A Formal Study of Information Retrieval Heuristics

Hui Fang
Department of Computer Science
University of Illinois at Urbana Champaign
Urbana,IL 61801
hfang@cs.uiuc.edu

Tao Tao
Department of Computer Science
University of Illinois at Urbana Champaign
Urbana,IL 61801
taotao@cs.uiuc.edu

ChengXiang Zhai
Department of Computer Science
University of Illinois at Urbana Champaign
Urbana,IL 61801
czhai@cs.uiuc.edu

ABSTRACT

Empirical studies of information retrieval methods show that good retrieval performance is closely related to the use of various retrieval heuristics, such as TF-IDF weighting. One basic research question is thus what exactly are these "necessary" heuristics that seem to cause good retrieval performance. In this paper, we present a *formal* study of retrieval heuristics. We formally define a set of basic desirable constraints that any reasonable retrieval function should satisfy, and check these constraints on a variety of representative retrieval functions. We find that none of these retrieval functions satisfies all the constraints unconditionally. Empirical results show that when a constraint is not satisfied, it often indicates non-optimality of the method, and when a constraint is satisfied only for a certain range of parameter values, its performance tends to be poor when the parameter is out of the range. In general, we find that the empirical performance of a retrieval formula is tightly related to how well it satisfies these constraints. Thus the proposed constraints provide a good explanation of many empirical observations and make it possible to evaluate any existing or new retrieval formula *analytically*.

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

General Terms: Experimentation

Keywords: Retrieval heuristics, constraints, formal models, TF-IDF weighting

1. INTRODUCTION

The study of retrieval models is central to information retrieval. Many different retrieval models have been proposed and tested, including vector space models [13, 12, 10], probabilistic models[7, 16, 15, 3, 6, 5], and logic-based models[17, 19, 2]. Despite this progress in the development of formal retrieval models, good empirical performance rarely comes directly from a theoretically well-motivated model; rather, heuristic modification of a model is often necessary in order to achieve optimal retrieval performance. Indeed, many empirical studies show that good retrieval performance is closely related to the use of various retrieval heuristics, especially TF-IDF weighting and document length normalization. Many empirically effective retrieval formulas tend to boil down to an explicit or implicit implementation of these retrieval heuristics, even though they may be motivated quite differently [18]. Even the recently developed language modeling approach has been shown to be connected with these heuristics [20]. It thus appears that these heuristics are somehow *necessary* for achieving good retrieval performance. However, it is unclear at all what exactly are these "necessary heuristics" mathematically. A basic research question is then how we can *formally* define and characterize these necessary retrieval heuristics.

In this paper, we present a formal study of retrieval heuristics. We formally define a set of basic desirable constraints that any reasonable retrieval formula should satisfy, and check these constraints on a variety of retrieval formulas, which respectively represent the vector space model (pivoted normalization), the classic probabilistic retrieval model (Okapi), and the recently proposed language modeling approach (Dirichlet prior smoothing). We find that none of these retrieval formulas satisfies all the constraints unconditionally, though some formulas violate more constraints or violate some constraints more "seriously" than others. Empirical results show that when a constraint is not satisfied, it often indicates non-optimality of the method, and when a constraint is satisfied only for a certain range of parameter values, its performance tends to be poor when the parameter is out of the range. In general, we find that the empirical performance of a retrieval formula is tightly related to how well it satisfies these constraints. Thus the proposed constraints provide a good explanation of many empirical observations about retrieval methods. Moreover, these constraints make it possible to evaluate any existing or new retrieval formula *analytically* and suggest how we may further improve a retrieval formula.

The rest of the paper is organized as follows. We first present six formal constraints in Section 2. In Section 3, we apply these constraints to a variety of representative retrieval formulas and propose some hypotheses about the performance behavior of these formulas based on the analysis results. We test these hypotheses with systematic experiments in Section 4. Finally, we discuss our findings and future research directions in Section 5.

2. FORMAL DEFINITIONS OF HEURISTIC RETRIEVAL CONSTRAINTS

In this section, we formally define six intuitive and desirable constraints that any reasonable retrieval formula should satisfy. They capture the commonly used retrieval heuristics, such as TF-IDF weighting, in a formal way, making it possible to apply them to any retrieval formula analytically.

These constraints are motivated by the following observations on some common characteristics of typical retrieval formulas. First, most retrieval methods assume a "bag of words" (more precisely, "bag of terms") representation of both documents and queries. Second, a highly effective retrieval function typically involves a TF part, an IDF part, and a document length normalization part [11, 21]. The TF part intends to give a higher score to a document that has more occurrences of a query term, while the IDF part is to penalize words that are popular in the whole collection. The document length normalization is to avoid favoring long documents; long documents generally have more chances to match a query term simply because they contain more words. Finally, different retrieval formulas do differ in their way of combining all these factors, even though their empirical performances may be similar.

These observations suggest that there are some "basic requirements" that all reasonable retrieval formulas should follow. For example, if a retrieval formula does not penalize common words, then it somehow violates the "IDF requirement", thus can be regarded as "unreasonable." However, some of these requirements may compromise each other. For example, while the TF heuristic intends to assign a higher score to a document that has more occurrences of a query term, the document length normalization component may cause a long document with a higher TF to receive a lower score than a short document with a lower TF. Similarly, if two documents match precisely one single, but different query term, the IDF heuristic may allow a document with a lower TF to "beat" the one with a much higher TF. A critical question is thus how we can regulate such interactions so that they will all be "playing a fair game"? Clearly, in order to answer this question, we must first define what is a "fair game", i.e., we must define what exactly is a *reasonable* retrieval function.

Our idea is to characterize a reasonable retrieval formula by listing the desirable constraints that any reasonable retrieval formula must satisfy. We now formally define six such desirable constraints. Note that these constraints are *necessary*, but not necessarily sufficient, and should not be regarded as the *only* constraints that we want a retrieval function to satisfy; indeed, it is not hard to come up with additional constraints that may also make sense. However, we focus on these six basic constraints in this paper because they capture the major well-known IR heuristics, particularly TF-IDF weighting and length normalization.

Let us first introduce some notations. We use d or d_i to denote a document, q to denote a query, w or w_i to represent a query term, and w' to represent a non-query term. $c(w,d)$ is the count of word w in document d. $|d|$ denotes the length of document d. f denotes a retrieval function, and $f(d,q)$ gives the score of document d with respect to query q. $idf(w)$ denotes any IDF-like discrimination value of a term w.

2.1 Term Frequency Constraints (TFCs)

TFC1: Let $q = \{w\}$ be a query with only one term w. Assume $|d_1| = |d_2|$. If $c(w,d_1) > c(w,d_2)$, then $f(d_1,q) > f(d_2,q)$.

TFC2: Let $q = \{w\}$ be a query with only one term w. Assume $|d_1| = |d_2| = |d_3|$ and $c(w,d_1) > 0$. If $c(w,d_2) - c(w,d_1) = 1$ and $c(w,d_3) - c(w,d_2) = 1$, then $f(d_2,q) - f(d_1,q) > f(d_3,q) - f(d_2,q)$.

Both constraints are to capture the desired contribution of the TF of a term to scoring. The first constraint captures the basic TF heuristic, which gives a higher score to a document with more occurrences of a query term when the only difference between two documents is the occurrences of the query term. In other words, the score of retrieval formula will increase with the increase in TF (i.e., the first partial derivative of the formula w.r.t. the TF variable should be positive). The second constraint ensures that the increase in the score due to an increase in TF is smaller for larger TFs (i.e., the second partial derivative w.r.t. the TF variable should be negative). Here, the intuition is that the change in the score caused by increasing TF from 1 to 2 should be larger than that caused by increasing TF from 100 to 101.

Interestingly, it can be shown that the TFC2 constraint also implies another desirable property – if two documents have the same total occurrences of all query terms, a higher score will be given to the document covering more distinct query terms. This property can be formalized as follows. Let q be a query and $w_1, w_2 \in q$ be two query terms. Assume $|d_1| = |d_2|$ and $idf(w_1) = idf(w_2)$. If $c(w_1,d_1) = c(w_1,d_2) + c(w_2,d_2)$ and $c(w_2,d_1) = 0$, $c(w_1,d_2) \neq 0$,$c(w_2,d_2) \neq 0$, then $f(d_1,q) < f(d_2,q)$.

2.2 Term Discrimination Constraint (TDC)

TDC: Let q be a query and $w_1, w_2 \in q$ be two query terms. Assume $|d_1| = |d_2|$, $c(w_1,d_1)+c(w_2,d_1) = c(w_1,d_2)+c(w_2,d_2)$. If $idf(w_1) \geq idf(w_2)$ and $c(w_1,d_1) \geq c(w_1,d_2)$, then $f(d_1,q) \geq f(d_2,q)$.

This constraint regulates the interaction between TF and IDF, and accurately describes the effect of using IDF in scoring. It ensures that, given a fixed number of occurrences of query terms, we should favor a document that has more occurrences of *discriminative* terms (i.e., high IDF terms). Clearly, simply weighting each term with an IDF factor does *not* ensure that this constraint be satisfied. When applying this constraint, IDF can be any reasonable measure of term discrimination value (usually based on term popularity in a collection).

2.3 Length Normalization Constraints (LNCs)

LNC1: Let q be a query and d_1, d_2 be two documents. If for some word $w' \notin q$, $c(w',d_2) = c(w',d_1) + 1$ but for any query term w, $c(w,d_2) = c(w,d_1)$, then $f(d_1,q) \geq f(d_2,q)$.

LNC2: Let q be a query. $\forall k > 1$, if d_1 and d_2 are two documents such that $|d_1| = k \cdot |d_2|$ and for all terms w, $c(w,d_1) = k \cdot c(w,d_2)$, then $f(d_1,q) \geq f(d_2,q)$.

The first constraint says that the score of a document should decrease if we add one extra occurrence of a "non-relevant word" (i.e., a word not in the query), thus intends to penalize long documents. The second constraint intends to avoid over-penalizing long documents, as it says that if we concatenate a document with itself k times to form a new document, then the score of the new document should not be lower than the original document. Here, we make the assumption that the redundance issue is not considered.

Table 1: Summary of intuitions for each formalized constraint

Constraints	Intuitions
TFC1	to favor a document with more occurrence of a query term
TFC2	to favor document matching more distinct query terms
TFC2	to make sure that the change in the score caused by increasing TF from 1 to 2 is larger than that caused by increasing TF from 100 to 101.
TDC	to regulate the impact of TF and IDF
LNC1	to penalize a long document(assuming equal TF)
LNC2, TF-LNC	to avoid over-penalizing a long document
TF-LNC	to regulate the interaction of TF and document length

2.4 TF-LENGTH Constraint (TF-LNC)

TF-LNC: Let $q = \{w\}$ be a query with only one term w. If d_1 and d_2 are two documents such that $c(w, d_1) > c(w, d_2)$ and $|d_1| = |d_2| + c(w, d_1) - c(w, d_2)$, then $f(d_1, q) > f(d_2, q)$.

This constraint regulates the interaction between TF and document length. The intuition is that if d_1 is generated by adding more occurrences of the query term to d_2, the score of d_1 should be higher than d_2.

Based on TF-LNC and LNC1, it is not hard to derive the following constraint:

Let $q = \{w\}$ be a query with only one term w. If d_2 and d_3 are two documents such that $c(w, d_3) > c(w, d_2)$ and $|d_3| < |d_2| + c(w, d_3) - c(w, d_2)$, then $f(d_3, q) > f(d_2, q)$.

The above constraint can be derived in the following way. Assume we have a document d_1 such that $|d_1| = |d_2| + c(w, d_1) - c(w, d_2)$ and $c(w, d_3) = c(w, d_1)$. It is obvious that the only difference between d_1 and d_3 is that d_1 has more occurrences of the non-query terms. According to LNC1, we know that $f(d_3, q) \geq f(d_1, q)$. Since $f(d_1, q) > f(d_2, q)$ follows from TF-LNC, it is clear that $f(d_3, q) > f(d_2, q)$.

This constraint ensures that document d_1, which has a higher TF for the query term, should have a higher score than d_2, which has a lower TF, as long as d_1 is not too much longer than d_2.

The first three constraints (i.e. TFCs and TDC) are intended to capture the desired scoring preferences when two documents have equal lengths. The other three constraints are applicable when we have variable document lengths. In Table 1, we summarize the intuitions behind each formalized constraint.

These constraints are basic and non-redundant in the sense that none of them can be derived from the others. Formally, suppose C_i represents the set of all the retrieval functions satisfying the i-th Constraint, then we can show that $\forall i, j$, $\exists e \in C_i$, such that $e \in C_i - C_j$.

We must emphasize that the constraints proposed in this section are necessary constraints for a "reasonable" retrieval formula, but not necessarily sufficient, and should not be regarded as the *only* constraints that a "reasonable" retrieval formula needs to satisfy. Thus when a constraint is violated, we know the retrieval function may not perform well empirically since it is not entirely consistent with our intuitive preferences, but satisfying all the constraints does not necessarily guarantee good performance.

3. ANALYSIS OF THREE REPRESENTATIVE RETRIEVAL FORMULAS

In this section, we apply the six constraints defined in the previous section to three specific retrieval formulas, which respectively represent the vector space model, the classical probabilistic retrieval model, and the language modeling approach. Our goal is to see how well each retrieval formula satisfies the proposed constraints. As will be shown, it turns out that none of these retrieval formulas satisfies all the constraints *unconditionally*, though some models violate more constraints or violate some constraints more "seriously" than others. The analysis thus suggests some hypotheses regarding the empirical behavior of these retrieval formulas, which will be tested in the next section.

The following notations will be used in this section:
$c(w, d)$ is the count of word w in the document d.
$c(w, q)$ is the count of word w in the query q.
N is the total number of documents in the collection.
$df(w)$ is the number of documents that contain the term w.
$|d|$ is the length of document d.
$avdl$ is the average document length.

3.1 Pivoted Normalization Method

The pivoted normalization retrieval formula [14] is one of the best performing vector space retrieval formulas. In the vector space model, text is represented by a vector of terms. Documents are ranked by the similarity between the query vector and the document vector. According to [14], the pivoted normalization retrieval formula is

$$\sum_{w \in q \cap d} \frac{1 + ln(1 + ln(c(w, d)))}{(1 - s) + s\frac{|d|}{avdl}} \cdot c(w, q) \cdot ln\frac{N + 1}{df(w)}$$

The results of analyzing the pivoted normalization formula

Table 2: Constraint analysis results (Pivoted)

TFCs	TDC	LNC1	LNC2	TF-LNC
Yes	Cond.	Yes	Cond.	Cond.

are summarized in Table 2. ("Cond" means "Conditional".) TFCs and LNC1 are easily seen to be satisfied. We now examine some of the non-trivial constraints.

First, let us check the TF-LNC constraint. Consider a common case when $|d_1| = avdl$. It can be shown that the TF-LNC constraint is equivalent to the following constraint on the parameter s:

$$s \leq \frac{h(c(w,d_1)) - h(c(w,d_2))}{(c(w,d_1) - c(w,d_2)) \times (1 + h(c(w,d_1)))} \times avdl$$

where $h(x) = ln(1 + ln(x))$.

This means that TF-LNC is satisfied only if s is below a certain upper bound. The TF-LNC constraint thus provides an upper bound for s, which is tighter for a larger $c(w,d_1)$.

Next, we consider the TDC constraint. It can be shown that TDC is equivalent to $c(w_2,d_1) \geq c(w_1,d_2)$, which is only conditionally satisfied.

Finally, we show that the LNC2 leads to an upper bound for parameter s. The LNC2 constraint is equivalent to

$$\frac{1 + ln(1 + ln(k \times c(w,d_2)))}{1 - s + s\frac{k \times |d_2|}{avdl}} \cdot c(w,q) \cdot ln\frac{N+1}{df(w)} \geq \frac{1 + ln(1 + ln(c(w,d_2)))}{1 - s + s\frac{|d_2|}{avdl}} \cdot c(w,q) \cdot ln\frac{N+1}{df(w)}$$

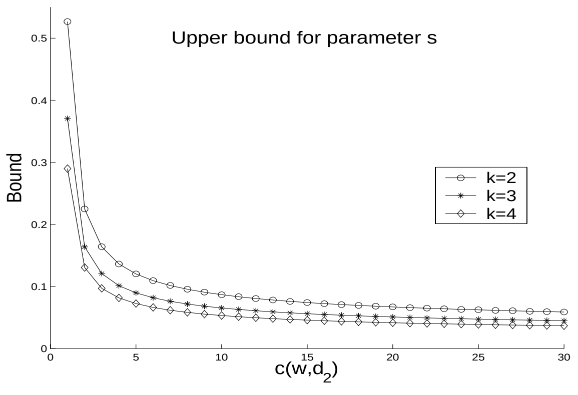

Figure 1: Upper bound of parameter s.

Therefore, the upper bound of s can be derived as:

$$s \leq \frac{tf_1 - tf_2}{(k\frac{|d_2|}{avdl} - 1)tf_2 - (\frac{|d_2|}{avdl} - 1)tf_1}$$

where $tf_1 = 1 + ln(1 + ln(k \times c(w,d_2))), tf_2 = 1 + ln(1 + ln(c(w,d_2)))$. In order to get a sense of what the bound is exactly, consider a common case when $|d_2| = avdl$. We have

$$s \leq \frac{1}{k-1} \times (\frac{tf_1}{tf_2} - 1).$$

As shown in the Figure 1, the bound becomes tighter when k increases or when the term frequency is larger. This bound shows that in order to avoid over-penalizing a long document, a reasonable value for s should be generally small – it should be below 0.4 even in the case of a small k, and we know that for a larger k the bound would be even tighter. This analysis thus suggests that the performance can be bad for a large s, which is confirmed by our experiments.

3.2 Okapi Method

The Okapi formula is another highly effective retrieval formula that represents the classical probabilistic retrieval model [8]. The formula as presented in [14] is [1]

$$\sum_{w \in q \cap d} \left(ln\frac{N - df(w) + 0.5}{df(w) + 0.5} \times \frac{(k_1 + 1) \times c(w,d)}{k_1((1-b) + b\frac{|d|}{avdl}) + c(w,d)} \times \frac{(k_3 + 1) \times c(w,q)}{k_3 + c(w,q)} \right)$$

where k_1 (between 1.0-2.0), b (usually 0.75), and k_3 (between 0-1000) are constants.

The major difference between Okapi and other retrieval formulas is the possibly negative value of the IDF part in the formula, which has been discussed in [9]. It is trivial to show that if $df(w) > N/2$, the IDF value would be negative.

When the IDF part is positive (which is mostly true for keyword queries), TFCs and LNCs are easily seen to be satisfied. By considering a common case when $|d_2| = avdl$, the TF-LNC constraint is shown to be equivalent to $b \leq \frac{avdl}{c(w,d_2)}$. As we know, the value of b is always smaller than 1. Therefore, TF-LNC can be satisfied unconditionally. Moreover, we can show that TDC is equivalent to $c(w_1,d_2) \leq c(w_2,d_1)$, which is the same as the result for the pivoted normalization method.

Although Okapi satisfies some constraints conditionally, unlike in the pivoted normalization method, the conditions do not provide any bound for the parameter b. Therefore, the performance of Okapi can be expected to be less sensitive to the length normalization parameter than the pivoted normalization method, which is confirmed by our experiments.

When the IDF part is negative, the Okapi formula would clearly violate the TFCs, LNCs and TF-LNC, since matching an additional occurrence of a query term could mean decreasing the score. It would satisfy TDC when $c(w_1,d_2) > c(w_2,d_1)$. Since a negative IDF only happens when a query term has a very high document frequency (e.g., when the query is verbose), our analysis suggests that the performance of Okapi may be relatively worse for verbose queries than for keyword queries.

A simple way to solve the problem of negative IDF is to replace the original IDF in Okapi with the regular IDF in the pivoted normalization formula. This modified Okapi satisfies all the constraints but TDC. We thus hypothesize that the performance of the modified Okapi would perform better than the original Okapi for verbose queries. As will be shown later, this is indeed true according to our experiment results.

Table 3: Constraint analysis results (Okapi)

Formula	TFCs	TDC	LNC1	LNC2	TF-LNC
Original	Cond	Cond	Cond	Cond	Cond
Modified	Yes	Cond	Yes	Yes	Yes

The results of analyzing the Okapi formula are summarized in Table 3. We distinguish two forms of the formula – the original formula and the one with a modified IDF part. The modification significantly affects the constraint analysis results as discussed above.

[1]There is a typo in the formula in [14], which is corrected here.

3.3 Dirichlet Prior Method

The Dirichlet prior retrieval method is one of the best performing language modeling approaches [20]. This method uses the Dirichlet prior smoothing method to smooth a document language model and then ranks documents according to the likelihood of the query according to the estimated language model of each document. With a notation consistent with those in the pivoted normalization and Okapi formulas, the Dirichlet prior retrieval function is

$$\sum_{w \in q \cap d} c(w,q) \cdot ln(1 + \frac{c(w,d)}{\mu \cdot p(w|C)}) + |q| \cdot ln\frac{\mu}{|d| + \mu}$$

where, $|q|$ is the query length, and $p(w|C)$ is the probability of a term w given by the collection language model. $p(w|C)$ indicates how popular the term w is in the whole collection, thus is quite similar to the document frequency $df(w)$. The results of analyzing the Dirichlet prior formula are summarized in Table 4. TFCs, LNC1 and TF-LNC are easily seen to be satisfied. So we only examine some of the non-trivial constraints.

Table 4: Constraint analysis results (Dirichlet)

TFCs	TDC	LNC1	LNC2	TF-LNC
Yes	Cond	Yes	Cond	Yes

The LNC2 constraint can be shown to be equivalent to $c(w,d_2) \geq |d_2| \cdot p(w|C)$, which is usually satisfied for content-carrying words. If all the query terms are discriminative words, long documents will not be over-penalized. Thus, compared to pivoted normalization, Dirichlet prior appears to have a more robust length normalization mechanism, even though none of them satisfies the LNC2 constraint unconditionally.

Another interesting observation is that TDC constraint may lead to some lower bound for parameter μ, as derived below. Assume $p(w_1|C) \leq p(w_2|C)$ (roughly equivalent to $idf(w_1) > idf(w_2)$). TDC implies

$$ln(1 + \frac{c(w_1,d_1)}{\mu p(w_1|C)}) + ln(1 + \frac{c(w_2,d_1)}{\mu p(w_2|C)}) + 2ln\frac{\mu}{\mu + |d_1|} \geq$$
$$ln(1 + \frac{c(w_1,d_2)}{\mu p(w_1|C)}) + ln(1 + \frac{c(w_2,d_2)}{\mu p(w_2|C)}) + 2ln\frac{\mu}{\mu + |d_2|}$$

After some simplification, we can obtain a lower bound for μ:

$$\mu \geq \frac{c(w_1,d_1) - c(w_2,d_2)}{p(w_2|C) - p(w_1|C)}$$

In order to have a sense of the exact value of this bound, let us consider a common case of w_2 such that $p(w_2|C) = \frac{1}{avdl}$ (i.e. w_2 is expected to occur once in a document). We have

$$\begin{aligned} \mu &> \frac{c(w_1,d_1) - c(w_2,d_2)}{p(w_2|C)} \\ &= avdl \times (c(w_1,d_1) - c(w_2,d_2)) \end{aligned}$$

It means that for discriminative words with a high term frequency in a document, μ needs to be sufficiently large (at least as large as the average document length) in order to balance TF and IDF appropriately. In general, the analysis shows that μ has a lower bound, and a very small μ might cause poor retrieval performance. This is also confirmed by our experiments.

Table 5: Comparison between different retrieval formulas

Formula	TFCs	TDC	LNC1	LNC2	TF-LNC
Pivoted	Yes	C_1	Yes	C_2^*	C_3^*
Dirichlet	Yes	C_4^*	Yes	C_5	Yes
Okapi (original)	C_6	$C_1 \cap C_6$ $\neg C_1 \cap \neg C_6$	C_6	C_6	C_6
Okapi (modified)	Yes	C_1	Yes	Yes	Yes

3.4 Summary

We have applied our six constraints to three representative retrieval formulas. The results are summarized in Table 3.4, where a "Yes" means the corresponding model satisfies the particular constraint and a "C_x" means corresponding model satisfies the particular constraint under some particular conditions (irrelevant to parameter setting), and a "C_x^*" means the model satisfies the constraint only when the parameter is in some range. The specific conditions are

$$\begin{aligned}
C_1 &\Leftrightarrow c(w_1,d_2) \leq c(w_2,d_1) \\
C_2^* &\Leftrightarrow s \leq \frac{tf_1 - tf_2}{(k\frac{|d_2|}{avdl} - 1)tf_2 - (\frac{|d_2|}{avdl} - 1)tf_1} \\
C_3^* &\Leftrightarrow s \leq \frac{(h(c(w,d_1)) - h(c(w,d_2))) \times avdl}{(c(w,d_1) - c(w,d_2)) \times (1 + h(c(w,d_1)))} \\
C_4^* &\Leftrightarrow \mu \geq \frac{c(w_1,d_1) - c(w_2,d_2)}{p(w_2|C) - p(w_1|C)} \\
&\qquad > avdl \times (c(w_1,d_1) - c(w_2,d_2)) \\
C_5 &\Leftrightarrow c(w,d_2) \geq |d_2| \cdot p(w|C) \\
C_6 &\Leftrightarrow idf(w) \geq 0 \Leftrightarrow df(w) \leq N/2
\end{aligned}$$

Based on the results, we can make several interesting observations:

First, it is surprising that all the methods, including a highly effective TF-IDF model, fail to satisfy the TDC (essentially the IDF heuristics) unconditionally.

Second, it is also surprising that the original IDF part of Okapi formula causes the formula to violate almost all constraints, thus we may predict that the Okapi formula may have a worse performance for verbose queries.

Finally, C_2, C_3 and C_4 provide an approximate bound for the parameters in pivoted normalization method and Dirichlet prior method. In contrast, by checking the constraints, we have not found any particular bound for the parameter in Okapi. Therefore, we predict that the performance of Okapi is less sensitive to parameter setting than that of the other two methods.

4. EXPERIMENTS

In the previous section, we have examined three representative retrieval formulas analytically. Based on the analysis, we propose some hypotheses about the performance for each retrieval formula. In this section, we test these hypotheses through carefully designed experiments. Our experiment results show that the proposed constraints can both explain the performance difference in various retrieval models and provide an approximate bound for the parameters in a retrieval formula.

Table 6: Document set characteristic

	AP	DOE	FR	ADF	Web	Trec7	Trec8
#qry	142	35	42	144	50	50	50
#rel/q	103	57	33	126	46	93	95
size	491MB	184MB	469MB	1GB	2GB	2GB	2GB
#doc(k)	165K	226K	204K	437K	247K	528K	528K
#voc(k)	361K	163K	204K	700K	1968K	908K	908K
mean(dl)	454	117	1338	372	975	477	477
dev(dl)	239	58	5226	1739	2536	789	789
mean(rdl)	546	136	12466	1515	6596	1127	1325

4.1 Experiment Design

As is well-known, retrieval performance can vary significantly from one test collection to another. We thus construct several quite different and representative test collections using the existing TREC test collections.

To cover different types of queries, we follow [20] , and vary two factors: query length and verbosity, which gives us four different combinations : short-keyword (SK, keyword title), short-verbose (SV, one sentence description), long-keyword (LK, keyword list), and long-verbose (LV, multiple sentences). The number of queries is usually larger than 50. To cover different types of documents, we construct our document collections by varying several factors, including (1) the type of documents; (2) document length; (3) collection size(varies from 165K documents to 528K documents); and (4) collection homogeneity. Our choice of document collection has been decided to be news articles (AP), technical reports (DOE), government documents (FR), a combination of AP, DOE, and FR (ADF), the Web data used in TREC8(Web), the ad hoc data used in TREC7(Trec7) and the ad hoc data used in TREC8(Trec8). Table 6 shows some document set characteristics, including the number of queries used on the document set, the average number of relevant documents per query, the collection size, the number of documents, the vocabulary size, the mean document length, the standard deviation of document length, and the mean length of relevant documents.

The preprocessing of documents and queries is minimum, involving only stemming with the Porter's stemmer. No stop words have been removed, as it would introduce at least one extra parameter (e.g., the number of stop words) into our experiments. On each test collection, for every retrieval method, we vary the retrieval parameter to cover a reasonably wide range of values. This allows us to see a complete picture of how sensitive each method is to its parameter.

4.2 Parameter Sensitivity

Based on the analysis in Section 3, we formulate the following hypotheses: (1) The pivoted normalization method is sensitive to the value of parameter s. The analysis of LNC2 suggests that the reasonable value for s should be generally smaller than 0.4 and the performance can be bad for a large s. (2) Okapi is more stable with the change of parameter b. (3) The Dirichlet prior method is sensitive to the value of parameter μ. The analysis of TDC shows that μ has some lower bound, and a very small μ might cause poor retrieval performance.

We now discuss the experiment results. First, let us consider the experiment result for pivoted normalization. The optimal value of s is shown in Table 7. As shown in the table, the optimal value of s to maximize average precision has been found to be indeed quite small in all cases. Moreover, we also see that when s is large, which causes the method not to satisfy the LNC2 constraint, the performance is significantly worse. In Figure 2, we show how the average precision is influenced by the parameter value in the pivoted normalization method on the AP document set and long-keyword queries; the curves are similar for all other data sets.

Table 7: Optimal s (for average precision) in the pivoted normalization method

	AP	DOE	FR	ADF	Web	Trec7	Trec8
lk	0.2	0.2	0.05	0.2	—	—	—
sk	0.01	0.2	0.01	0.05	0.01	0.05	0.05
lv	0.3	0.3	0.1	0.2	0.2	0.2	0.2
sv	0.2	0.3	0.1	0.2	0.1	0.1	0.2

Next, we experiment with the Okapi method. Assume $k_1 = 1.2, k_3 = 1000$ and b changes from 0.1 to 1.0. The performance of Okapi is indeed more stable compared with the pivoted normalization (shown in Figure 2). By checking the constraints, we have not found any particular bound for the parameter, which may explain why the performance is much less sensitive to the parameter value than in the pivoted normalization method where a bound for parameter s is implied by the LNC2 constraint.

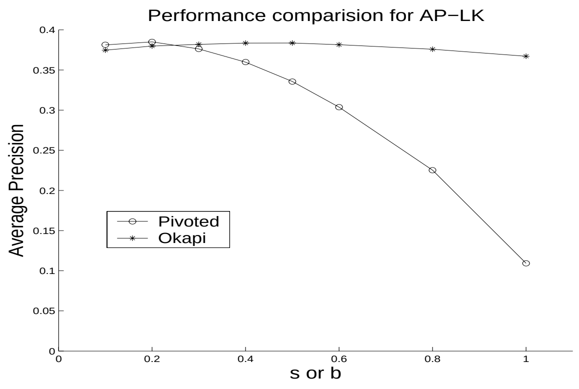

Figure 2: Performance Comparison between Okapi and Pivoted for AP-LK.

Finally, the optimal values of μ in Dirichlet are shown in Table 8. We see that these optimal values are all greater than the average document length, also shown in the same table. We further plot how the average precision is influenced by the parameter value in Figure 3. Clearly, when μ is larger than a specific value, the performance is relatively stable. However, when μ is small, the performance is noticeably worse.

Table 8: Optimal μ (for average precision) in the Dirichlet prior method

	AP	DOE	FR	ADF	Web	Trec7	Trec8
lk	2000	2000	20000	1000	—	—	—
sk	2000	2000	5000	2000	4000	2000	800
lv	3000	1000	15000	3000	8000	3000	2000
sv	8000	4000	20000	3000	10000	8000	5000
avdl	454	117	1338	372	975	477	477

Therefore, in general it seems that the constraints could provide an empirical bound for the parameter in the retrieval

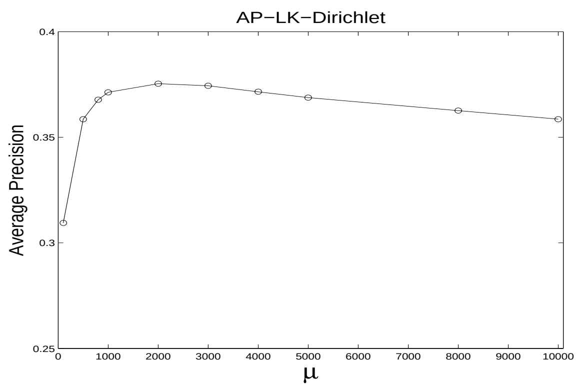

Figure 3: Performance of Dirichlet for AP-LK.

formula and the performance would tend to be poor when the parameter is out of the bound.

4.3 Performance Comparison

In this subsection, we compare the performance of these three retrieval formulas through systematic experiments. Our goal is to see whether the experiment results are consistent with the analytical results based on formalized heuristics. We form the following hypotheses based on the constraint analysis:(1) For any query type, the performance of Dirichlet prior method is comparable to pivoted normalization method when the retrieval parameters are set to an optimal value. (2) For keyword queries, the performance of Okapi is comparable to the other two retrieval formulas. (3) For verbose queries, the performance of Okapi may be worse than others, due to the possible negative IDF part in the formula. As mentioned in Section 3, when IDF is negative, Okapi violates almost all the constraints. However, if we modify the Okapi formula by replacing the original IDF part with IDF part of the pivoted normalization method, then the formula would satisfy almost all the constraints for any query type, therefore we hypothesize that the modified Okapi formula performs better than the original one for verbose queries.

In order to test these hypotheses, we run experiments over seven collections and four query sets by using the pivoted normalization method, the Dirichlet prior method, Okapi and the modified Okapi formula (which replaces the IDF part in Okapi with the IDF part in the pivoted normalization formula). We use average precision as the evaluation measure. The optimal performance for each formula is summarized in Table 9. The results show that for verbose queries, the performance of the Mod-Okapi is significantly better than that of Okapi; the *p*-values of the Wilcoxin signed rank test are all below 0.013.

We see that, indeed, for keyword queries, the performances of three retrieval formulas are comparable. However, for verbose queries, in most cases the performance of Okapi is worse than others, which may be caused by the negative IDF scores for common words. This hypothesis is verified by the performance of the modified Okapi. After replacing the IDF part in Okapi with the IDF part of the pivoted normalization formula, the performance is improved significantly for the verbose queries. See Figure 2 and Figure 4 for plots of these comparisons.

From Figure 4, we may conclude that satisfying more constraints appears to be correlated with a better performance. Therefore, the proposed constraints provide a plausible explanation for the performance difference in various retrieval models, and suggest how we may improve a retrieval formula further.

Table 9: Comparison of optimal performance for four formulas.

		AP	DOE	FR	ADF	Web	Trec7	Trec8
lk	Piv	0.39	0.28	0.33	0.27	—	—	—
lk	Dir	0.38	0.28	0.32	0.25	—	—	—
lk	Okapi	0.38	0.27	0.28	0.33	—	—	—
lk	Mod-Okapi	0.39	0.28	0.28	0.33	—	—	—
sk	Piv	0.23	0.18	0.19	0.22	0.29	0.18	0.24
sk	Dir	0.22	0.18	0.18	0.21	0.30	0.19	0.26
sk	Okapi	0.23	0.19	0.23	0.19	0.31	0.19	0.25
sk	Mod-Okapi	0.23	0.19	0.23	0.19	0.31	0.19	0.25
lv	Piv	0.29	0.21	0.23	0.21	0.22	0.20	0.23
lv	Dir	0.29	0.23	0.24	0.24	0.28	0.22	0.26
lv	Okapi	0.03	0.07	0.09	0.06	0.23	0.08	0.11
lv	Mod-Okapi	0.30	0.24	0.25	0.23	0.28	0.26	0.25
sv	Piv	0.19	0.10	0.14	0.14	0.21	0.15	0.20
sv	Dir	0.20	0.13	0.16	0.16	0.27	0.18	0.23
sv	Okapi	0.08	0.08	0.08	0.09	0.21	0.09	0.10
sv	Mod-Okapi	0.19	0.12	0.16	0.14	0.25	0.16	0.22

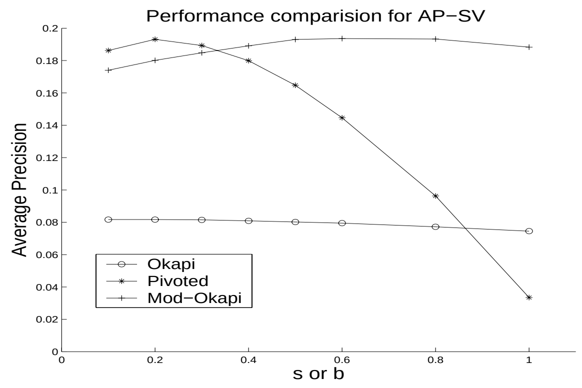

Figure 4: Performance Comparison between modified Okapi, Okapi and Pivoted for AP-SV.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we study the problem of formalizing the necessary heuristics for good retrieval performance. Motivated by some observations on common characteristics of typical retrieval formulas, we formally define six basic constraints that any reasonable retrieval function should satisfy. These constraints correspond to some desirable intuitive heuristics, such as term frequency weighting, term discrimination weighting and document length normalization. We check these six constraints on three representative retrieval formulas analytically and derive specific conditions when a constraint is conditionally satisfied. The constraint analysis suggests many interesting hypotheses about the expected performance behavior of all these retrieval functions. We design experiments to test these hypotheses using different types of queries and different document collections. We find that in many cases the empirical results are indeed consistent with these hypotheses. Specifically, when a constraint is not satisfied, it often indicates non-optimality of the method. This is most evident from the analysis of Okapi formula, based on which we successfully predict the non-optimality for verbose queries. In some other cases, when a method only satisfies a constraint for a certain range of parameter values, its performance tends to be poor when the parameter is out of this range, which is evident in the analysis of the pivoted normalization and the Dirichlet prior. In general, we find that the empirical performance of a retrieval formula is tightly related to how well they satisfy these constraints. Thus the proposed constraints can provide a good explanation of many empirical observations (e.g., the relatively stable performance of the Okapi formula) and make it possible to evaluate any existing or new retrieval formula *analytically*, which is extremely valuable for testing new retrieval models. Moreover, when a constraint is not satisfied by a retrieval function, it also suggests a possible way to improve the retrieval formula.

There are many interesting future research directions based on this work. First, it will be interesting to repeat all the experiments by removing the stop words with a standard list to see if the way a retrieval formula treats stop words might have an impact on the results. Second, since our constraints do not cover all the desirable properties, it would be interesting to explore additional necessary heuristics for a reasonable retrieval formula. This will help us further understand the performance behavior of different retrieval methods. A more ambitious direction is to develop a constraint-based methodology for studying retrieval models (e.g., along the line of [4]). Third, we will apply these constraints to many other retrieval models proposed in the literature [1] and different smoothing methods for language models as well [20]. Previous work [11, 21] has attempted to identify an effective retrieval formula through extensive empirical experiments, but the results are generally inconclusive with some formulas performing better under some conditions. Analysis of formalized retrieval constraints as explored in this paper may shed some light on what these conditions are exactly. Finally, the fact that none of the existing formulas that we have analyzed can satisfy all the constraints unconditionally suggests that it would be very interesting to see how we can improve the existing retrieval methods so that they would satisfy *all* the constraints, which presumably would perform better empirically than these existing methods.

6. ACKNOWLEDGMENTS

We thank anonymous reviewers for their useful comments. We also thank Robin Dhamanka and Jing Jiang for their help to improve the English in this paper.

7. REFERENCES

[1] G. Amati and C. J. V. Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems*, 20(4):357–389, 2002.

[2] N. Fuhr. Language models and uncertain inference in information retrieval. In *Proceedings of the Language Modeling and IR workshop*.

[3] N. Fuhr. Probabilistic models in information retrieval. *The Computer Journal*, 35(3):243–255, 1992.

[4] J. Kleinberg. An impossibility theorem for clustering. In *Advances in NIPS 15*, 2002.

[5] J. Lafferty and C. Zhai. Probabilistic relevance models based on document and query generation. In W. B. Croft and J. Lafferty, editors, *Language Modeling and Information Retrieval*. Kluwer Academic Publishers, 2003.

[6] J. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the ACM SIGIR'98*, pages 275–281, 1998.

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

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

[9] S. Robertson and S. Walker. On relevance weights with little relevance information. In *Proceedings of SIGIR'97*, pages 16–24, 1997.

[10] G. Salton. *Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer*. Addison-Wesley, 1989.

[11] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24:513–523, 1988.

[12] G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

[13] G. Salton, C. S. Yang, and C. T. Yu. A theory of term importance in automatic text analysis. *Journal of the American Society for Information Science*, 26(1):33–44, Jan-Feb 1975.

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

[15] H. Turtle and W. B. Croft. Evaluation of an inference network-based retrieval model. *ACM Transactions on Information Systems*, 9(3):187–222, 1991.

[16] C. J. van Rijbergen. A theoretical basis for theuse of co-occurrence data in information retrieval. *Journal of Documentation*, pages 106–119, 1977.

[17] C. J. van Rijsbergen. A non-classical logic for information retrieval. *The Computer Journal*, 29(6), 1986.

[18] E. Voorhees and D. Harman, editors. *Proceedings of Text REtrieval Conference (TREC1-9)*. NIST Special Publications, 2001. http://trec.nist.gov/pubs.html.

[19] S. K. M. Wong and Y. Y. Yao. On modeling information retrieval with probabilistic inference. *ACM Transactions on Information Systems*, 13(1):69–99, 1995.

[20] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of SIGIR'01*, pages 334–342, Sept 2001.

[21] J. Zobel and A. Moffat. Exploring the similarity space. *SIGIR Forum*, 31(1):18–34, 1998.