An Exploration of Formalized Retrieval Heuristics

Hui Fang, Tao Tao, and ChengXiang Zhai
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801
Email: {hfang,taotao,czhai}@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. Any effective retrieval formula, no matter how it is originally motivated, also often boils down to an explicit or implicit implementation of these heuristics. One basic research question is thus what are exactly these "necessary" heuristics that seem to cause good retrieval performance. In this paper, we present a *formal* study of these 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 only satisfied 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 they satisfy these constraints. Thus the proposed constraints can provide a good explanation of many empirical observations and make it possible to evaluate any existing or new retrieval formula *analytically*.

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 (Salton et al., 1975; Salton and McGill, 1983; Salton, 1989), probabilistic models(Robertson and Sparck Jones, 1976; van Rijbergen, 1977; Turtle and Croft, 1991; Fuhr, 1992; Ponte and Croft, 1998; Lafferty and Zhai, 2003), and logic-based models(van Rijsbergen, 1986; Wong and Yao, 1995; Fuhr, 2001). 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 (Voorhees and Harman, 2001). Even the language modeling approach has been shown to be connected with these heuristics (Zhai and Lafferty, 2001b). It thus appears that these heuristics are somehow *necessary* for achieving good retrieval performance. However, it is unclear at all what are *exactly* these "necessary heuristics" mathematically. A basic research question is then how can we *formally* define or 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 function should satisfy, and check these constraints on a variety of representative retrieval functions, representing the vector space model (pivoted normalization), the classic probabilistic retrieval model (Okapi), and the recently proposed language modeling approach (Dirichlet prior and KL-divergence). We find that none of the retrieval functions we studied satisfies all the constraints unconditionally, though some models violate more constraints or violate some constraints more seriously than others. Experimental results show that whether a retrieval formula satisfies these constraints is tightly related to its empirical performance behavior. More specifically, we find that when a retrieval method does not satisfy a certain constraint, it often indicates non-optimality of the method, and when a constraint is only satisfied for a certain range of parameter values, the method's performance would tend to be poor when the parameter is out of the range. Thus the proposed constraints provide a good explanation of many empirical observations about some retrieval methods. Moreover, these constraints make it possible to evaluate any existing or new retrieval formula *analytically* and shed light on how to further improve a retrieval formula.

The rest of the paper is organized as follows. We first present the four formal constraints in Section 2. We then apply these constraints to a variety of retrieval functions in Section 3. Finally, we discuss our findings and the future research directions in Section 4.

2 Formal Definition of Heuristic Retrieval Constraints

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

These four 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 document length normalization. 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. 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 functions do differ in their way to combine all these factors, even though their empirical performances may be similar.

These observations suggest that there are some "basic requirements" that all reasonable retrieval functions should follow. For example, if a retrieval function 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 heuristics intends to assign a high score to a document that has more occurrences of a query term, the document length normalization mechanism may cause a long document with high 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 heuristics may allow a document with a lower TF to "beat" the other with a much higher TF. So how can we regulate such interactions so that they will all be "playing a fair game?" Clearly, in order to answer this question, we must define what is a "fair game," i.e., we must define what is exactly a *reasonable* retrieval function.

Our idea is to characterize a reasonable retrieval function by listing the desirable constraints that any reasonable retrieval function must satisfy. We now formally define four such desirable constraints. Note that they 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 four basic constraints in this paper because they seem to capture the major known IR heuristics, particularly the TF-IDF weighting and length normalization.

Let us first introduce some notations. We use d_i to denote a document, and q to denote a query. w or w_i represents a term, and $c(w, d)$ is the counts 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. We are now ready to present the four constraints:

2.1 *Term Frequency* Constraint (TFC)

TFC: Let $q = w$ be a query with only one term w. If $c(w, d_1) > c(w, d_2)$ and $|d_1| \leq |d_2| + c(w, d_1) - c(w, d_2)$, then $f(d_1, q) > f(d_2, q)$.

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. As a minimum, if d_1 were generated by adding more occurrences of the query term to d_2, the score of d_1 should be higher than d_2; if the length of d_1 is shorter than this hypothesized d_1, then we have more reason for giving d_1 a higher score. Thus it captures the TF heuristic in a conservative way.

2.2 *Term Discrimination* Constraint (TDC)

TDC: Let q be a query and $w_1, w_2 \in q$ be two query terms. Assume $|d_1| = |d_2|$, $c(w_1, d_1) + c(w_2, d_1) = c(w_1, d_2) + c(w_2, d_2)$ and $c(w, d_1) = c(w, d_2)$ for all other words w. 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 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). Note that this constraint regulates the interaction between TF and IDF, and accurately describes the effect of using IDF in scoring. Clearly, weighting each term with an IDF factor does not imply that this constraint is 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 (LNC)

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 all other word w, $c(w, d_2) = c(w, d_1)$, then $f(d_1, q) \geq f(d_2, q)$.
LNC2: Let $q = w$ be a query. If $\forall k > 1, |d_1| = k \cdot |d_2|$ and $c(w, d_1) = k \cdot c(w, d_2)$,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", thus intends to penalize long documents. The second constraint intends to avoid over-penalizing long documents, since it says that if we copy a document k times to form a new document. then the score of the new document should not be lower than the original document.

3 Apply constraints to different retrieval models

In the previous section, four necessary retrieval constraints have been proposed according to some commonly used retrieval heuristics. In this section, we apply these formally defined constraints to some specific retrieval functions that represent the vector space model, the classical probabilistic retrieval model, and the language modeling approach, and check whether they satisfy all these constraints. Through such analysis and some empirical evaluation, we show two benefits of these constraints: (1) They can provide an approximate bound for the parameters in a retrieval formula. (2) They can explain the performance difference in various retrieval models.

3.1 Pivoted Normalization Method

The pivoted normalization retrieval formula (Singhal, 2001) is one of the best performing vector space retrieval functions. In the vector space model, text is represented by a vector of terms. Documents are ranked by the distance, which is considered as "similarity", between the query vector and the document vector. Given a query, a document's score is proportional to its similarity to the query. According to (Singhal, 2001), the pivoted normalization retrieval function is

$$\sum_{t \in Q,D} \frac{1 + ln(1 + ln(tf))}{(1-s) + s\frac{dl}{avdl}} \cdot qtf \cdot ln\frac{N+1}{df}$$

where,
tf is the term's frequency in document
qtf is the term's frequency in query
N is the total number of documents in the collection
df is the number of documents that contain the term
dl is the document length
$avdl$ is the average document length
The results of constraint analysis for the pivoted normalization method are summarized in Table 1. "Yes" means a constraint is satisfied. "No" means it is not satisfied. "Cond" means the constraint is satisfied conditionally for any range of parameter; "$Cond.^*$" means the constraint is satisfied for a particular range of parameter values. We now examine some of the constraints that are not trivially satisfied in some detail.

Table 1. Constraints in Pivoted

TFC	TDC	LNC1	LNC2
Cond.	Cond.	Yes	$Cond.^*$

First,let us consider TFC constraint. Let $\delta = c(w, d_1) - c(w, d_2)$. Consider a common case when $d_1 = avdl$. It can be shown that the TFC constraint is equivalent to the following constraint on the parameter s:

$$s \quad \leq \quad l(c(w, d_1), \delta) \times avdl$$

where

$$l(x, \delta) = \frac{g(x) - g(x - \delta)}{(1 + g(x)) \times \delta}$$

and $g(x) = ln(1 + ln(x))$.

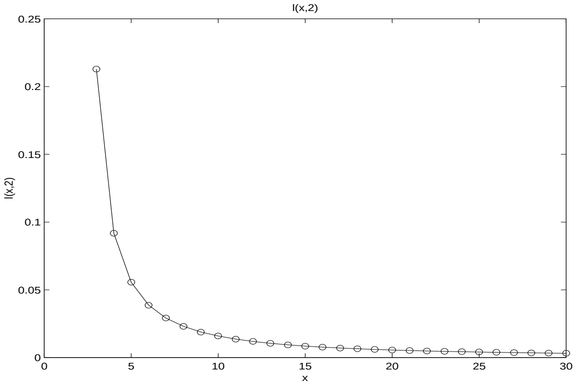

Figure 1. Plot for function $l(x, 2)$

This means that the TFC is satisfied only if s is below certain upper bound. To have some sense of what this bound is, we show the plot of l(x,2) (i.e., $\delta = 2$) in Figure 1. It is clear that when $c(w, d_1)$ is larger,TF constraint may provide an upper bound for s, that is $s \leq l(c(w, d_1), 2) \times avdl$. On the other hand, when $c(w, d_1)$ is small, TF constraint does not provide any effective bound for s,since $s \leq 1$.

Next, we check the TDC constraint. Let $c = c(w_1, d_1) + c(w_2, d_1)$. It can be shown that, when $idf(w_1)$ and $idf(w_2)$ are close, the TDC is equivalent to $h(c(w_1, d_1) - h(c(w_1, d_2)) \geq 0$, where

$$h(x) = ln(x) + ln(c - x) + ln(x) \times ln(c - x)$$

By analyzing $h(x)$, we see that when $x \leq c/2$, h(x) monotonically increases; when $x \geq c/2$, however, h(x) is monotonically decreasing. A plot of h(x) for $c = 10$ is shown in Figure 2. This means that when $idf(w_1)$ and $idf(w_2)$ are close, the TDC is satisfied only if $c(w_1, d_1) \leq c(w_2, d_1)$. It is interesting to see that the TDC, which is essentially capturing the IDF heuristics, is not unconditionally satisfied even for a highly effective TF-IDF scoring formula!

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

$$\frac{1 + ln(1 + c(w, d))}{1 - s + s\frac{dl}{avdl}} qtf(w)idf(w) \leq \frac{1 + ln(1 + kc(w, d))}{1 - s + sk\frac{dl}{avdl}} qtf(w)idf(w)$$

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

$$s \leq \frac{tf_2 - tf_1}{(k\frac{dl}{avdl} - 1)tf_1 - (\frac{dl}{avdl} - 1)tf_2} \qquad (1)$$

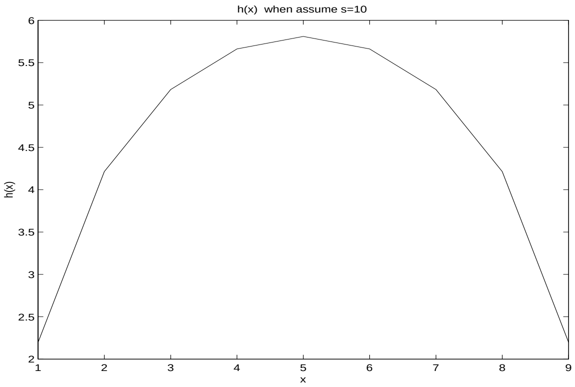

Figure 2. Plot for function h(x) when assume c=10

where

$$\begin{aligned} tf_2 &= 1 + ln(1 + kc(w, d)), \\ tf_1 &= 1 + ln(1 + c(w, d)). \end{aligned}$$

Again, in order to get a sense of what the bound is exactly, let us consider a common case when $|d_2| = avdl$. We have

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

It can be shown that the bound becomes tighter when k increases or when the TF is larger. A plot of the bound for $k = 2$ (i.e., double the document) is shown in Figure 3. This bound shows that in order to avoid over-penalizing a long document, reasonable value for s should be generally small – it should be below 0.4 even in the case of a small k ($k = 2$), and we know that for large k the bound would be even tighter. This analysis thus suggests that the performance can be bad for a large s.

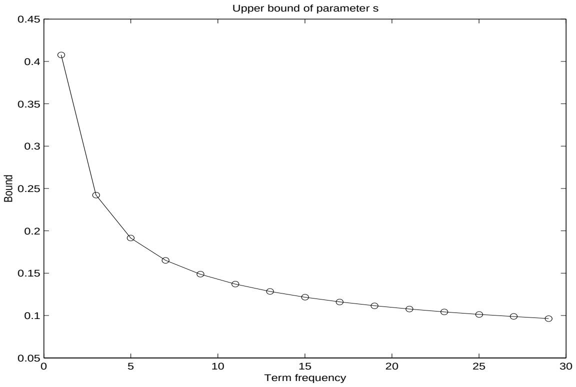

Figure 3. Upper bound of parameter s.

In order to verify this hypothesis, we tested the method on several different collections and for several different types of queries. To cover different types of queries, we follow (Zhai and Lafferty, 2001b) , 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

Table 2. Optimal s (for average precision) in 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

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).
Under this carefully designed experiment setting, the optimal value of s for average precision has been found to be indeed quite small in all cases (shown in Table 2). Moreover, we also see that when s is large, that is out of the range where the method satisfies the LNC2 constraint, the performance is significantly worse. In Figure 4, we show how the average precision is influenced by the parameter value in pivoted normalization method on AP document set and long-keyword query; the curves are similar in all other cases.

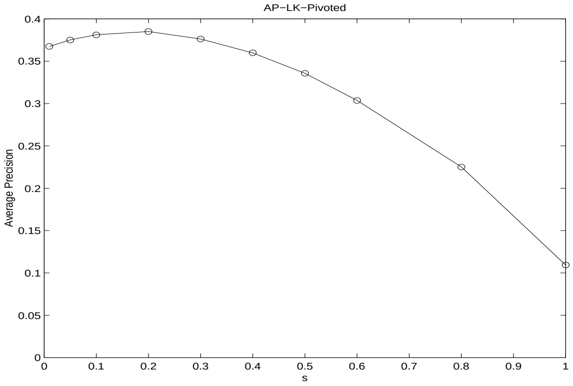

Figure 4. Performance of Pivoted for AP-LK.

Therefore,it seems that the constraints could provide an empirical bound for the parameter in the retrieval formula and the methods's performance would tend to be poor when the parameter is out of the bound.

3.2 Okapi

The Okapi retrieval function is another highly effective retrieval formula that represents the classical probabilistic retrieval model (Robertson and Walker, 1994). The Okapi retrieval function as presented in (Singhal, 2001) is

$$\sum_{t\in Q,D} ln\frac{N-df+0.5}{df+0.5}\times\frac{(k_1+1)tf}{k_1((1-b)+b\frac{dl}{avdl})+tf}\times\frac{(k_3+1)qtf}{k_3+qtf}$$

where k_1 (between 1.0-2.0), b (usually 0.75), and k_3 (between 0-1000) are constants.
We check all the constraints on the Okapi formula and summarize the results in Table 3.

Table 3. Summary of Constraints in Okapi

Query Type	TFC	TDC	LN1	LN2
Any	Cond.	Cond.	Yes	Yes
Verbose	No	Cond.	Yes	Yes
Keyword	Yes	Cond.	Yes	Yes

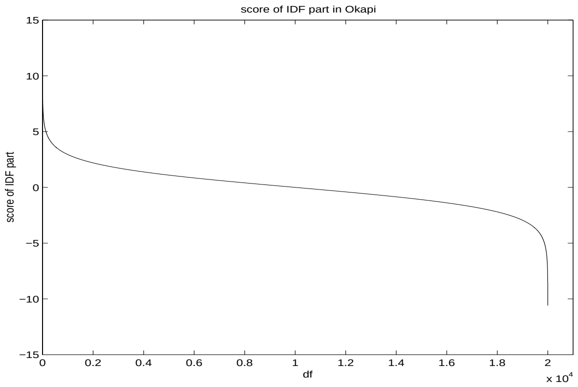

Figure 5. IDF score in Okapi.

Let us check the TFC constraint first. 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 (Robertson and Walker, 1997). Figure 5 shows the IDF score in Okapi formula when we assume the number of documents in the whole collection to be $20,000$. The x-axis is the number of documents that contain the term. The y-axis is the score of the Okapi IDF function. It is clear that when the document frequency is large, the IDF value would be negative.

When the IDF part is negative, Okapi formula would definitely violate the TFC constraint. However, this would have an impact only when a query term has a very high document frequency, i.e., when the query is verbose or somehow uses some general terms. When the IDF part is positive (usually for keyword query), the TFC constraint is shown to be equivalent to

$$s \leq \frac{1}{1 - \frac{|d_1| - c(w,d_1)}{avdl}}$$

Since $d_1 \geq c(w, d_1)$, so it is equivalent to $s \leq 1$, which is obviously satisfied. Therefore, the TFC is satisfied as long as the IDF value is positive. This analysis suggests that the performance of Okapi may be worse for verbose queries than for keyword queries.

Next, we consider the TDC constraint. Again, by making the assumption that $idf(w_1)$ and $idf(w_2)$ are very close, we can show that the TDC constraint is equivalent to $c(w_2, d_2) - c(w_1, d_1) \geq 0$. Although this constraint is also conditionally satisfied, unlike in the pivoted normalization method, it does not provide a bound for the parameter b. Therefore, the performance of Okapi can be expected to be more stable than pivoted normalization method.

To see if these hypotheses motivated by the constraint analysis are consistent with Okapi's actual performance, we test the Okapi method under the same experimental settings as in pivoted normalization. We 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 pivoted normalization. We also see that for keyword query, the performances of these two methods are similar. However, for verbose query, the performance of Okapi is much worse, which may be caused by the negative IDF score for common words. To see if this is true, we replace the IDF part in Okapi with the IDF part of the pivoted normalization formula, and the performance is improved significantly for the verbose queries. See Figure 6 and Figure 7 for plots of these comparisons.

It is clear from Figure 7 that satisfying more constraints appear to be correlated with a better performance. Moreover, 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.

3.3 Dirichlet Prior Method

The Dirichlet prior retrieval method is one of the best performing language modeling approaches (Zhai and Lafferty, 2001b). This method uses the Dirichlet prior smoothing method to smooth a document language

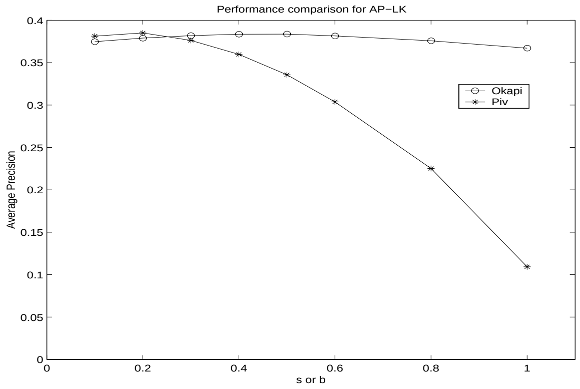

Figure 6. Performance Comparison between Okapi and Pivoted for AP-LK.

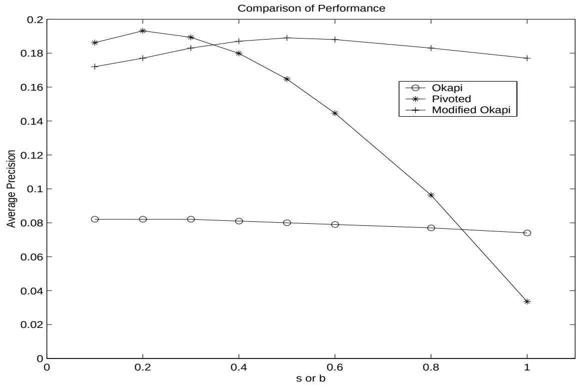

Figure 7. Performance Comparison between modified Okapi, Okapi and Pivoted for AP-SV.

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

$$\sum_{t \in Q,D} qtf \cdot ln(1 + \frac{tf}{\mu \cdot p(t|C)}) + ql \cdot ln\frac{\mu}{dl + \mu}$$

where, ql is the query length, and $p(t|C)$ is the probability of term t given by the collection language model. $p(t|C)$ indicates how popular term t is in the whole collection, thus is quite similar to the document frequency df.

After checking all constraints on Dirichlet Prior method, we summarize the results in Table 5

It is straightforward to verify that the TFC and LN1 are both satisfied. The LNC2 can be shown to be equivalent to $c(w, d) \geq |d| \cdot p(w|C)$, which is usually satisfied for content-carrying words. For non-informative high-frequency words, even if the constraint is not satisfied, slightly over-penalization may not hurt retrieval performance. Thus, compared with pivoted normalization, Dirichlet prior appears to have a more robust length normalization mechanism, even though both satisfy the LNC2 constraint conditionally.

Another interesting observation is the TDC constraint leads to some interesting lower bound for parameter μ, which is derived as follows.

Assume $p(w_1|C) \leq p(w_2|C)$ (roughly equivalent to $idf(w_1) > idf(w_2)$). TDC implies

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

Table 4. Optimal μ (for average precision) in 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

Table 5. Summary of Constraints in Dirichlet

TFC	TDC	LNC1	LN2
Yes	$Cond.^*$	Yes	Cond.

After some simplification, with the help of the constraint $c(w_1, d_1) + c(w_2, d_1) = c(w_1, d_2) + c(w_2, d_2)$, 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 how low this bound could be, 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). Thus, 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 in order to balance TF and IDF. In general, the analysis shows that μ has some lower bound, and a very small μ might cause poor retrieval performance.

In order to test this hypothesis, once again, we use the same experimental setting as in the previous sections. The optimal values of μ in Dirichlet are shown in Table 4. 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 8. Clearly, when μ is larger than a specific value, the performance keeps stable. However, when μ is small, the performance is noticeably worse. Thus, just like what we have found in the experimental part of pivoted normalization method, the constraint analysis here also suggests some bound for the retrieval parameter, and the methods's performance would be poor if the parameter is out of the bound.

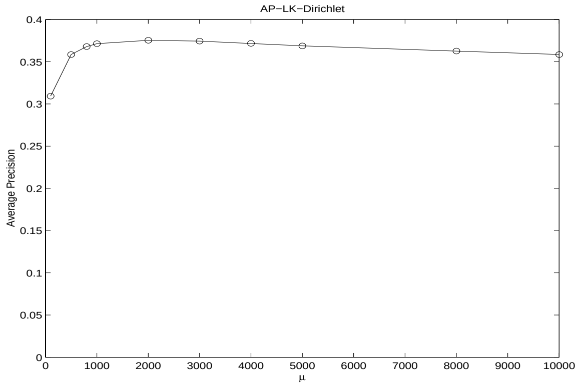

Figure 8. Performance of Dirichlet for AP-LK.

3.4 Divergence retrieval formulas

Since the emerging language modeling approaches to information retrieval have led to several interesting new effective retrieval formulas, we further examine a few more retrieval functions that use language models. In particular, we assume that a query and a document are both generated by a hidden unigram model, and the retrieval task is to estimate the generative models and then to compare the model similarity between a query model and document models. In general, many different similarity functions are possible. In (Lafferty and Zhai, 2001; Zhai and Lafferty, 2001a), the Kullback-Leibler (KL) divergence has been used. However, the (asymmetric) KL divergence is only of the many divergence-based measures (Lin, 1991). In this section, we use our constraints to evaluate some of these alternative divergence functions. We also present some preliminary experimental results using these divergence functions for retrieval. It is interesting to see that the constraint analysis can suggest which function is unlikely to perform well, and this prediction is indeed consistent with the empirical results.

We first give the definition of the basic KL divergence as other divergence functions can all be expressed in terms of it. Given two probability mass functions $p_1(x)$ and $p_2(x)$, $D(p_1||p_2)$, the KL divergence (or relative entropy) between p_1 and p_2 is defined as

$$D(p_1||p_2) = \sum_x p_1(x) \log \frac{p_1(x)}{p_2(x)}$$

It is easy to show that $D(p_1||p_2)$ is always non-negative and is zero if and only if $p_1 = p_2$ (Cover and Thomas, 1991). In retrieval, p_1 and p_2 usually correspond to the query language model and the document language model.

Assume that a query q is generated by a unigram language model $\theta_Q)$ and a document d is generated by a unigram model θ_D. For any word w, its probability in the query is $p(w|\theta_Q)$, and its probability in a document is $p(w|\theta_D)$. Dirichlet prior smoothing is applied to estimate θ_D, giving the following probabilities.

$$\begin{cases} p(w|\theta_D) = \frac{c(w,d)+\mu p(w|C)}{|d|+\mu} & \text{w is seen} \\ p(w|\theta_D) = \frac{\mu p(w|C)}{\mu+|d|} & \text{w is unseen} \end{cases}$$

where $c(w,d)$ is the counts of word w in document d, μ is a parameter, and $p(w|C)$ is the probability of w in the whole collection. To make the analysis more tractable, we further make a simplification assumption that when taking the sum in any divergence formula, we only consider the query terms (as if they are the only terms in the term space). This also allows us to use the maximum likelihood estimator to estimate the query model θ_Q. While this may lead to a non-optimal retrieval formula, it is unlikely to affect the performance significantly, and the constraint analysis would still be interesting.

We consider and evaluate the following four different divergence functions.

1. $\mathcal{KL}$ divergence

   $$\mathcal{KL}(q,d) = D(\theta_Q||\theta_D)$$

2. $\mathcal{J}$ divergence

   $$\mathcal{J}(q,d) = \mathcal{D}(\theta_Q||\theta_D) + \mathcal{D}(\theta_D||\theta_Q)$$

   It is the symmetric form of the $\mathcal{KL}$ divergence.

3. $\mathcal{K}$ divergence

   $$\mathcal{K}(q,d) = \mathcal{D}(\theta_Q||\theta_{\frac{q+d}{2}})$$

   where $p(w|\theta_{\frac{q+d}{2}}) = (p(w|\theta_Q) + p(w|\theta_D))/2$. It takes $\mathcal{KL}$ divergence between query model and the average model of the query and the document.

4. Jensen-Shannon divergence

$$\mathcal{JS}_{\pi}(q,d) \quad = \quad \pi\mathcal{D}(\theta_Q||\theta_{\pi q+(1-\pi)d}) + (1-\pi)\mathcal{D}(\theta_D||\theta_{\pi q+(1-\pi)d})$$

where $\pi \in (0,1)$ is a parameter to adjust the weights of averaging. $\theta_{\pi q+(1-\pi)d}$ is an average model of the query and document such that $p(w|\theta_{\pi q+(1-\pi)d}) = \pi p(w|\theta_Q) + (1-\pi)p(w|\theta_D)$.

All divergence functions are nonnegative. They reach the value 0 if and only if the two distributions are identical. In order to give relevant documents larger values, we use negative divergence $-\mathcal{KL}$, $-\mathcal{J}$, $-\mathcal{K}$, and $-\mathcal{JS}$ as the scores.

Table 6 shows the results of applying the constraints to all these divergence functions.

Table 6. Constraint results for divergence retrieval model

Retrieval model	TFC	TDC	LNC1	LNC2
$\mathcal{KL}$	Yes	Cond.	Yes	Cond.
$\mathcal{J}$	Yes	-	Yes	Cond.
$\mathcal{JS}$	No	-	No	Cond.
$\mathcal{K}$	Yes	Cond.	Yes	Cond.

The table shows that the three divergence measures $\mathcal{KL}$, and $\mathcal{K}$ share the same properties. They all satisfy TFC and LNC1 constraints and conditionally satisfy TDC and LNC2 constraints. $\mathcal{J}$ shares the properties with $\mathcal{KL}$ and $\mathcal{K}$ except that its TDC constraint is hard to verity. We leave its TDC constraint for further exploration in the future. However, the JS divergence is quite different; it failed to satisfy the very basic TFC and LNC1. Some details about the analysis of the JS divergence are given below. Note that we use $-\mathcal{JS}$ for scoring.

First, consider a single term query, $p(w|\theta_q) = 1$. Let $p = p(w|\theta_Q)$, and $\pi_1 = \pi$, $\pi_2 = 1-\pi$. We get $f(q,d) = -\mathcal{JS}(q,d) = \pi_2 p\log p - (\pi_1+\pi_2 p)\log(\pi_1+\pi_2 p)$. Since $\frac{\partial f}{\partial p} = \pi_2\log p - \pi_2\log(\pi_1+\pi_2 p) <= 0$. , we see that $f(q,d)$ is a monotonically decreasing function in terms of p. Thus the negative JS divergence violates TFC as well as LNC1. Second, in order to check the LNC2 constraint, we set $|d_2| = k\cdot|d_1|$. Since the function is monotonically decreasing in terms of p, in order to make $f(q,d_2) \geq f(q,d_1)$, we need to let $p(w|d_2) \leq p(w|d_1)$. Thus we have the following derivation:

$$\begin{aligned}
& p(w|d_2) \leq p(w|d_1) \\
\Leftrightarrow \quad & \frac{k\cdot c(w,d_1)+\mu p(w|C)}{k\cdot|d_1|+\mu} \leq \frac{c(w,d_1)+\mu p(w|C)}{|d_1|+\mu} \\
\Leftrightarrow \quad & \text{it is monotonically decreasing in terms of k} \\
\Leftrightarrow \quad & \frac{\partial}{\partial k}\left(\frac{k\cdot c(w,d_1)+\mu p(w|C)}{k\cdot|d_1|+\mu}\right) \leq 0 \\
\Leftrightarrow \quad & c(w,d_1) \leq |d_1|\cdot p(w|C)
\end{aligned}$$

The last condition is seen to be precisely the opposite of the condition derived in the analysis of the Dirichlet prior method. Since a content word usually satisfies $c(w,d_1) \geq |d_1|\cdot p(w|C)$, the $\mathcal{JS}$ divergence generally would not satisfy the LNC2 constraint for content words. It is hard to give a theoretical analysis of the TDC constraint for $\mathcal{JS}$. However, since it does not satisfy the two basic constraints (TFC and LNC1), the analysis suggests that the $\mathcal{JS}$ divergence is likely to perform very poorly and significantly worse than the other three, which are expected to perform similarly.

We ran experiments on four different types of queries as mentioned in the analysis of the pivoted normalization method. The average precision of each divergence function is shown in Table 7.

It is interesting to note that the results indeed confirm the predictions given by our theoretical analysis. Specifically, $\mathcal{KL}$, $\mathcal{J}$, $\mathcal{JS}$ satisfy the same constraints requirement, and they perform similarly. On the other hand, the $\mathcal{JS}$ divergence cannot satisfy two basic constraints (TFC and $LNC2$), and it performs much worse than the other three.

Table 7. Constraint results for divergence retrieval model

	$\mathcal{KL}$	$\mathcal{J}$	$\mathcal{JS}$	$\mathcal{K}$
SK	0.1926	0.1925	0.0349	0.1926
LK	0.3191	0.3195	0.0757	0.3191
SV	0.1617	0.1613	0.0082	0.1617
LV	0.2474	0.2472	0.0102	0.2473

Table 8. Comparison between different retrieval models

Retrieval model	TFC	TDC	LNC1	LNC2
Pivoted	C_1	C_2	Yes	C_3^*
Dirichlet	Yes	C_4^*	Yes	C_5
Okapi	C_6	C_7	Yes	Yes
KL	Yes	C_4^*	Yes	C_5
J	Yes	-	Yes	C_5
JS	No	-	No	$\neg C_5$
K	Yes	C_4^*	Yes	C_5

3.5 Summary

We have applied our four constraints to seven different scoring functions. The results are summarized in Table 8. where a “Yes” means the corresponding model satisfies the particular constraint, a “No” means the corresponding model *DOES NOT* satisfy the particular constraint, 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

$$\begin{aligned}
C_1 &\Leftrightarrow s \le f(c(w,d_1),\delta) \times avdl \\
C_2 &\Leftrightarrow c(w_1,d_1) \le c(w_2,d_1) \\
C_3^* &\Leftrightarrow s \le \frac{tf_2 - tf_1}{(k\frac{dl}{avdl} - 1)tf_2 - (\frac{dl}{avdl} - 1)tf_1} \\
C_4^* &\Leftrightarrow \mu \ge \frac{c(w_1,d_1) - c(w_2,d_2)}{p(w_1|C) - p(w_2|C)} \\
&\qquad > avdl \times (c(w_1,d_1) - c(w_2,d_2)) \\
C_5 &\Leftrightarrow c(w,d) \ge |d| \cdot p(w|C) \\
\neg C_5 &\Leftrightarrow c(w,d) \le |d| \cdot p(w|C) \\
C_6 &\Leftrightarrow w \in \text{content words} \\
C_7 &\Leftrightarrow c(w_1,d_1) \le c(w_2,d_2)
\end{aligned}$$

We can make several interesting observations:

- All the language modeling approaches (i.e., Dirichlet and all divergence methods), except the JS divergence, satisfy all the constraints in the same way.
- It is somehow surprising to see that the seemingly reasonable JS divergence formula actually fails to satisfy some basic constraints and performs poorly.
- It is even more surprising that all the methods, including a highly effective TF-IDF model, fail to satisfy the TDC constraint (essentially the IDF heuristics) unconditionally.

- Among all the methods that we examined, the Okapi formula (with the IDF replaced by the normal IDF) appears to satisfy most of the constraints, and empirically, it also appears to have a more stable performance which is often better than or comparable to the best performance achieved by other methods.

4 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 four basic constraints that any reasonable retrieval function should satisfy, corresponding to three desirable intuitive constraints – term frequency constraint, term discrimination constraint and length normalization constraint. We check the four constraints on eight representative retrieval functions 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, and find that in many cases these hypotheses are indeed consistent with the empirical results. Specifically, when a constraint is not satisfied, it often indicates non-optimality of the method. This is most clear from the analysis of the divergence functions, which successfully predicts the non-optimality of the JS divergence function. 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 the range, as shown 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 good 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.

There are several interesting future research directions based on our work. First, 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. Second, we may apply these constraints to many other retrieval methods proposed in the literature, especially those using language models (e.g., different smoothing methods). 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.

References

Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*. Wiley.

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

Fuhr, N. (2001). Language models and uncertain inference in information retrieval. In *Proceedings of the Language Modeling and IR workshop*, pages 6–11.

Lafferty, J. and Zhai, C. (2001). Document language models, query models, and risk minimization for information retrieval. In *Proceedings of SIGIR'01*, pages 111–119.

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

Lin, J. (1991). Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151.

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

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

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

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

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

Salton, G. and McGill, M. (1983). *Introduction to Modern Information Retrieval*. McGraw-Hill.

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

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

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

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

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

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

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

Zhai, C. and Lafferty, J. (2001a). Model-based feedback in the KL-divergence retrieval model. In *Tenth International Conference on Information and Knowledge Management (CIKM 2001)*, pages 403–410.

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