Analysing Ranking Functions in Information Retrieval Using Constraints

Ronan Cummins
School of Computing Science,
University of Glasgow, United Kingdom

Colm O'Riordan
Department of Information Technology,
National University of Ireland, Galway, Ireland

1 Introduction

Text-based information retrieval (IR) systems deal with natural language documents and queries. They attempt to limit the problem of 'information overload' by automatically returning only documents that are relevant to a users information need (query). Ambiguity, for example, inherent in natural language, is one such problem that automated systems have difficulties in resolving.

Current web search engines have benefited greatly from the early research into text-based information retrieval and library systems. Term-weighting functions are one of the most important parts of a web search engine (or indeed any information retrieval system) as they aim to rank relevant items before non-relevant items. Fundamentally, it is the quality of these term-weighting functions that determines the usefulness of the system. Many approaches to term-weighting have been developed over the years. These term-weighting functions have been produced from various types of models, ranging from empirically-based learning models to purely theoretical models.

An axiomatic approach to IR [Fang & Zhai, 2005] has previously been developed that refines a number of constraints (axioms) [Fang et al., 2004] to which all *good* weighting functions should adhere. This approach and, in particular, the constraints developed therein, are useful in attempting to theoretically motivate term-weighting functions that are developed from purely automated learning (empirically-based) models. Often the aim of these purely automated learning models is to learn a function that best 'fits' the training data, while ensuring some generality. It is important that we can explain the output of such learning algorithms so we can better understand retrieval in general. In particular, we believe that the better functions produced from these automatic learning approaches to IR should adhere to these existing constraints, as the satisfaction of the constraints serve as a useful guide to the optimality of the solutions produced.

In this chapter, we aim to show that these axioms, that are deemed valid in a *bag of words* model, can be used to accurately estimate the performance of term-weighting functions. The unconditional satisfaction of the constraints have been shown to serve as a useful guide to their performance [Fang & Zhai, 2005, Cummins & O'Riordan, 2007a]. These constraints can potentially be used in a number of different ways.

They can be used, as they are in this chapter, to validate the 'correctness' of a specific retrieval model by showing that any term-weighting scheme produced therein adheres to them. However, they could also be used to constrain the space of term-weighting functions so that a specific learning approach searches a much smaller space, where it is known *good* that term-weighting schemes lie. However, it has not been shown how often a constraint may be violated in a standard retrieval setting, if the constraint cannot be unconditionally satisfied by a particular term-weighting scheme. Furthermore, the number of violations that occur in an actual retrieval setting could potentially be used as an evaluation metric, if the number of constraint violations is correlated to performance. This is interesting as it does not rely on relevance judgments and relies purely on the axioms developed from an inductive approach, which closely models the notion of relevance.

We review four constraints (deemed valid in a *bag of words* model) and describe how term-weighting functions should be modelled in order to minimise the possible ways in which a function could violate these constraints. We develop properties that can be derived from the four initial constraints (axioms) and that help to describe how the axioms can constrain the constitution of state of the art term-weighting functions. We outline a method which adopts an inductive approach to measure the number of actual constraint violations for a number of state of the art term-weighting functions on standard test collections for ad hoc retrieval. We find that the number of constraint violations that occur for many of the axioms is higher than was stated in the original works. Finally, we show that the best performing term-weighting function (which was developed using a learning approach) does not violate the constraints as often as other functions.

2 Background and Related Research

The goal of a term-weighting function is to score a document for a given user query, and therefore, is crucial in most, if not all, information retrieval (IR) systems. There have been numerous models developed over the years that have yielded many different term-weighting functions. Some of these include: vector space models [Salton et al., 1975], probabilistic models [Robertson et al., 1995], language models [Ponte & Croft, 1998], divergence from randomness models [Amati & van Rijsbergen, 2002], learning models [Fuhr & Buckley, 1991, Radlinski & Joachims, 2005] and some other more unconventional models [Shi et al., 2005]. However, many, or all, of these approaches consists of the aggregation of weights applied to the terms in common with the document and query. Once all documents are scored with respect to a given query, the list of document scores is sorted and the top N documents are returned to the user. These approaches are often called *bag of words* models as they do not take into account the complex interactions and interdependence of terms (although term-dependence may more easily be incorporated into some of the models in a more intuitive manner). The simple 'bag of words' representation has survived over the years mainly because it is easily implemented and more importantly, it is very difficult to outperform this simple representation in terms of effectiveness (usually measured using variations of precision and recall).

In a number of works, term-weighting schemes are described using two triples [Salton & Buckley, 1988, Zobel & Moffat, 1998]. One triple describes the weight assigned to the terms in the document, while the second triple describes the weight assigned to the terms in the query. Each triple contains a term-discrimination element, a term-frequency element and a form of normalisation. However, with the advent of TREC data [Harman, 1993], it has been noted that the triple describing the weight assigned to terms in the query can be reduced to a simple linear within-query term-frequency [Singhal, 2001, Fang & Zhai, 2005]. The framework outlined here is also consistent with this view of a term-weighting scheme.

The following function (equation (1)) can be thought of as a generalisation of a family of term-weighting schemes. While it does not represent the complete space of entire term-weighting schemes (which is boundless), it does incorporate most, if not all, term-weighting schemes reported in the literature. The score (S())

of a document D in relation to a query Q can be calculated as follows:

$$S(Q,D) = \sum_{t \in Q \cap D} (ntf(D) \cdot gw(t) \cdot tf_t^Q) \quad (1)$$

where $ntf()$ is a normalised term-frequency, $gw(t)$ is a term-discrimination factor and tf_t^Q is the frequency of term t in the query Q. In this framework, there is a basic term-discrimination element (or global component), a normalised term-frequency element (or within-document component) and a query term element (or within-query component). The term-discrimination element ($gw(t)$) aims to determine the usefulness of a search term by using characteristics of the term in the collection as a whole. Typically, terms that occur in fewer documents are given a higher weight as they tend to be better descriptors of that document. Most term-weighting approaches include some type of term-discrimination element either directly or indirectly in order to promote terms that are likely to be better able to identify certain documents. The normalised term-frequency ($ntf(D)$) aims to provide two effects on each specific document D using within-document measures. Its first aim is to promote documents that have a higher occurrence of query terms. This is achieved using a term-frequency influence component. It is intuitive that a document with more occurrences of query terms should be ranked higher than a document with fewer occurrences. However, not all documents are of similar length and thus, the term-frequency is normalised in some way to avoid over-weighting longer documents simply because they contain more of these terms. A document that is longer may simply have a broader topic and should not be promoted over shorter documents which may be more concise and preferable to the user. Basically, the concept of normalisation is a measure of the concentration of query terms in a document. Documents with a greater concentration of query terms should be promoted ahead of documents with a lower concentration. The remaining component of the framework (equation (1)) describes the weight assigned to the terms appearing in the actual query. This component is typically a simple description and it has been shown in many studies that using the actual query term-frequency for such a component does not lead to any decrease in performance compared to a more complex form for this component [Fang & Zhai, 2005]. This is typically because queries are quite short and supply a limited amount of information about the frequencies and characteristics of the terms themselves.

Much research has focused on combining these three heuristic factors into effective term-weighting schemes. There have been many attempts to exhaustively search a limited space of term-weighting functions [Salton & Buckley, 1988, Zobel & Moffat, 1998]. These approaches to developing retrieval functions are unlikely to produce any substantial increase in performance, as there is no guarantee that existing parts of functions (which are limited in form at such a coarse level) can be effectively combined (in an ad hoc manner) to create a high performance weighting function. Another point worth mentioning is that for such exhaustive searches, the parts of the functions to be combined must be quite complex (non-atomic) in order to render the search space tractable. The divergence from randomness model (DFR) [Amati & van Rijsbergen, 2002] has been developed by combining these three factors into a term-weighting scheme in a more theoretical framework. This work creates a number of term-weighting functions based on different models for the distribution of terms in a document and collection. Nevertheless, the search space of term-weighting schemes is so large that an exhaustive search of the entire space is infeasible (if not impossible).

3 Constraining Term-Weighting Functions

We will briefly outline four axioms for term-weighting and then discuss how modern term-weighting schemes can be constrained by these.

3.1 Axioms for a *Bag of Words* Model

A number of axioms have been previously postulated and these can be used to validate or to develop term-weighting schemes in a constrained space [Fang & Zhai, 2005]. Thus, we use the terms axiom and constraint analogously in this paper. We will briefly introduce some constraints that were developed using an inductive framework [Fang & Zhai, 2005]. The idea of this inductive framework is to define a base case function 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 two separate functions. A *document growth function* describes the change in the document score when a single term is added to the document, while a *query growth function* describes the change in the document score when a single term is added to the query. This is an elegant approach to formalising necessary characteristics of *good* term-weighting functions. This inductive approach may more accurately model the human process of determining the relevance of a document, as one can imagine that person's notion of relevance changes, when terms that are either on or off-topic are encountered, during a linear reading process.

Assume $S(Q,D)$ is a function which scores a document D in relation to a query Q in a standard *bag of words* retrieval model. With notation similar in style to [Fang & Zhai, 2005], the constraints can be formalised as follows, where $t \in T$ is a term t in the set of terms in a corpus and $\delta_t(t,D,Q) = S(Q,D \cup \{t\}) - S(Q,D)$ (i.e. the change in score as t is added to the document D):

Constraint 1: $\forall Q,D$ and $t \in T$, if $t \in Q$, $S(Q,D \cup \{t\}) > S(Q,D)$

The first constraint (constraint 1) states that adding a new query term to the document must *always* increase the score of that document. This seems intuitive for all terms as no matter how little information content is in a term, if it occurs in the document, it indicates that it is closer (possibly negligibly) to the topic of the query.

Constraint 2: $\forall Q,D$ and $t \in T$, if $t \notin Q$, $S(Q,D \cup \{t\}) < S(Q,D)$

The second constraint (constraint 2) states that adding a non-query term to a document must *always* decrease the score of that document. Again, this constraint seems intuitive as it ensures that document with more off-topic terms will be assigned a lower score. As more off-topic terms are encountered the score of the document should be reduced.

Constraint 3: $\forall Q,D$ and $t \in T$, if $t \in Q$, $\delta_t(t,D,Q) > \delta_t(t,D \cup \{t\},Q)$

The third constraint (constraint 3) states that adding successive query terms to a document should increase the score of the document less with each successive addition. The intuition behind this constraint is that it is ultimately the first occurrence of a term that indicates that the document is on-topic (i.e. related to the query). Due to characteristics of natural language, it is known that when a term first appears in a document, the likelihood of re-appearance increases. Thus, the weight given to successive occurrences of a query term should be reduced. This is due to authors repeatedly using similar terms to identify similar topics within a document.

A fourth constraint can be formalised as follows, where $t \in T$ is a term in the set of terms in a corpus and $\delta_t^{-1}(t,D,Q) = S(Q,D \cup \{t\})^{-1} - S(Q,D)^{-1}$ (i.e. the change in inverse score as t is added to the document D):

Constraint 4: $\forall Q,D$ and $t \in T$, if $t \notin Q$, $\delta_t^{-1}(t,D,Q) > \delta_t^{-1}(t,D \cup \{t\},Q)$

The fourth constraint (constraint 4) states that adding more non-query terms to a document should decrease the score of a document less with each occurrence. According to Heaps' law [Heaps, 1978], the appearance of new unseen terms in a corpus grows in roughly a square-root relationship (sub-linearly) to the document length (in words). Therefore, as *non-query terms* appear in a document they should be penalised less with successive occurrences. This constraint avoids over-penalising longer documents by ensuring that the normalisation aspect is sub-linear. 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 occurrences (term-frequency) of these unique terms that indicates the strength of each different aspect (i.e. dimension of the vector) of the topic. As non-query terms appear the topic of the document does not drift from the query linearly. This again is due to authors using the same words repeatedly to identify similar topics in a document. As such, it is the first appearance of a non-query term that ultimately indicates a change in the topic of a document and successive occurrences of this term does not indicate that the topic of that document is drifting from the query topic to the same degree.

3.2 Analysis of Constraints

We have not, as yet, indicated how the inductive approach and the axioms developed therein constrain parts of modern term-weighting functions. In this section, we will map the characteristics of state-of-the-art term-weighting functions to the constraints previously outlined. The following are some necessary properties of term-weighting schemes that can be deduced from the axioms. These properties (labelled **P**) are also derived from the characteristics of the features used in term-weighting schemes and the characteristics of the way in which the three parts of a term-weighting scheme (described earlier in section 2) interact.

P1: The measure of information content (typically some type of idf) of a query-term that occurs should never be assigned a value that is lower than the score assigned to a query-term not occurring (for most term-weighting approaches this is 0). Therefore, for most term-weighting schemes (and those included in this work), the term-discrimination factor should never be assigned a negative value. If this property is not present, it will lead to violations of constraints 1 and 3, as a negative score will lead to a decrease in the score of a document when a query term occurs. It can also lead to violations of constraints 2 and 4 because, in certain cases, the document score will be negative, and thus normalisation will actually increase the document score.

P2: A term-frequency aspect must be present and must always be positively increasing. If this property is not present it will lead to violations of constraints 1 and 3.

P3: There must be some type of normalisation aspect present. There must be some method of penalising the score of a document for the occurrences of non-query terms. This could be achieved by ensuring the score of a document (or term) is inversely proportional to the length of a document (as is the case in most term-weighting schemes) or simply by subtracting some weight for each occurrence of a non-query term. In either case if this property is not present it will lead to violations of constraints 2 and 4.

P4: The normalisation factor must be measured in repeated terms (i.e. must contain a measure that reduces the score for *every* non-query term). If, for example, a coarse measure of document length is used for normalisation (e.g. the document vector length), repeated occurrences of non-query terms would not be penalised. Thus, this property ensures that the length measure used in normalisation is granular and leads to a more refined normalisation. If this property is not present it will lead to violations of constraints 2 and 4.

P5: The actual term-frequency should be normalised instead of the term-frequency function. For example, consider $S1(Q,D)$ and $S2(Q,D)$ which describe two possible ways of applying normalisation in a term-weighting function.

$$S1(Q,D) = \sum_{t \in Q \cap D} \left(\frac{tff(tf_t^D)}{n()} \cdot gw(t) \cdot tf_t^Q\right) \tag{2}$$

where $gw(t)$ is the term-discrimination aspect, $tff()$ is the term-frequency function, $n()$ is some normalisation aspect and tf_t^D is the actual term-frequency of t in D. Other functions penalise the actual term-frequency as follows:

$$S2(Q,D) = \sum_{t \in Q \cap D} \left(tff\left(\frac{tf_t^D}{n()}\right) \cdot gw(t) \cdot tf_t^Q\right) \tag{3}$$

The first method of normalisation presented ($S1(Q,D)$) violates constraints 1 and 3, as the normalisation ($n()$) is independent of the term-frequency function ($tff()$) and, therefore, it may grow to such a degree that the penalisation more than outweighs the increase in weight that the term-frequency provides. Consider these two somewhat similar methods of applying normalisation (i.e. $S1(Q,D)$ and $S2(Q,D)$) from an inductive perspective. Let x define the term-frequency for a query term. Consider a document that consists of successive occurrences of this term. In such a case, x also defines the document length. Let $log(x)$ be the term-frequency factor and $\sqrt{x}$ be the normalisation aspect. In isolation they would appear to adhere to the aforementioned constraints (i.e. the term-frequency is sub-linear and the normalisation is sub-linear). Figure 3 shows that $S1(Q,D)$ (i.e. $log(x)/\sqrt{x}$) does not always increase for successive occurrences of the query-term x. $S2(Q,D)$ does adhere to this constraint in this simple inductive case. Thus, when normalisation is explicitly used to penalise documents, it should be applied to the actual term-frequency as in $S2(Q,D)$ (i.e. $log(x/\sqrt{x})$) to help satisfy constraint 1 for this simplest inductive case.

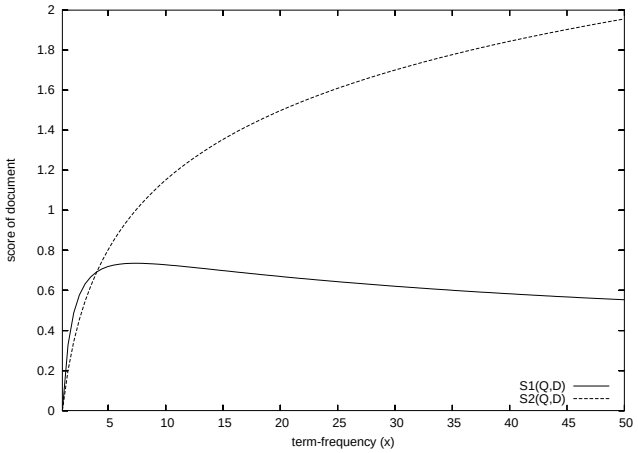

Figure 1: Violation of constraint 1 for Property P5

P6: The term-frequency aspect must be sub-linear (as more of the same query terms occur, they must increase the score of the document less with each successive occurrence). A scheme which does not exhibit this property will violate constraint 3.

P7: The normalisation aspect must be sub-linear (as more non-query terms occur, the penalisation must become smaller). A scheme which does not exhibit this property will violate constraint 4.

P8: The increase in score due to a query-term being added must be greater than the penalisation due to the document increasing in length. This is the only property that cannot unconditionally be enforced by most modern term-weighting schemes. This will lead to violations of constraints 1 and 3. Due to the nature of the normalisation schemes used in modern retrieval functions, when a term-weighting scheme uses the document length explicitly to penalise the document, constraint 1 (and consequently constraint 3) can *never* be satisfied unconditionally. Consider the case where a term with an extremely low idf value (i.e. where the term has negligible semantic content) is added to a document. The penalisation due to the document increasing in length will more than offset the increase in weight as the term is added (as all existing terms in the document are penalised by the document length accordingly). For many schemes this will only tend to happen for terms with a very low information content (ie. term-discrimination scheme).

Figure 2 shows a set of query terms with some basic weights assigned to them. Document 1 contains

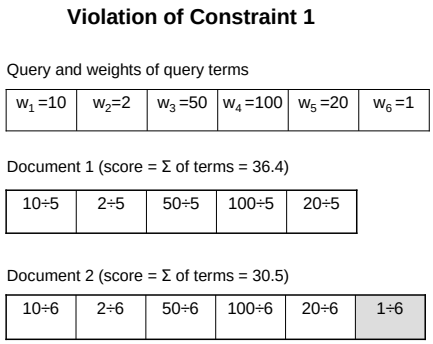

Figure 2: Violation of constraint 1 for P8

5 distinct query terms while document 2 contains 6 distinct query terms. The normalisation function, in the example, uses the document length. The normalisation (division by the document length) reduces the weight of *all* of the existing terms in the document and therefore, the score of a document may not increase as a query term is added. In the example shown, the score of document 1 is calculated by summing up the scores of the 5 query terms (36.4). The score of document 2 is calculated similarly (summing up the 6 query terms). As the query term, added to document 2, has a very low term-discrimination weight ($w_6 = 1$) compared to the other query terms, the increase in weight due to this query term being added does not offset the increase in penalisation. The score of document 2 is only 30.5, although document 2 is created by adding a query term to document 1. However, the potential for violations of the type just described may be more prevalent in different types of term-weighting schemes. While these violations cannot be prevented, it may be possible to minimise the violations that occur due to this phenomenon, by using different term-discrimination measures. Interestingly, this property constrains the interaction between the measure of information content and normalisation in some way. We will now briefly look at state of the art term-weighting functions and indicate the properties they contain.

4 Term-weighting Analysis

In this section, we present two different ways of determining the constraint violations of term-weighting schemes. Firstly in Section 4.1, we present a number of term-weighting scheme and analytically analyse them (similarly to [Cummins & O'Riordan, 2007a]) to determine if they unconditionally adhere to the constraints. Secondly in Section 4.2, we outline a method whereby, the number of constraint violations in an actual retrieval setting are counted.

4.1 Strict Satisfaction of Constraints

In this section we present several term-weighting schemes and briefly analyse them to determine their satisfaction to the constraints by identifying which schemes contain the aforementioned properties.

4.1.1 Pivoted Document Length Normalisation

The pivoted document length normalisation approach [Singhal et al., 1996] is often used to weight term in the vector space model [Salton et al., 1975] and is defined as follows:

$$PIV(Q,D) = \sum_{t \in Q \cap D} \left(\frac{1 + log(1 + log(tf_t^D))}{(1-s) + s \cdot \frac{dl}{dl_{avg}}} \cdot log\left(\frac{N+1}{df_t}\right) \cdot tf_t^Q\right) \quad (4)$$

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 measured in non-unique terms. N is the number of documents in the collection and df_t is the number of documents in which term t appears. The tuning parameter, s, is used to tune the normalisation component and has a default value of 0.2.

The normalisation function used in the pivoted document length normalisation scheme normalises the term-frequency function and not the actual term-frequency directly. Therefore, property P5 is not present. The normalisation function used is also linear and thus property P7 is not present. As a result, violations of constraints 1, 3 and 4 can occur due to properties P5, P7 and P8 not being present for this scheme.

4.1.2 *BM*25

The *BM*25 weighting scheme [Robertson et al., 1995] 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:

$$BM25(Q,D) = \sum_{t \in Q \cap D} \left(\frac{tf_t^D}{tf_t^D + k_1 \cdot ((1-b) + b \cdot \frac{dl}{dl_{avg}})} \cdot log\left(\frac{N - df_t + 0.5}{df_t + 0.5}\right) \cdot tf_t^Q\right) \quad (5)$$

where 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 [Robertson et al., 1995] but has been used successfully in many studies [Fang & Zhai, 2005]. The document normalisation influence tuning parameter, b, has a default value of 0.75.

The idf used in this scheme can assign negative values for terms with a low information content and thus P1 is not present. The normalisation used is also linear and thus property P7 is not present. Therefore, violations of constraints 1, 2, 3 and 4 will occur due to properties P1, P7 and P8 not being present in this scheme.

4.1.3 Modified BM25 Analysis

A modified BM25 scheme (MBM25) can be created by replacing the idf factor used in the *BM*25 scheme with the idf factor used in the pivoted document length normalisation scheme. This scheme should break less constraints than the original BM25 scheme as only properties P7 and P8 are not present.

4.1.4 Evolutionary Learned Scheme

An incremental evolutionary learning approach [Cummins & O'Riordan, 2007b] which develops an entire weighting function has previously been explored. The search space is separated into three parts. Firstly, schemes are learned that 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 Mean Average Precision (MAP), the term-frequency aspect of the scheme is learned while the term-discrimination (i.e. some type of idf) 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.

This process results in a complete term-weighting scheme. Although the shape of the possible function is constrained by the manner in which the three aspects of a term-weighting function are combined, the form (shape) of the constituent functions are not constrained by the aforementioned constraints (axioms). Therefore, this approach (which is data-driven) is only constrained by the representation used within the learning algorithm, and is driven purely by fitness (i.e. performance). The following term-weighting function [Cummins & O'Riordan, 2007b] was developed by this evolutionary learning approach:

$$ES(D,Q) = \sum_{t \in Q \cap D} \left(\frac{tf_t^D}{tf_t^D + 0.45 \cdot \sqrt{\frac{dl}{dl_{avg}}}} \cdot \sqrt{\frac{cf_t^3 \cdot N}{df_t^4}} \cdot tf_t^Q\right) \tag{6}$$

Again, this formula contains the three term-weighting components outlined earlier, and it should be noted that there are no tuning parameters in this function. The term-discrimination scheme is always positive. The actual term-frequency is normalised as $tf_t^D / tf_t^D + 0.45 \cdot \sqrt{\frac{dl}{dl_{avg}}}$ can be re-written as $(tf_t^D / \sqrt{\frac{dl}{dl_{avg}}}) / (tf_t^D / \sqrt{\frac{dl}{dl_{avg}}}) + 0.45$. The normalisation scheme used is sublinear, as is the term-frequency function. Therefore, violations of constraints 1 and 3 can occur only because property P8 is absent. All other properties (P1-P7) are present.

4.1.5 Divergence From Randomness (DFR)

One of the best performing term-weighting functions, produced from the DRF approach, as outlined in [Amati & van Rijsbergen, 2002] is the following:

$$DFR(Q,D) = \sum_{t \in Q \cap D} \left(\frac{tf_t^D \cdot log(1 + \frac{dl_{avg}}{dl})}{1 + tf_t^D \cdot log(1 + \frac{dl_{avg}}{dl})} \cdot log\left(\frac{N+1}{df_t + 0.5}\right) \cdot tf_t^Q\right) \tag{7}$$

This term-weighting scheme also has no tuning parameters (although in some studies a tuning parameter c has been introduced into the normalisation component to improve performance). Violations of constraints 1 and 3 can occur, as property P8 is not present. All other properties (P1-P7) are present.

While we have described which constraints are satisfied unconditionally and which constraints may be violated, it is not indicated how often these constraints will be violated on a standard test collection. Previous research [Fang & Zhai, 2005, Cummins & O'Riordan, 2007a] has shown that the strict adherence to the constraints is a useful guide to constructing effective term-weighting schemes.

4.1.6 Language Modelling Scheme

A language modelling approach to information retrieval has also been successful in developing high performance weighting functions. The following function [Fang et al., 2004] is an example of one such function based on dirichlet priors:

$$LM(Q,D) = \sum_{t \in Q \cap D} log\left(\frac{P_s(q_i|d)}{\alpha_d \cdot P(q_i|C)}\right) + n \cdot log(\alpha_d) \tag{8}$$

where $|C|$ is the number of terms in the collection, $P(q_i|C) = cf_t/|C|$, $P_s(q_i|d) = (tf + u \cdot P(q_i|C))/(dl + u)$, $u = 2000$ and $\alpha_d = u/(dl+u)$. Violations of constraints 1 and 3 can occur only because property P8 is absent. The normalisation is not sublinear and therefore constraint 4 is violated.

4.1.7 Axiomatic Term-Weighting Scheme

A term-weighting function ($F2EXP$) that was developed in conjunction with the original axioms has also been developed and shown to achieve a very high performance on a number of test collections.

$$F2EXP(Q,D) = \sum_{t \in Q \cap D} \left(\frac{tf_t^D}{tf_t^D + 0.5 + 0.5 \cdot \frac{dl}{dl_{avg}}} \cdot \frac{N}{df_t}^{0.35} \cdot tf_t^Q \right) \quad (9)$$

Again, Violations of constraints 1 and 3 can occur only because property P8 is absent. The normalisation is not sublinear and therefore constraint 4 is violated.

4.2 Measuring Constraint Violations

In this section, we describe how an automatic system can measure the number of constraint violations on an actual test collection. The approach, used to measure the number of constraint violations, takes a query and a stemmed document as input. The terms in the document remain in the same order in which they naturally appear. A pseudo-document is created by using the first term appearing in the document. This pseudo-document is matched against the query using a term-weighting function and the score is recorded. A further pseudo-document is created by including the next term appearing in the document. This is then matched against the query and the score is again recorded. This process continues until the complete document is scored against the query. As the score is recorded at each stage, the violations of each constraint that occur can be counted measured each time a new term is added to the pseudo-document. In this process, we only start counting constraint violations once the first query term is encountered, as until that point the score of the document will be zero.

If the score of a document does not increase when a query term is added to the pseudo-document, a violation of constraint 1 is recorded. If the score of a document does not decrease when a non-query term is added, a violation of constraint 2 is recorded. If the increase in score of the document when a query term is added is equal to, or greater than, the increase in score when the previous occurrence of that query term was added, a violation of constraint 3 is recorded. Finally, if three non-query terms appear in succession and the inverse of the score reduction is not sublinear, a violation of constraint 4 is recorded. The approach adopted to counting the violations of constraint 4 is actually a lower estimation. However, for our experiments we used the same top ranked documents for each of the term-weighting schemes and thus, results is a fair comparison.

Due to the computational complexity of such an approach, it is infeasible to do this for an entire test collection. However, typically only the top 1000 documents are returned by a retrieval system as these are deemed most likely to be relevant. Therefore, we measure the number of violations of constraints on the top 1000 documents returned from the best performing approach for the term-weighting schemes. The top 1000 documents should represent a set of documents with a high number of query terms, and therefore, is a good sample of documents on which to measure the number of constraint violations. In the next section, we will test this automatic way of measuring constraints, to see if the total number of constraint violations is inversely related to the ranking function quality (measured by MAP) on test data.

5 Results

This section presents experimental results that measures the number of actual constraint violations for a number of term-weighting schemes, in the manner outlined in the previous section. We also present the performance of these schemes on the test data.

5.1 Document Collections

We use the LATIMES, FBIS, FR documents from TREC disks 4 and 5 and topics 251 to 450 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 the OHSUMED collection and its topics. Table 1 shows some of the characteristics of the collections used in this research.

Table 1: Characteristics of Collections

Collection	LATIMES	FBIS	FR	OHSUMED
No. of Documents	131,896	130,471	55,630	293,856
Average Doc. Length	468	501	670	158
Standard Dev.	489	812	1380	60

As per the original axiomatic study [Fang & Zhai, 2005], we performed stemming, but did *not* remove stopwords. A term-weighting function that correctly models relevance should be able to correctly weight all types of terms. A complete theoretical model for retrieval should not exclude terms based on some arbitrary list. This increases the probability of violations of constraint 1 (due to property 8 not being present) as there are many terms in the documents that have a low information content. Thus, when these terms occur, the increase in penalisation may be greater than the weight added due to the term occurring.

5.2 Comparison of Schemes Using Constraint Violations

Table 2: No. of constraint violations on average per document and query on FR collection for short, medium and long queries

Topics	short				medium				long			
Schemes	C1	C2	C3	C4	C1	C2	C3	C4	C1	C2	C3	C4
PIV	2.5	0.0	1.9	396.5	79.4	0.0	52.3	306.1	223.5	0.0	148.6	259.0
*BM*25	3.6	38.3	3.5	338.3	99.0	274.6	89.9	215.5	264.6	408.9	234.5	115.3
*MBM*25	1.9	0.0	1.4	314.9	78.1	0.0	57.1	17.5	219.4	0.0	156.1	4.5
DRF	2.0	0.0	1.5	0.0	79.5	0.0	61.1	0.0	223.4	0.0	164.4	0.0
ES	0.05	0.0	0.3	0.0	6.7	0.0	8.8	0.0	57.8	0.0	45.7	0.0
LM	0.07	0.0	0.1	518.0	9.6	0.0	13.4	479.4	49.2	0.0	49.5	445.7
F2EXP	0.26	0.0	0.33	314.3	38.3	0.0	19.4	17.5	138.7	0.0	79.5	4.5

Table 2 shows the number of constraint violations averaged per document, per query, for the FR collection for the top 1000 documents of one of the best retrieval runs. For example, for a long query, the original *BM*25 scheme violates all the constraints, and violates constraint 1 an average of 264 times per document. We can see that constraint 2 is satisfied by most of the schemes. The remainder of the test collections show very similar results over a set of topics.

The first thing to notice is that none of the schemes adhere to all of the constraints unconditionally as indicated by our analysis (Section 4). Furthermore, the number of violations of constraints 1 and 3 also increases as the queries get longer. This is because there are more query terms being matched to the documents.

For longer queries, there is a greater chance of spurious terms or terms of minimal information content being introduced. This will cause more violations of constraints 1 and 3, because property P8 is absent in all of the term-weighting schemes. The *MBM*25 scheme violates each of the constraints a fewer number of times compared to the original *BM*25 scheme and should perform better in all cases. The pivoted document normalisation scheme (*PIV*) violates constraint 4 a large number of times for short, medium and long queries, which would tend to indicate that its document normalisation is poor. Our analysis has shown that the normalisation used in this scheme is poor.

Of the two schemes that adhere to most of the properties (ie. *DRF* and *ES*), the *ES* scheme violates less constraints. Both schemes unconditionally adhere to constraints 2 and 4, but the number of violations of constraints 1 and 3 for the *ES* scheme is less than a third of those of the *DFR* scheme, for all query lengths. This is an interesting result as it shows that the measure of information content (i.e. term-discrimination measure) used by the *ES* scheme and the normalisation applied therein, seem to break the constraints less often that the *DFR* scheme. Although, both schemes contains the same number of *good* term-weighting properties. The language modelling scheme (*LM*) also have very few violations of constraints 1 and 3. It seems to break constraint 4 quite frequently however. From this discussion we can predict that the best performing scheme (in general) should be the *ES* scheme and the worst scheme should be the unmodified *BM*25 scheme. The *DFR* scheme should perform better than the modified *BM*25 on most data due to its satisfaction of constraint 4.

5.3 Performance Comparison of Schemes

Table 3: MAP on test collections for short (title) queries

Schemes	LATIMES	FBIS	FR
Topics	301-450	301-450	251-450
ES	0.2256	0.2678	0.2912
F2EXP	0.2276	0.2505	0.2956
LM	0.2248	0.2596	0.2798
DRF	0.2121	0.2355	0.2796
*MBM*25	0.2106	0.2305	0.2766
PIV	0.2020	0.2163	0.2381
*BM*25	0.2080	0.2273	0.2729
ρ	-0.5	-0.5	-0.5

Tables 3, 4 and 5 show the performance of the schemes on standard test collections.The ρ measure is the Spearman correlation between the number of constraint violations for a scheme on a particular collection, and the MAP (performance) of the scheme on that collection. We can see that the best performing schemes across the collections is the scheme that breaks constraints least often on the test collections (i.e. the *ES* scheme). The *DFR* scheme slightly outperforms the modified *BM*25 scheme on short and medium queries. On longer queries the schemes perform more similarly. This correlates with the similar number of violations of constraint 4 for long queries for these schemes. We can see that although the sample size is quite small, the data indicates that there is a consistent inverse correlation between the ranking of the schemes by performance, and the ranking of schemes by the number of constraint violations. The larger number of violations of constraints on medium and long queries, for the original *BM*25 schemes, explains the very poor performance of this scheme on these types of queries (as indicated in the original work [Fang & Zhai, 2005]).

Table 4: MAP on test collections for medium length (title+desc) queries

Schemes	LATIMES	FBIS	FR	OHSUMED
Topics	301-450	301-450	251-450	1-63
ES	0.2277	0.2687	0.3150	0.3318
F2EXP	0.2445	0.2661	0.3103	0.3252
LM	0.2357	0.2833	0.3049	0.2903
DRF	0.2334	0.2447	0.2869	0.3149
*MBM*25	0.2328	0.2420	0.2825	0.3127
PIV	0.2219	0.2253	0.2475	0.3164
*BM*25	0.1695	0.1852	0.1666	0.2779
ρ	-0.39	-0.35	-0.464	-0.785

Table 5: MAP on test collections for long (title+desc+long) queries

Schemes	LATIMES	FBIS	FR
Topics	301-450	301-450	251-450
ES	0.2316	0.2395	0.3448
F2EXP	0.2634	0.2657	0.3453
LM	0.2032	0.1979	0.2144
DRF	0.2393	0.2397	0.3169
*MBM*25	0.2415	0.2395	0.3188
PIV	0.2174	0.2213	0.2832
*BM*25	0.1212	0.0445	0.0544
ρ	-0.78	-0.78	-0.928

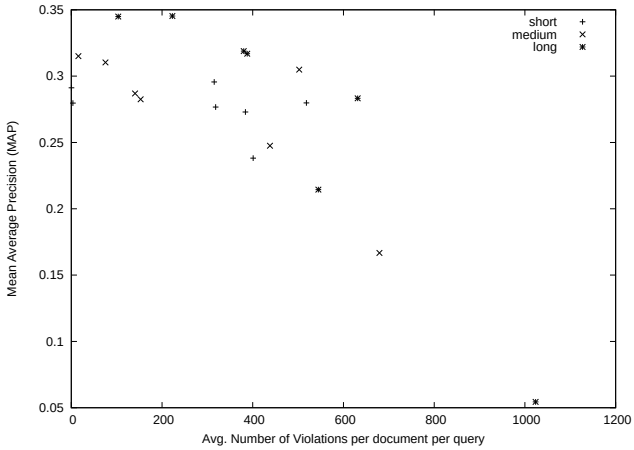

Figure 3: Number of violations vs performance on FR collection

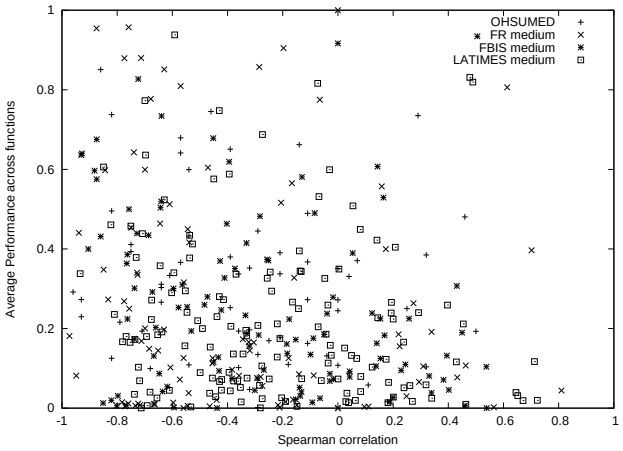

Figure 4: Spearman correlation vs average performance for all medium queries

Figure 3 shows a plot of the performance of each term-weighting function on the FR collection and the number of violations for each scheme on that collection. We can see that there is a general inverse correlation. The plot for the other collections is quite similar. Figure 4 shows the Spearman correlation of all the medium length queries vs the average performance of each query across the different functions. We can see that for most queries (80%) there is a negative correlation between violations and performance (i.e. most of the queries lie on the negative side of the plot).

5.4 Related Work: Relaxing the *bag of words* assumption

Recently, the proximity of terms in a piece of text has been shown to be useful feature in IR [Tao & Zhai, 2007, Lv & Zhai, 2009]. Several works have shown that proximity can be incorporated into retrieval functions to boost the performance at low levels of recall. These proximity based ranking functions relax the *bag of words* assumption and therefore, may violate the simplistic constraints outlined earlier.

Related work has developed two further constraints for proximity [Tao & Zhai, 2007], which are intuitively motivated, for incorporating proximity into a *bag of words* retrieval function. Some recent work [Cummins & O'Riordan, 2009, Cummins et al., 2010] has used a similar learning approach to that outlined earlier in this work (ie. genetic programming), and produced functions that appear to validate the proximity constraints previously developed. This approach of validating, seemingly intuitive, constraints by learning functions that best 'fit' the data is useful for both areas of information retrieval and machine learning.

6 Conclusions

This chapter has reviewed a number of axioms that are intuitively motivated and that can be used to constrain current term-weighting schemes in many ways. We have outlined several properties that must be present in state of the art term-weighting schemes, in order to unconditionally adhere to the constraints. Our analysis shows that none of the current state of the art term-weighting schemes can unconditionally adhere to *all* of the constraints. However, the derivation of the properties can be used to model term-weighting schemes that limit or reduce the potential for constraint violations.

Furthermore, we have outlined an approach that counts the number of actual constraint violations on a sample of the top ranked documents from a retrieval run. Complimentary to our analysis, we show that all of the term-weighting schemes presented violate some of the constraints on test data. Furthermore, many of the schemes violate some of the constraints a large number of times on the collections used. Interestingly, we show that the number of violations of all constraints that occur per document for an average query, is inversely correlated to the performance of the schemes on that collection. This approach could be used to predict the best weighting scheme to used on a per collection basis. Future work includes identifying the most important constraint to satisfy in order to best predict retrieval performance.

Acknowledgements

This work is being carried out with the support of IRCSET (the Irish Research Council for Science, Engineering and Technology) under the IRCSET-Marie Curie International Mobility Fellowship in Science, Engineering and Technology.

References

[Amati & van Rijsbergen, 2002] Amati, G. & van Rijsbergen, C. J. (2002). Term frequency normalization via pareto distributions. In *Proceedings of the 24th BCS-IRSG European Colloquium on IR Research* (pp. 183–192). London, UK: Springer-Verlag.

[Cummins & O'Riordan, 2007a] Cummins, R. & O'Riordan, C. (2007a). An axiomatic comparison of learned term-weighting schemes in information retrieval: clarifications and extensions. *Artif. Intell. Rev.*, 28(1), 51–68.

[Cummins & O'Riordan, 2007b] Cummins, R. & O'Riordan, C. (2007b). An axiomatic study of learned term-weighting schemes. In T. Joachims, H. Li, T.-Y. Liu, & C. Zhai (Eds.), *SIGIR 2007 workshop: Learning to Rank for Information Retrieval*.

[Cummins & O'Riordan, 2009] Cummins, R. & O'Riordan, C. (2009). Learning in a pairwise term-term proximity framework for information retrieval. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval* (pp. 251–258). New York, NY, USA: ACM.

[Cummins et al., 2010] Cummins, R., O'Riordan, C., & Lalmas, M. (2010). An analysis of learned proximity functions. In *9th International Conference on Adaptivity, Personalisation and Fusion of Heterogeneous Information (RIAO 2010)*.

[Fang et al., 2004] Fang, H., Tao, T., & Zhai, C. (2004). 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* (pp. 49–56).: ACM Press.

[Fang & Zhai, 2005] Fang, H. & Zhai, C. (2005). 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* (pp. 480–487).: ACM Press.

[Fuhr & Buckley, 1991] Fuhr, N. & Buckley, C. (1991). A probabilistic learning approach for document indexing. *ACM TRANSACTIONS ON INFORMATION SYSTEMS*, 9, 223–248.

[Harman, 1993] Harman, D. (1993). Overview of the first trec conference. In *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 36–47). New York, NY, USA: ACM.

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

[Lv & Zhai, 2009] Lv, Y. & Zhai, C. (2009). Positional language models for information retrieval. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval* (pp. 299–306). New York, NY, USA: ACM.

[Ponte & Croft, 1998] Ponte, J. M. & Croft, W. B. (1998). A language modeling approach to information retrieval. In *Research and Development in Information Retrieval* (pp. 275–281).

[Radlinski & Joachims, 2005] Radlinski, F. & Joachims, T. (2005). Query chains: learning to rank from implicit feedback. In *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining* (pp. 239–248). New York, NY, USA: ACM.

[Robertson et al., 1995] Robertson, S. E., Walker, S., Hancock-Beaulieu, M., Gull, A., & Lau, M. (1995). Okapi at TREC-3. In *In D. K. Harman, editor, The Third Text REtrieval Conference (TREC-3) NIST*.

[Salton & Buckley, 1988] Salton, G. & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5), 513–523.

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

[Shi et al., 2005] Shi, S., Wen, J.-R., Yu, Q., Song, R., & Ma, W.-Y. (2005). 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* (pp. 488–495). New York, NY, USA: ACM.

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

[Singhal et al., 1996] Singhal, A., Buckley, C., & Mitra, M. (1996). Pivoted document length normalization. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 21–29).: ACM Press.

[Tao & Zhai, 2007] Tao, T. & Zhai, C. (2007). An exploration of proximity measures in information retrieval. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 295–302). New York, NY, USA: ACM.

[Zobel & Moffat, 1998] Zobel, J. & Moffat, A. (1998). Exploring the similarity space. *SIGIR Forum*, 32(1), 18–34.