collection. We also find that in most cases with respect to the evidence present in the query, that we again do not need much evidence to achieve reasonable performance. Regarding the evidence present in the documents themselves (local document evidence), the majority of the document must be scanned in order to achieved satisfactory performance. This indicates that there is important information, regarding relevance in all of the document.

Furthermore, we have introduced a term-weighting approach wherein the incremental linear traversal of a document is modelled to assign a relevance score to that document. While more computationally expensive that other approaches, we argue that this intuitive inductive framework for assigning relevance provides the potential to correctly model extra within-document evidence (e.g. proximity information, positional information). To illustrate the benefit of this approach, we have presented a set of experiments and results that show that this approach is at least as effective as current approaches and worth pursuing.

Future work will entail the incorporation of more features of within-document evidence into the weighting schemes. Interestingly, there are many other features that may possibly be used in this framework. Natural language processing could possibly be used to create new features. For example, different parts of speech (nouns, verbs etc) may be assigned different initial weights. Also, parse trees may be utilised to develop a different type of proximity (possibly a more realistic view of the proximity between terms in a sentence). It would be interesting, for future work, to automatically learn a term-weighting function within this model that can use proximity, position, normalisation and content features together to score a document.

## 7. ACKNOWLEDGMENTS

Ronan Cummins is funded by the Irish Research Council (IRCSET), co-funded by Marie Curie Actions under FP7. Mounia Lalmas is funded by Microsoft Research/Royal Academy of Engineering.

## 8. REFERENCES

[1] Gianni 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, 2002.

[2] Jaime Arguello, Fernando Diaz, Jamie Callan, and Jean-Francois Crespo. Sources of evidence for vertical selection. In *SIGIR '09:*, pages 315–322, New York, NY, USA, 2009. ACM.

[3] Leif Azzopardi. Query side evaluation: an empirical analysis of effectiveness and effort. In *SIGIR*, pages 556–563, 2009.

**Table 4: MAP (P@10) on test collections for proximity based functions. Schemes labelled $idf(t)$ should be compared to $BM25$, while those labelled $gw(t)$ should be compared to the $ES$ scheme. Significant increase or decrease (↑ and ↓) compared to respective baseline functions using a Wilcoxon signed-rank test at the 0.05 level.**

| Collections | FT | WSJ | FBIS | AP | AVG. |
|---|---|---|---|---|---|
| Topic Range | 251-450 | 051-200 | 301-450 | 051-200 | |
| | | | short queries | | |
| **BM25 + Tao()** | 0.2320 (0.268) | 0.1780 (0.368) | 0.2430 (0.307) | 0.1695 (0.271) | **0.2054 (0.301)** |
| $idf(t).nd_2$+AVG | 0.2267 (0.254) | 0.1760 (0.357) | 0.2416 (0.293) | 0.1622 (0.255) | |
| $idf(t).nd_2$+MAX | 0.2311 (0.266) | 0.1826 (0.373) | 0.2545 (0.313) | 0.1681 (0.271) | **0.2078 (0.303)** |
| **ES + Tao()** | 0.2422 (0.284) | 0.1824 (0.388) | 0.2695 (0.317) | 0.1667 (0.278) | **0.2141 (0.314)** |
| $gw(t).nd_2$+AVG | 0.2366 (0.270) | 0.1878 (0.390) | 0.2719 (0.318) | 0.1709 (0.266) | |
| $gw(t).nd_2$+MAX | 0.2413 (0.276) | 0.1908↑ (0.398) | 0.2754 (0.324) | 0.1728↑ (0.278) | **0.2185 (0.316)** |
| | | | medium queries | | |
| **BM25 + Tao()** | 0.2611 (0.312) | 0.2024 (0.429) | 0.2507 (0.327) | 0.1926 (0.309) | **0.2277 (0.343)** |
| $idf(t).nd_2$+AVG | 0.2699 (0.309) | 0.2030 (0.422) | 0.2703 (0.320) | 0.1924 (0.313) | |
| $idf(t).nd_2$+MAX | 0.2744↑ (0.314) | 0.2042 (0.426) | 0.2773↑ (0.331) | 0.1931 (0.312) | **0.2337 (0.344)** |
| **ES + Tao()** | 0.2661 (0.291) | 0.2015 (0.397) | 0.2928 (0.309) | 0.1786 (0.261) | **0.2337 (0.313)** |
| $gw(t).nd_2$+AVG | 0.2661 (0.296) | 0.2121 (0.433) | 0.2926 (0.306) | 0.1903 (0.285) | |
| $gw(t).nd_2$+MAX | 0.2667 (0.299) | 0.2133 ↑ (0.434) | 0.2972↑ (0.313) | 0.1917↑ (0.282) | **0.2404 (0.331)** |
| | | | long queries | | |
| **BM25 + Tao()** | 0.2831 (0.339) | 0.2879 (0.552) | 0.2612 (0.331) | 0.2654 (0.400) | **0.2758 (0.405)** |
| $idf(t).nd_2$+AVG | 0.2856 (0.344) | 0.2878 (0.547) | 0.2735 (0.322) | 0.2638 (0.398) | |
| $idf(t).nd_2$+MAX | 0.2859 (0.348) | 0.2883 (0.546) | 0.2763 (0.325) | 0.2647 (0.395) | **0.2795 (0.404)** |
| **ES + Tao()** | 0.2961 (0.322) | 0.2596 (0.484) | 0.2844 (0.312) | 0.2362 (0.335) | **0.2701 (0.363)** |
| $gw(t).nd_2$+AVG | 0.2906 (0.331) | 0.2699 (0.503) | 0.2993 (0.324) | 0.2454 (0.359) | |
| $gw(t).nd_2$+MAX | 0.2891 (0.333) | 0.2698↑ (0.504) | 0.3012↑ (0.330) | 0.2447↑ (0.356) | **0.2757 (0.380)** |


[4] Jamie Callan and Margaret Connell. Query-based sampling of text databases. *ACM Trans. Inf. Syst.*, 19(2):97–130, 2001.

[5] James Caverlee, Ling Liu, and Joonsoo Bae. Distributed query sampling: a quality-conscious approach. In *SIGIR '06:*, pages 340–347, New York, NY, USA, 2006. ACM.

[6] Ronan Cummins and Colm O'Riordan. Evolving local and global weighting schemes in information retrieval. *Information Retrieval*, 9(3):311–330, 2006.

[7] Ronan Cummins and Colm O'Riordan. Learning in a pairwise term-term proximity framework for information retrieval. In *SIGIR '09:*, pages 251–258, New York, NY, USA, 2009. ACM.

[8] Ronan Cummins and Colm O'Riordan. Measuring constraint violations in information retrieval. In *SIGIR '09:*, pages 722–723, New York, NY, USA, 2009. ACM.

[9] Hui Fang, Tao Tao, and ChengXiang Zhai. A formal study of information retrieval heuristics. In *SIGIR '04:*, pages 49–56, New York, NY, USA, 2004. ACM.

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

[11] Hui Fang and ChengXiang Zhai. Semantic term matching in axiomatic approaches to information retrieval. In *SIGIR '06:*, pages 115–122, New York, NY, USA, 2006. ACM.

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

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

[14] Tao Tao and ChengXiang Zhai. An exploration of proximity measures in information retrieval. In *SIGIR '07:*, pages 295–302, New York, NY, USA, 2007. ACM.

[15] Ellen M. Voorhees, Narendra K. Gupta, and Ben Johnson-Laird. Learning collection fusion strategies. In *SIGIR '95:*, pages 172–179, New York, NY, USA, 1995. ACM.

[16] Hans Friedrich Witschel. Estimation of global term weights for distributed and ubiquitous ir. In *Workshop on Ubiquitous Knowledge Discovery for users (UKDU'06) in ECML*, 2006.

[17] 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.

[18] Justin Zobel and Alistair Moffat. Exploring the similarity space. *SIGIR Forum*, 32(1):18–34, 1998.