An Axiomatic Study of Learned Term-Weighting Schemes

Ronan Cummins
Dept. of Information Technology
National University of Ireland
Galway, Ireland
ronan.cummins@nuigalway.ie

Colm O'Riordan
Dept. of Information Technology
National University of Ireland
Galway, Ireland
colm.oriordan@nuigalway.ie

ABSTRACT

At present, there exists many term-weighting schemes each based on different underlying models of retrieval. Learning approaches are increasingly being applied to the term-weighting problem, further increasing the number of useful term-weighting approaches available. Many of these term-weighting schemes have certain features and properties in common. As such, it is beneficial to formally model these common features and properties.

In this paper, we introduce a term-weighting scheme that has been developed incrementally using an evolutionary learning approach. We analyse one such term-weighting function produced from the evolutionary approach by decomposing it into inductive query and document growth functions. Consequently, we show that it is consistent with a number of axioms previously postulated for term-weighting schemes. Interestingly, we show that a further constraint can be derived from the resultant scheme.

Finally, we empirically validate our analysis, and the newly developed constraint, by showing that the newly developed nonparametric term-weighting scheme can outperform $BM25$ and the pivoted document length normalisation scheme over many different query types and collections. We conclude that the scheme produced from the learning approach adds further evidence to the validity of the axioms.

1. INTRODUCTION

Term-weighting is crucial to the problem of document ranking within most information retrieval (IR) systems. Many approaches to term-weighting have been developed over the years ranging from empirical models to purely theoretical models. Learning approaches to term-weighting (or indeed ranking) can be classed as purely empirical. Typically, they forgo developing an underlying theoretical model for IR and aim to improve the document ranking (using some objective function on training data) adopting a *bag of words* retrieval model [3, 4, 5]. Other models of retrieval adopt an existing similarity framework [19, 20] into which documents and queries can be transformed. For example, while the vector space model [19] is an intuitive and useful model for mapping documents and queries into an existing similarity framework, there is no theoretical basis for viewing document and queries as vectors. However, due to its simplicity and the high level of performance it achieves, it remains a popular model for IR. Conversely, many purely theoretical models [22] have been developed that often only reach the performance of their empirically based counterparts after some refinement. It is often difficult to ascertain if the term-weighting functions derived from the best performing models (be they empirical or theoretical) contain similar characteristics in terms of ranking.

Some recent research [12] has tried to bridge this gap by attempting to develop criteria for good term-weighting functions. An axiomatic approach to IR [13] has been developed which outlines a number of axioms to which all term-weighting functions should adhere. This approach and the constraints (axioms) in particular, are useful in attempting to theoretically motivate term-weighting functions that are developed from purely automated empirical (learning) models. They are also useful in determining if a proposed theoretical model's interpretation of relevance is indeed valid.

Evolutionary learning techniques, and in particular genetic programming (GP), are becoming popular due to the freedom they offer in the definition of the problem and representation of possible solutions. The basis behind many of these approaches is that useful features and properties within a population of solutions survive and propagate. The GP produces a symbolic representation of the solution, which is useful when a general solution is required. These representations are also useful for further analysis. There have been a number of attempts using GP to evolve term-weighting schemes in a *bag of words* retrieval framework [16, 11, 23, 8]. We see the aforementioned axiomatic approach as a useful tool in theoretically motivating the often useful output of these learning techniques. More specifically, we believe that the best functions produced from these learning techniques should adhere to the existing axioms. The axioms serve as a useful guide regarding the optimality of the solutions produced.

This paper presents an analysis of a learned term-weighting scheme using the existing axiomatic framework for IR. The term-weighting scheme herein has been developed incrementally using an evolutionary learning process in a *bag of words* retrieval framework similar to previous research [10]. We decompose the term-weighting scheme into an inductive query and document growth function using an axiomatic frame-

work and show that it satisfies a number of previously developed constraints [13] in a standard document collection setting. Furthermore, we present a new normalisation constraint which is satisfied by our learned term-weighting function and which is not satisfied by the standard benchmark schemes used herein. Finally, we present results which show that our new learned nonparametric weighting scheme does indeed outperform manually tuned benchmark schemes over multiple collections for various types of query.

The rest of the paper is organised as follows: Section 2 presents current term-weighting approaches and reviews some relevant research in the area. Section 3 outlines our learning framework and introduces term-weighting schemes produced from this approach. In section 4, this learned term-weighting scheme is decomposed into an inductive document and query growth function. Furthermore, we show that the learned term-weighting scheme adheres to all of the existing constraints and we also introduce a new normalisation constraint to which the term-weighting scheme adheres. Our experimental setup is described in section 5, while our results are presented in section 6. Finally, our conclusions are outlined in section 7.

2. BACKGROUND

In this section, we briefly outline current term-weighting approaches and summarise some background material relevant to this research.

2.1 Term-Weighting

2.1.1 $BM25$

The $BM25$ weighting scheme, developed by Robertson et al. [17], is a weighting scheme based on the probabilistic model. The score of a document D in relation to a given query Q can be calculated as follows:

$$BM25(D,Q) = \sum_{t \in Q \cap D} \left(\frac{tf_t^D \cdot w_1}{tf_t^D + k_1 \cdot ((1-b) + b \cdot \frac{dl}{dl_{avg}})} \cdot tf_t^Q\right) \quad (1)$$

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. k_1 is the term-frequency influence parameter which is set to 1.2 by default. The query term weighting used here (tf_t^Q) is slightly different to the original weighting method proposed [17] but has been used successfully in many studies. b is the document normalisation influence parameter and has a default value of 0.75. w_1 is the idf weight identified as follows:

$$w_1 = log(\frac{N - df_t + 0.5}{df_t + 0.5}) \quad (2)$$

where N is the number of documents in the collection and df_t is the number of documents in which term t appears.

2.1.2 Pivoted Normalisation Method

Another matching function is the pivoted document length normalisation scheme [21]. The score of a document in this scheme is calculated as follows:

$$Piv(D,Q) = \sum_{t \in q \cap d} \left(\frac{1 + log(1 + log(tf_t^D))}{(1-s) + s \cdot \frac{dl}{dl_{avg}}} \cdot w_2 \cdot tf_t^Q\right) \quad (3)$$

where s is the normalisation parameter referred to as the slope and has a default value of 0.2. w_2 is the idf function as found in the pivoted normalisation scheme and is identified as follows:

$$w_2 = log(\frac{N+1}{df_t}) \quad (4)$$

We can see that both $BM25$ and the pivoted document length normalisation scheme consist of a term-discrimination part (idf) and a type of normalised term-frequency.

2.2 Exploration of Term-Weighting Space

A constraint based approach to IR has previously been developed [12]. This work identifies a number of constraints to which all good weighting schemes should adhere. It is shown that when a scheme violates one of the proposed constrains it typically indicates non-optimality of the scheme. This work has been extended in a formal axiomatic framework [13]. However, the search for new functions still involves manually constructing weighting functions that adhere to these constraints. This approach is described in detail in the next section (2.3).

An exhaustive search of a limited space of term-weighting functions has previously been conducted [24] using a set of non-primitive (non-atomic) properties and features that comprise many previously existing retrieval functions. They conclude that the search space of similarity measures has a complex landscape making a simple hill-climbing algorithm ineffectual. Using non-primitive features of existing term-weighting schemes in any search (be it exhaustive or not) for term-weighting will bias (or limit) the search toward known forms (and shapes) of term-weighting functions.

A family of probabilistic term-weighting schemes has been developed analytically in an incremental three-stage approach [1]. Starting with developing measures for determining the information content of a term, a complete weighting approach is determined by adding two more methods of normalisation. The first method is a non-linear term-frequency measure which implicitly promotes documents containing more distinct query terms, while the second method explicitly uses the document length to penalise longer documents. This three-stage approach has been adopted by some [10] to learn these weighting functions and test them against standard benchmarks at each stage. This approach constrains the shape of the entire term-weighting function in some way by forcing all three aspects to be present. However, these assumptions seem reasonable and have the advantage of reducing the vast search space while not limiting the shape of, nor the features used within, the constituent functions.

Some research has separated the learning of these weighting approaches into two stages (local and global). A discretisation based learning approach has been used to learn the shape of the local (within-document) weighting function [18]. Similarly, the global (term-discrimination) weighting function is learned by placing terms into bins based on their features and weighting the bins separately.

A genetic programming (GP) approach to developing entire ranking functions has previously been adopted [23, 16, 11, 8]. Entire ranking functions are learned using a set of primitive features of the terms in the documents and collection. These functions are shown to outperform standard benchmarks in some cases. This approach has the advantage of creating new ranking functions using a stochastic global

search technique and does not restrict itself to local search alone. The ability to search this more complex landscape (as was identified in [24]) with its irregular topology lends itself well to this problem domain. However, little in-depth analysis of the solutions produced from such approaches has been conducted and in particular it has not been shown if these learned term-weighting schemes adhere to the existing axioms for IR.

The evolutionary approaches adopted in previous works [23, 16, 11, 8] are useful as they make few assumptions as to the possible makeup of good term-weighting schemes. For example, if indeed *idf* (equations 2 and 4) is the optimal type of basic weighting for terms, a GP approach should be able to find this. The use of primitive functions and terminals allow the process to combine useful terminals and search a large function space for correct function forms.

2.3 Inductive Framework and Axioms

We will briefly introduce the constraints previously developed [13] using the inductive framework. The idea of the inductive framework is to define a base case that describes the score (weight) assigned to a document containing a single term matching (or not matching) a query containing a single term. All other cases can be dealt with inductively, using a document growth function (which describes the change in the document score when a single term is added to the document) and a query growth function (which describes the change in the document score when a single term is added to the query). This is an elegant approach to formalising characteristics of a term-weighting function.

This description of term-weighting components is used to formally describe three axioms (or constraints) that are seen as self-evident (intuitive) in a term-weighting context. The first constraint (constraint 1) states that adding a new query term to the document should always increase the score of a document. This captures the basic behaviour of the term-frequency aspect. The second constraint (constraint 2) states that adding a non-query term to a document must always decrease the score of that document. This constraint ensures that some sort of normalisation is present and details its basic operating principle. The third constraint (constraint 3) states that adding successive query terms to a document will increase the score of the document less with each successive addition. Essentially, the term-frequency influence must be sublinear.

These constraints are used to check the validity of term-weighting schemes before evaluation. Furthermore, term-weighting schemes which adhere to these constraints are shown empirically to outperform weighting schemes that fail to adhere to one or more of the constraints [13]. The constraints are also useful in defining valid bounds on tuning parameters that appear in many existing term-weighting schemes. It should be noted that simply adhering to these constraints does not guarantee a high performance weighting scheme. Rather it is the violation of one of the constraint that indicates the performance is non-optimal (i.e. breaks some rule of the proposed model of relevance).

3. LEARNING FRAMEWORK

As previously indicated, an evolutionary learning approach (GP) [9, 10] has been used to develop the term-weighting scheme presented in this section. In this paradigm, solutions are created at random using a set of predefined features of the terms, documents and queries in a document collection. A 'survival of the fittest' approach is then used to determine what solutions will be used in the next generation. The population for this next generation is developed using the features from the fittest solutions in the previous generation. The objective function of this approach is to maximise the mean average precision (MAP) on a set of training data.

3.1 Incremental Learning

We adopt an incremental approach [1] to develop an entire weighting function. The search space is separated into three parts. Firstly, schemes are learned which aim to correctly measure the information content of a term (i.e. some type of term-discrimination measure). When a suitable measure has been determined (i.e. one that maximises MAP), the term-frequency aspect of the scheme is learned while the term-discrimination measure remains fixed. Once a suitable term-frequency scheme is found (again one that maximises MAP), it remains fixed in the weighting scheme while a normalisation scheme is learned. Once this process is complete an entire weighting scheme is produced.

Although the shape of the possible function is constrained by the manner in which we combine the three aspects of a term-weighting function, we do not enforce the form (shape) of the constituent function. The entire scoring function produced from this learning approach, which scores a document (D) in relation to a query (Q), can be described as follows:

$$LRF(D,Q) = \sum_{t \in Q \cap D} (ntf(tf_t^D, dl) \cdot w_3 \cdot tf_t^Q) \qquad (5)$$

where w_3 is the initial weight of a term and ntf is the normalised term-frequency. tf_t^D and tf_t^Q are the actual term-frequencies of term t in the document D and query Q respectively, while dl is the document length. It can be seen that both $BM25$ and the pivoted normalisation scheme fit this model of retrieval. However, rather than analytically developing schemes for each aspect of the function, we learn weighting schemes whose objective function is to maximise the MAP of a set of queries and document. Learning approaches can only generalise functions when the characteristics of the general data are present in the training collection.

3.2 Learned Term-Weighting Scheme

In this section, we will briefly introduce the term-weighting scheme developed. For a more indepth account of the process the interested reader is directed to [9, 10]. The set of functions that can potentially form part of each constituent function is $F = \{\times, -, +, /, square, \sqrt{}, log, exp\}$.

3.2.1 Term-Discrimination

One of the best term-discrimination schemes found on a training collection of approximately 32,000 documents[1] and outlined in [9] using a set of primitive terminals ($\{cf_t, df_t, N, V, C, 10, 0.5, 1\}$) and functions ($F$) is as follows:

$$w_3 = \sqrt{\frac{cf_t^3 \cdot N}{df_t^4}} \qquad (6)$$

where cf_t is the collection frequency of term t in a collection of size N. The terminals V and C are the vocabulary

[1]OHSUMED documents from 1988 and the 63 topics

and size of the collection (in repeated words) respectively. Many of these schemes were developed and can be found in [9], all of which achieve a level of performance that surpasses that of *idf* on standard test collections.

3.2.2 Term-Frequency Influence

Using this function (w_3) as a measure of the information content of a term, a term-frequency influence scheme can be developed using a similar approach to that in [10] using a set of terminals ($\{tf_t^D, 10, 0.5, 1\}$) and functions (F). One of the best learned term-frequency factors ignoring normalisation (i.e. $ntf(tf_t^D)$), learned on approximately 32,000 documents[2] using w_3 as the measure of information content is as follows:

$$ntf(tf_t^D) = 10 + \frac{log(0.5)}{tf_t^D} + \frac{log(tf_t^D)}{log(1+tf_t^D)} + \frac{log(tf_t^D)}{log(1+tf_t^D) \cdot log(log(10))} \quad (7)$$

Although, this seems complex, it is trivial to show that it is sublinear with respect to the term-frequency (tf_t^D) and will be analysed in a later section. Again, multiple term-frequency functions were developed. The better schemes achieved a similar performance on the training set.

3.2.3 Normalisation

A normalised term-frequency similar to the $BM25$ model is assumed. As such, a normalisation function can be added to the chosen term-frequency (7) as follows:

$$ntf(tf_t^D, dl) = 10 + \frac{log(0.5)}{\frac{tf_t^D}{n(dl)}} + \frac{log(\frac{tf_t^D}{n(dl)})}{log(1+\frac{tf_t^D}{n(dl)})} + \frac{log(\frac{tf_t^D}{n(dl)})}{log(1+\frac{tf_t^D}{n(dl)}) \cdot log(log(10))} \quad (8)$$

where $n(dl)$ is some normalisation function. It is worth noting that this structure does not enforce any of the constraints previously outlined in [13].

It has been noted that normalisation schemes developed to date tend to be collection specific [14] and that tuning parameters are used to adapt these schemes for use on different collections and different topic lengths. In an attempt to develop collection independant normalisation schemes, we adopted the following approach: we constructed training data which consisted of three very similar small collections (approximately 10,000 documents[3] in each). These collections contained different distributions of document length although the majority of documents and queries in each collection were very similar. The objective function of the learning approach was simply the average MAP over the three collections (as they had similar numbers of queries and documents). This allowed the learning approach to adapt the normalisation schemes to the document length distribution in different collections. Again, multiple normalisation functions were learned using a set of terminals ($\{dl, dl_{avg}, \sigma(dl), 10, 0.5, 1\}$) and functions ($F$), one of which is as follows:

$$n(dl) = \sqrt{\frac{dl}{dl_{avg}}} \quad (9)$$

where dl is the length of the document in words, dl_{avg} is the average length document in words and $\sigma(dl)$ is the standard deviation of document length in the collection. This formula is typical of the solutions produced in terms of structure and performance.

The entire scheme outlined has been developed in three stages. It is worth noting that there are no free parameters in this entire weighting scheme. The approach adopted constrains the form of the possible weighting formulas. However, the makeup of the constituent parts remains as unconstrained as possible as we use primitive functions and terminals in each stage. By adopting this framework, resulting schemes are not gauranteed to satisfy the constraints (axioms) [13]. This is outlined in the next section.

4. AXIOMATIC ANALYSIS

In this section, we analyse the learned term-weighting formula using the aforementioned axiomatic framework [13]. We also introduce a new constraint to which our learned term-weighting forumla adheres.

4.1 Inductive Growth Functions

We now describe the inductive growth functions [13] for our newly learned weighting approach and indicate which parts of the function were constrained by our framework. We will then look at the constraints (axioms) outlined in the axiomatic framework [13] and compare the newly developed weighting schemes outlined to see if they satisfy these constraints.

For the inductive process, we use $\{q\}$ to describe a term added to the query and $\{d\}$ to describe a term added to a document. The base case simply describes the weight given to a one-term query matching (or not matching) a one-term document and is described as follows:

$$S(Q,D) = f(q,d) = \begin{cases} weight(q) = weight(d) & q = d \\ penalty(q,d) & q \neq d \end{cases}$$

where $S(Q,D)$ scores a document D in relation to a query Q. The function assigns a score of $weight(q)$ to the document when d matches q, otherwise it assigns a penalty score of $penalty(q,d)$. The learned weighting function outlined in section 3.2 can be rewritten as follows using notation similar in style to that in [13] where $\{x,y\} \in \mathbb{Z} > 0$ refer to the term-frequency and the document length respectively:

$$S(Q,D) = \sum_{t \in Q \cap D} (ntf(x,y) \cdot \sqrt{\frac{cf_t^3 \cdot N}{df_t^4}} \cdot tf_t^Q)$$

where $ntf(x,y)$ is as (8) and $n(y) = \sqrt{\frac{y}{dl_{avg}}}$ (9). We can write the base case of this function as follows:

$$weight(q) = \sqrt{\frac{cf_t^3 N}{df_t^4}} \cdot ntf(1,1)$$

[2]LATIMES documents and 37 medium length topics (301-350)

[3]LATIMES documents and the same 29 medium length topics (301-350) on each of the three collections

which accurately describes the weight assigned to a one term query matching a one term document and was learned in our framework. The following case describes the weight given to a query term that does not match a document term:

$$penalty(q, d) = 0$$

It should be noted that $penalty(q, d) = 0$ because of the way we created our GP framework and not as a result of the GP process itself. Thus, we constrained our search to schemes which did not penalise terms explicitly for not occurring. The following query growth function describes the change in weight assigned to a document as a term is added to the query:

$$g() = S(Q, D) + S(\{q\}, D)$$

This is similar to the pivoted normalisation query growth function as the weight grows linearly as terms are added to the query. Again, this query growth function was imposed by our framework as it can be seen that we weighted query terms in a simplistic manner. It has been previously noted that this is a simplistic form of growth function. However, there has been no justification for a more complex form [13]. The following function is the document growth function which can be written in a somewhat similar manner to that of the $BM25$ weighting [13]:

$$h() = \sum_{t \in Q \cap D - \{d\}} S(Q, \{t\}).\frac{ntf(tf_t^D, dl + 1)}{ntf(1, 1)} + S(Q, \{d\}).\frac{ntf(tf_d^D + 1, dl + 1))}{ntf(1, 1)}$$

This document growth function has been learned in stages and was restricted in a certain sense by the properties and features initally supplied to the learning approach, but as is identified in the next section, it was not constrained by the three axiomatic constraints developed in previous research. This re-writing process helps to examine the difference between this scheme, the $BM25$ scheme and the pivoted document normalisation scheme. It can also be useful for developing new schemes in a similar manner to previous research [13].

4.2 Analysis of growth functions using constraints

In this section we show that this scheme satisfies all three existing constraints [13] for typical test collections. Strictly speaking the constraints are not satified unconditionally but the circumstances which would lead the constraints to being violated are not present on the training or test data used in this research. As we have no tuning parameters in our learned functions all constraints are satisfied for the TREC collections used in this work[4].

The first constraint (constraint 1) states that adding a new query term to the document should always increase the score of a document. In our scheme we can show that $ntf(x + 1, y + 1) > ntf(x, y) \quad \forall x, y > 0$. It is trivial to show that $ntf(x + 1, y) > ntf(x, y) \quad \forall x, y > 0$ which is if we simply ignore the length aspect of the document. As our term-frequency ($ntf(x, y)$) is normalised using $\frac{x}{n(y)}$ when x increases by 1, y will increase by 1. Thus, as long as $n(y)$ is sublinear (as is the case in our formula) or is linear with $n(0) > 0$, this will be satisfied.

The second constraint (constraint 2) states that adding a non-query term must decrease the score of a document. It is true that $ntf(x, y + 1) < ntf(x, y)$ for our scheme as the normalisation scheme identified $n(y)$ increases $\forall y > 0$. This will decrease the score of a document. This constraint enforces some sort of document normalisation approach. As the first two constraints hold it is obvious that $ntf(x+1, y+1) > ntf(x, y+1)$ which simply indicates that adding a query term to a document will achieve a higher score than adding a non-query term to a document.

The third constraint (constraint 3) states that adding successive query terms to a document will increase the score of the document less with each successive occurrence. Essentially the term-frequency influence must be sublinear. Thus, $(ntf(x + 1, y + 1) - ntf(x, y)) > (ntf(x + 2, y + 2) - ntf(x + 1, y + 1)) \quad \forall x, y > 0$ which again is true for all x and y in our formula. As the first constraint is satisfied the normalisation part of the $ntf(x, y)$ formula can be ignored. Thus, showing that $(ntf(x + 1, 1) - ntf(x, 1)) > (ntf(x+2, 1) - ntf(x+1, 1)) \quad \forall x > 0$ is sufficient and trivial.

4.3 New normalisation constraint

We now propose a new constraint to which the best of our evolved normalisation schemes adhere. We show that our evolved formula adheres to this new constraint in most cases and that neither the $BM25$ nor the pivoted normalisation adhere to the constraint. With notation similar to that used in [13] our new constraint can be formalised as follows, where T is the set of terms in a corpus and $\delta_d(d, D, Q) = S(Q, D \cup \{d\}) - S(Q, D)$ (i.e. the change in score as a term is added to the document):

Constraint 4: $\forall Q, D$ and $d \in T$, if $d \notin Q$, $|\delta_d(d, D, Q)|^{-1} > |\delta_d(d, D \cup \{d\}, Q)|^{-1}$.

According to Heaps' law [15], the appearance of new (unseen) terms in a corpus grows in roughly a square-root relationship (sub-linearly) to the document length (in words). Ultimately, it is the number of unique terms that is the best measure of how broad the topic of the document is likely to be. For example, consider a document that has 9 words ($dl = 9$) and contains 3 unique terms (i.e. vector length of 3). If this document grows in length to 100 words ($dl = 100$), the expected number of unique terms would be approximately 10. Thus, as the document grows in length, the topic broadens sub-linearly. Furthermore, it is the number of occurences (term-frequency) of these unique terms

[4]It should be noted that the function $ntf(x, y)$ (8) can yield a negative result when $(x/\sqrt{y/dl_{avg}}) < 0.344$ (i.e. for some positive integer values of x and y). Typically, top ranked documents have multiple occurrences of query terms (i.e. $x > 1$). For a term-frequency of one (i.e. $x = 1$) a negative score can be only be achieved when the document length (y) is approximately 8.5 times longer than the average document length and thus this violation has not affected the collections used in this research. This phenomenon can be easily eliminated by that ensuring that $ntf(0, y) = 0$ for the term-frequency function (7) or by re-adjusting the $penalty(q, d)$ function. One such function that goes through the origin ($ntf(0, y) = 0$) and has a similar term-frequency influence to (7) is $\frac{x}{x+0.45}$. Nevertheless, these characteristics were not present in the training collection. This is not of major consequence for the ranking on the test collections used herein.

that indicates the strength of each different aspect (i.e. dimension of the vector) of the topic.

Simply using the vector length for normalisation might seem an intuitive approach after considering such an argument. However, using the vector length as the normalisation factor will lead to a violation of constraint 2. Consider a non-query term, which has already appeared in the document. If this term re-occurs, the weight of the document will not decrease as the vector length remains unchanged.

Following from this, the above constraint avoids over-penalising longer documents by ensuring that the normalisation aspect (measured in repeated terms) is sublinear. Therefore, as *non-query terms* appear in a document they should be penalised less with successive occurrences. Essentially, the inverse of the score reduction due to non-query terms being added should be sub-linear. The normalisation used in term-weighting schemes is inversely related to the weight to apply and as such is typically the denominator in such functions.

As previously mentioned, the better learned normalisation schemes satisfy this constraint for the collections used within. Due to a possible negative result for certain positive values in equation 8, the inverse of the weight change due to adding a non-query term can be exponential in certain circumstances. However, these cases are not typical and do not tend to affect the retrieval of documents in the higher ranks (i.e. 1000) as previously mentioned.

It is worth noting that none of the three existing constraints were enforced by the way we created our framework. Instead, they are characteristics of the best schemes developed using the evolutionary learning approach. This reinforces the validity of the existing constraints and interestingly can be used to identify new constraints.

4.4 Violations of New Constraint

Interestingly, the new constraint developed (constraint 4) is violated by both $BM25$ and the pivoted normalisation scheme. Basically, $\forall x, y > 0\ (ntf(x, y+1) - ntf(x, y))^{-1} > (ntf(x, y+2) - ntf(x, y+1))^{-1}$ must be true for the constraint to be satisfied. It can be seen that this is not true when the following normalisation function is used in both $BM25$ and the pivoted normalisation scheme (for any value of b):

$$n_b = ((1-b) + b \cdot \frac{dl}{dl_{avg}})$$

as this is linear with respect to dl. This analysis suggests that when using this function, b needs to be tuned on each specific collection. If the collection contains some long documents compared to the average document length, it would be important to have a low value for b as it would otherwise unfairly penalise these longer documents. It is also interesting that if the normalisation function ($n(dl)$) is sublinear with respect to dl and the term-frequency is normalised using $\frac{tf_t^D}{n(dl)}$ (which is used in our framework), then the first constraint (constraint 1) will always be satisfied.

Table 1: Document Collections

Name	Collection	#docs	dl_{avg}	$\sigma(dl)$
FR	Federal Register (1994)	55,630	387.1	1365.2
LATIMES	LA Times	131,896	251.7	251.9
FT	Financial Times (1991-1993)	138,668	221.8	196.4
FBIS	FBIS	130,471	249.9	554.4
OHSU	OHSUMED (1990-1991)	148,162	81.4	64.0

Table 2: Topics

Topic Range	Average Topic Length		
	short	med	long
301-350	2.4	12.44	43.92
351-400	2.4	10.46	32.96
401-450	2.4	9.02	27.5
301-400	2.4	11.45	38.44
1-63 (OHSU)	2.2	5.11	None

5. EXPERIMENTAL SETUP

Now that we have determined that the learned weighting function conforms to all the existing constraints, it is worth empirically validating this analysis. These experiments will also test the validity of the new normalisation constraint (constraint 4). In this section, we introduce an experimental methodology to test all three schemes ($BM25$, pivoted normalisation and the learned scheme) on test data.

5.1 Document Collections

We use collections from TREC disks 4 and 5 as test collections. For each set of topics we create a short query set, consisting of the title field of the topics, a medium length query set, consisting of the title and description fields, and a long query set consisting of the title, description and narrative fields. We also use documents 1990 and 1991 from the OHSUMED collection as a test collection. We created short queries for the OHSUMED collection by simply removing terms from the description field. Standard stop-words from the Brown Corpus[5] are removed and remaining words are stemmed using Porter's algorithm. No additional words are removed from the narrative fields as is the case in some approaches. Tables 1 and 2 shows some characteristics of the documents and topics used in this research.

5.2 Benchmarks

We tuned the $BM25$ scheme (1) using two values of k_1 (1.2 and 2.0, as they are the most commonly reported in the literature) and nine values of b (0, 0.125, 0.25, 0.375, 0.5, 0.625, 0.75, 0.825 and 1) for each query type (short, medium and long). The best performing k_1 value on all collections was the default value of 1.2 which we used for all our experiments. The best values of b for short, medium and long queries on our test data were 0.125, 0.375 and 0.625 repectively which we used when dealing with these query types.

We also manually tuned the pivoted normalisation function using seven values of s (0, 0.025, 0.05, 0.1, 0.2, 0.3 and 0.4) for each query type. Values greater than 0.4 for s can typically violate some of the constraints [13]. The best values of s for short, medium and long queries on our test data were 0.025, 0.05 and 0.2 repectively.

It has been stated that the *idf* in the $BM25$ scheme will often lead to poor results due to a possible negative weight

[5]http://www.lextek.com/manuals/onix/stopwords1.html

for certain frequent terms [12, 13]. As we removed standard stopwords this did not occur for long (verbose) queries. For the $BM25$ scheme, our long queries achieve a MAP which is higher in most cases than those of the medium length queries.

6. RESULTS

Tables 3, 4 and 5 show the results for short, medium and long queries on multiple test collections. As our results show, the learned LRF scheme is comparable to, and often better than, a tuned $BM25$ function and a tuned pivoted normalisation scheme.

Table 3: MAP for short queries

Collection	Topics	#Topics	$Piv_{0.025}$	$BM25_{0.125}$	LRF
FR	301-400	45	0.2813	0.2897	0.3089
FT	351-400	47	0.2379	0.2438	0.2488
FBIS	351-400	38	0.1989	0.2053	0.2096
FT	401-450	49	0.3174	0.3277	0.3243
LATIMES	401-450	45	0.2408	0.2435	0.2704
OHSU	1-63	63	0.2322	0.2380	0.2558

Table 4: MAP for medium queries

Collection	Topics	#Topics	$Piv_{0.05}$	$BM25_{0.375}$	LRF
FR	301-400	45	0.3063	0.2944	0.3212
FT	351-400	47	0.2320	0.2330	0.2489
FBIS	351-400	38	0.2118	0.2353	0.2408
FT	401-450	49	0.3550	0.3572	0.3575
LATIMES	401-450	45	0.2394	0.2523	0.2675
OHSU	1-63	63	0.2540	0.2738	0.3005

Table 5: MAP for long queries

Collection	Topics	#Topics	$Piv_{0.2}$	$BM25_{0.625}$	LRF
FR	301-400	45	0.2913	0.3407	0.3809
FT	351-400	47	0.2713	0.2943	0.3142
FBIS	351-400	38	0.1759	0.2017	0.2400
FT	401-450	49	0.3449	0.3622	0.3790
LATIMES	401-450	45	0.2640	0.2855	0.2829

For all query lengths (without the need for tuning) our new learned formula compares favourably to the best manually tuned benchmark. On most collections for different query lengths our new learned formula outperforms the manually tuned benchmark. These results are encouraging particularly as the new weighting function is nonparametric. This adds further empirical evidence to support our normalisation constraint (constraint 4).

6.1 Statistical tests

A one-tailed t-test did not report a statistically significant increase ($p\text{-}value < 0.05$) over the tuned $BM25$ on many of the collections tested. However, due to the consistent increases (albeit small in some cases) especially for long queries, we pooled the same queries types from different collections together. To validate this pooling, we performed a one-way ANOVA on the differences in average precision between LRF and the tuned $BM25$ for each query type. It showed that there is no variance between the differences in AP across the collections for a specific query type. Thus, the difference in performance across the different collections does not vary in any statistical sense ($F\text{-}value < 1$ and $p\text{-}value > 0.05$).

It can be seen from Table 6 that our new learned formula (LRF) achieves a significantly higher MAP than $BM25$ for all query types, although for short and medium queries that difference is only about 1.3%. However, for long queries we can see that there is a significant increase in MAP of about 2.1% MAP, compared to a tuned $BM25$ scheme. It is also worth noting that if a single $BM25$ scheme was chosen for all query types, the LRF function would outperform it by a greater margin.

Table 6: Significance tests for LRF and ΔMAP

Query Type	#Topics	Piv	ΔMAP	$BM25$	ΔMAP
		$p\text{-}value$		$p\text{-}value$	
Short	287	0.001	0.0196	0.016	0.0130
Medium	287	0.001	0.0243	0.037	0.0132
Long	224	0.001	0.0301	0.035	0.0210

6.2 Discussion

It can be seen that the learned scheme achieves a high MAP on many collections and varying query lengths. It compares favourably on unseen data against a highly tuned $BM25$ scheme. As $BM25$ and the pivoted normalisation scheme violate our newly developed constraint (constraint 4), the performance for any one value of b or s will be sub-optimal for different collections and over different query lengths. We can see that the optimal settings for $BM25$ and the pivoted normalisation scheme tends to increase as the query length increases. This phenomenon has previously been reported [14, 7]. It is also worth noting that the most optimal setting also varies for each collection for queries of the same length [6].

This fourth normalisation constraint (constraint 4) fits neatly into the axiomatic framework previously developed and interestingly is a characteristic of the best evolved normalisation functions. Other normalisation functions reported in the literature [1, 2] adhere to the new normalisation constraint. However, many of these functions contain tuning parameters and need some tuning for different queries and collections.

7. CONCLUSION

We have analysed a term-weighting scheme that has been developed using an evolutionary learning technique. This weighting scheme is decomposed into document and query growth functions using the axiomatic framework and is shown to satisfy previously known constraints. This purely empirical learning approach further validates the correctness of the existing constraints. We develop a new normalisation constraint to which our learned scheme adheres.

We show that the normalisation influence component of the standard benchmark schemes requires much tuning on specific collections and specific query lengths in order to achieve high performance. Our experiments show that our newly developed scheme outperforms the manually tuned benchmarks on most of the collections tested without the need for tuning. An interesting future direction would be to constrain the search space using the existing axioms and then use a learning technique to search this reduced space.

8. ACKNOWLEDGMENTS

This work is being carried out with the support of IRCSET (the Irish Research Council for Science, Engineering

and Technology) under the Embark Initiative. The authors would also like to thank the anonymous reviewers for their useful comments.

9. REFERENCES

[1] Gianni Amati and Cornelis Joost Van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, 20(4):357–389, 2002.

[2] Gianni Amati and C. J. van Rijsbergen. Term frequency normalization via pareto distributions. In *Proceedings of the 24th BCS-IRSG European Colloquium on IR Research*, pages 183–192, London, UK, 2002. Springer-Verlag.

[3] Brian T. Bartell, Garrison W. Cottrell, and Richard K. Belew. Learning the optimal parameters in a ranked retrieval system using multi-query relevance feedback. In *Symposium on Document Analysis and Information Retrieval*, 1994.

[4] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 89–96, New York, NY, USA, 2005. ACM Press.

[5] Yunbo Cao, Jun Xu, Tie-Yan Liu, Hang Li, Yalou Huang, and Hsiao-Wuen Hon. Adapting ranking svm to document retrieval. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 186–193, New York, NY, USA, 2006. ACM Press.

[6] Abdur Chowdhury, M. Catherine McCabe, David Grossman, and Ophir Frieder. Document normalization revisited. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 381–382. ACM Press, 2002.

[7] Tze Leung Chung, Robert Wing Pong Luk, Kam Fai Wong, Kui Lam Kwok, and Dik Lun Lee. Adapting pivoted document-length normalization for query size: Experiments in chinese and english. *ACM Transactions on Asian Language Information Processing (TALIP)*, 5(3):245–263, 2006.

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

[9] Ronan Cummins and Colm O'Riordan. A framework for the study of evolved term-weighting schemes in information retrieval. In Benno Stein and Odej Kao, editors, *TIR-06 Text based Information Retrieval, Workshop. ECAI 2006*, Riva del Garda, Italy, 29 August 2006.

[10] Ronan Cummins and Colm O'Riordan. Term-weighting in information retrieval using genetic programming: A three stage process. In *The 17th European Conference on Artificial Intelligence, ECAI-2006*, Riva del Garda, Italy, August 28th - September 1st 2006.

[11] Weiguo Fan, Michael D. Gordon, and Praveen Pathak. A generic ranking function discovery framework by genetic programming for information retrieval. *Information Processing & Management*, 2004.

[12] Hui Fang, Tao Tao, and ChengXiang Zhai. A formal study of information retrieval heuristics. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 49–56. ACM Press, 2004.

[13] Hui Fang and ChengXiang Zhai. An exploration of axiomatic approaches to information retrieval. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 480–487. ACM Press, 2005.

[14] Ben HE and Iadh Ounis. A study of parameter tuning for term frequency normalization. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 10–16. ACM Press, 2003.

[15] H. S. Heaps. *Information Retrieval: Computational and Theoretical Aspects*. Academic Press, Inc., Orlando, FL, USA, 1978.

[16] N. Oren. Re-examining tf.idf based information retrieval with genetic programming. *Proceedings of SAICSIT*, 2002.

[17] Stephen E. Robertson, Steve Walker, Micheline Hancock-Beaulieu, Aarron Gull, and Marianna Lau. Okapi at TREC-3. In *In D. K. Harman, editor, The Third Text REtrieval Conference (TREC-3) NIST*, 1995.

[18] Dmitri Roussinov, Weiguo Fan, and Fernando A. Das Neves. Discretization based learning approach to information retrieval. In *CIKM*, pages 321–322, 2005.

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

[20] Shuming Shi, Ji-Rong Wen, Qing Yu, Ruihua Song, and Wei-Ying Ma. Gravitation-based model for information retrieval. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 488–495. ACM Press, 2005.

[21] Amit Singhal, Chris Buckley, and Mandar Mitra. Pivoted document length normalization. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 21–29. ACM Press, 1996.

[22] Fei Song and W. Bruce Croft. A general language model for information retrieval (poster abstract). In *Research and Development in Information Retrieval*, pages 279–280, 1999.

[23] Andrew Trotman. Learning to rank. *Information Retrieval*, 8:359 – 381, 2005.

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