The Effect of Normalization when Recall Really Matters

April Kontostathis and Scott Kulp
Department of Mathematics and Computer Science, Ursinus College, Collegeville PA USA

Abstract: *In this article we describe a series of search and retrieval experiments to compare the impact of two different types of weighting and normalization schemes on recall. We first describe both the state-of-the-art BM25 algorithm and the newly introduced power normalization algorithm in detail. We then determine the optimal parameter settings for both systems. Finally we compare their retrieval performance. Our results show that BM25 does not maximize recall for two different collections (TREC Legal and TREC Disks 1 & 2) and three different query sets. Designers of applications which require maximal recall, such as legal discovery, should consider using power normalization instead of BM25.*

Keywords: Search and Retrieval, Normalization, Term Weighting

1 Introduction

Most search and retrieval applications, such as web search, are concerned primarily with returning relevant documents at top ranks only, since users rarely look past the first page of query results. However, there are some search and retrieval applications, such as legal discovery, medical diagnosis, and prior research identification, which are much more concerned with returning *all* relevant materials for a query. In other words, while high precision is helpful, high recall is absolutely critical for the users of these systems.

In this article, we compare a straight-forward term weighting and document normalization scheme, power normalization [4], to the BM25 algorithm [5] to retrieve documents in the legal domain. We show that, while BM25 outperforms power normalization at top ranks, power normalization returns more relevant documents before rank 5000, and also returns these documents at much lower thresholds. We use a second, more general purpose, corpus to verify these results.

In the next section we discuss the datasets and metrics we used for our experiments. Section 3 describes the weighting and normalization schemes we tested. Section 4 describes our experimental methodology and results. We offer our conclusions in Section 5.

2 Datasets and Metrics

In this section we describe the datasets we used in our experiments, along with the metrics used to measure retrieval performance.

2.1 Datasets

The primary dataset we used for our experiments is the IIT Complex Document Information Processing test collection, version 1.0 (IIT CDIP 1.0). This collection consists of roughly 7 million documents, approximately 57 GB of uncompressed text, taken from the Legacy Tobacco Document Library hosted by the University of California at San Francisco. These documents were made public during various legal cases involving US tobacco companies as part of the settlement agreement.

In 2006, the Text REtrieval Conference (TREC) initiated a new track, TREC Legal, based on the IIT CDIP 1.0 test collection. In 2006, 43 topics were developed for this dataset. As a result of the 2006 competition, 39 of these queries have relevance judgment data available [1]. An average of 814 documents were judged for each of these 39 queries (low 539, high 937). An average of 111 were rated as relevant, 703 as non-relevant. There was a wide range of relevant documents between queries, however. Two queries only had 1 relevant document, and one had 502 relevant documents. We refer to this query set as LEGAL06 in our experimental results.

In 2007, 50 additional topics were developed for the IIT CDIP 1.0 data set. Of these, 43 had relevance judgement data, with an average of 568 judged documents for each query. The mean number of relevant documents was 101 (high 391, low 11). For each query, there were some documents for which relevance could not be determined. We considered these documents to be unjudged, and excluded them when computing precision and recall. We

refer to this query set at LEGAL07 in our experimental results section.

In addition to the legal data, we wanted to compare these weighting/normalization schemes on a more general dataset to see if the results were similar. Thus, we chose to run these experiments on the corpus that was used for the ad hoc retrieval task at the early TREC conferences (commonly referred to as Tipster or TREC disks 1 and 2). This dataset contains approximately 650,000 documents (2 GBs of uncompressed text) taken from eight different sources. The dataset was designed to include articles with varied length, writing style, and vocabulary [2]. We used the second set of 50 queries (and relevance judgment data) that were generated for the data set (queries 101-150) in the experiments we refer to as TREC1 below.

2.2 Metrics

We use standard recall and precision to measure retrieval performance. Although we are primarily interested in retrieving all relevant documents, we also would like to know how many documents we need to retrieve to get all the relevant ones. Thus we are interested in the rank where recall is maximized. In general, a document is said to be ranked r, if it is the r^{th} document returned by the retrieval system (it is assumed that the retrieval system will return documents in reverse order based on some scoring function that measures the relevance of a particular document to a given query). We measure recall and precision at rank r, by identifying the relevant documents with $rank <= r$. Precision and recall at rank r are defined as follows:

$$P_r = \frac{numRelevant}{r} \tag{1}$$

$$R_r = \frac{numRelevant}{totalNumberOfRelevant} \tag{2}$$

In these very large corpora it is impossible to judge every document for relevance to each query, so our system will return some unjudged documents. We take the standard approach and skip these documents when we do our counting. Thus, for our purposes, a document is said to be ranked r, if it is the r^{th} $judged$ document returned by the retrieval system.

3 Weighting and Normalization

In this section we describe the BM25 and the power normalization methods for weighting and normalization.

3.1 BM25

The BM25 algorithm was introduced at TREC 3 [5]. It combines an inverse collection frequency weight with collection specific scaling for documents and queries. The weight function for a given document (d) and query (Q) appears in Equation 3. In this equation, idf_t refers to the inverse document frequency weight for a given term (Equation 4), K appears in Equation 5, tf is the number of times term t appears in document d, qtf is the number of times term t appears in the query Q, N is the number of documents in the collection, n is the number of documents containing t, dl is the document length (we used words), adl is the average document length for the corpus (also words), and b, $k1$, and $k3$ are tuning parameters.

$$w_d = \sum_{t \in Q} idf_t \frac{(k1+1)tf}{(K+tf)} \frac{(k3+1)qtf}{(k3+qtf)} \tag{3}$$

$$idf_t = \frac{N - n + .5}{n + .5} \tag{4}$$

$$K = k1((1-b) + b\frac{dl}{adl}) \tag{5}$$

The full BM25 weighting scheme is a bit more complicated than this and requires two additional tuning parameters, but BM25 reduces to the above formula when relevance feedback is not used and when those two parameters take their standard values. Interested readers should refer to [5] for details.

In practice, retrieval performance with BM25 weighting is not particularly sensitive to the values given to b, $k1$ and $k3$. Common values for b range from .5 to .9, $k1$ ranges from 1 to 2.5, and $k3$ ranges from 2 to 10. We ran a performance loop for each collection and saw only minor differences in retrieval performance at various ranks as these parameters changed. We did notice, however, that BM25 is very sensitive to the average document length. We used the average length of just the judged documents in our experiments. Using the collection average document length significantly decreased the performance of BM25.

3.2 Cosine Normalization

The most common form of document normalization in the early information retrieval literature is cosine normalization [7]. The score using cosine normalization weighting for each document/query pair is given in Equation 6, where w_d is the final document weight, dtw is the term weight within the document, before normalization, nd is the number of unique terms in document d, qtw is the term weight within the query, before normalization, and nq is the number of unique terms in the query. Cosine normalization ensures that each document in the collection is given equal weight during the query process.

$$w_d = \sum_{t \in Q} \left(\frac{dtw}{\sqrt{\sum_{i=1}^{nd} dtw_i^2}} \frac{qtw}{\sqrt{\sum_{i=1}^{nq} qtw_i^2}}\right) \quad (6)$$

Since BM25 includes both a term weighting component and a normalization factor, we needed to choose a weighting scheme for our cosine and power normalization experiments. We chose the log entropy weighting scheme, described in [6]. For the log-entropy weighting scheme, we calculate both a term's local weight, or how important the term is to a specific document, and the term's global weight, or how important the term is to the data set as a whole. The local weight of term i in document j is defined by Equation 7, where f_{ij} is the frequency of term i in document j. This way, as the importance of a term is kept relatively constant as its frequency within a document becomes very large. The global weight of term i is defined by Equation 8, where n is the number of documents in the collection and f_i is the total frequency of term i among all documents in the data set. The final weight for each term in a specific document (or query) is the product of the local and global weights (Equation 9).

$$l_{ij} = log(1 + f_{ij}) \quad (7)$$

$$g_i = \frac{\Sigma_{i=1}^{n}(f_{ij}/f_i) * log(f_{ij}/fi)}{log(n)} + 1 \quad (8)$$

$$dtw = l_i g_i \quad (9)$$

3.3 Power Normalization

The power normalization algorithm was first described in [4]. Additional experiments appear in [3]. Power normalization is similar to cosine normalization, but instead of making the length of all documents equal, longer documents are given a slight advantage over shorter documents. The extent of the advantage is determined by a single parameter. The document score is determined by Equation 10. Here dc is total term count for terms in document d, qc is the total term count for the query and p is the power. For example, if $p = .333$ the normalization factor is approximately the cube root of the number of terms in a given document d.

$$w_d = \sum_{t \in Q} \left(\frac{dtw}{dc^p} \frac{qtw}{qc^p}\right) \quad (10)$$

Experiments show that power normalization using $p = .333$ and $p = .25$ results in some improvement over cosine normalization at low ranks for the TREC Legal collections [4]. A similar method, which we will refer to as log normalization uses $log(dc)$ as a normalization factor, and is shown in Equation 11.

Table 1: Optimal Parameter Settings for BM25

Collection	b	k1	k3	Recall
LEGAL06	.7	2.5	2	.928
LEGAL07	.5	2	4	.916
TREC1	.6	1	2	.942

$$w_d = \sum_{t \in Q} \left(\frac{dtw}{\log(dc)} \frac{qtw}{\log(qc)}\right) \quad (11)$$

A close look at Equations 6-11 shows that the normalization factor can be pulled out of the summation. However, since all documents have different lengths, the factor will be different in all documents, resulting in slight differences in the final scoring for each document. Power normalization and log normalization give long documents very similar lengths, because these functions flatten out when $p < 1$ [4].

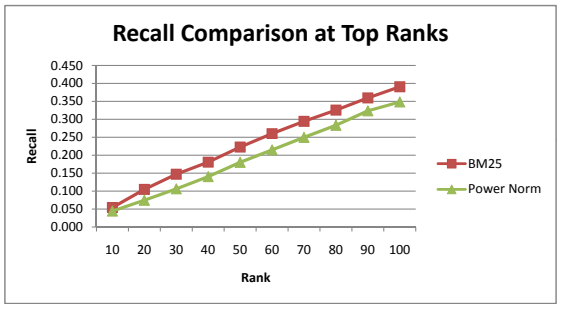
4 Experiments and Results

In this section we describe the experimental design and results.

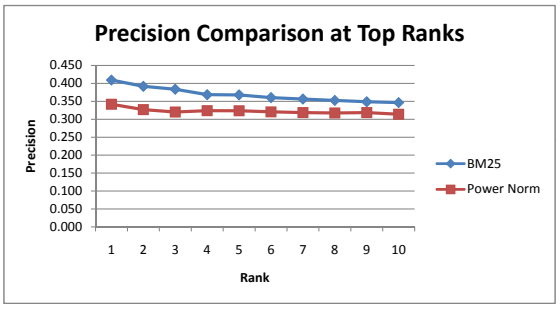
4.1 Experimental Design

Since the BM25 algorithm requires three tuning parameters, we began by running an optimization loop on these three parameters. We tried all possible combinations of the following: values for b ranging from .5 to .9 in .1 increments, values for $k1$ ranging from 1 to 2.5 in .5 increments, and values for $k3$ ranging from 2 to 10 in increments of 2. Thus, a total of 100 experiments were run for each collection. In each run we measured precision and recall at ranks 10 through 5000 in increments of 10. To identify the optimal parameter settings, we then looked at the maximum recall at 5000 for each set of parameters. LEGAL07 and LEGAL06 had multiple runs of parameters with this maximum recall, so we used the average recall across all ranks (10-5000) as a tie breaker. The parameters with the best recall were chosen for comparison with the power normalization schemes. These settings appear in Table 1.

We compared BM25 to a variety of the power normalization approaches for each collection. We looked at cosine and log, as well as power normalization with $p = .25$ (fourth root), $p = .333$ (cube root), $p = .5$ (square root), and a value of p that was determined to be optimal. We identified the optimal p for each collection by measuring mean average precision for each collection as p ranged from .02 to 1 (incrementing by .02). Optimal p for LE-



(a) Recall.



(b) Precision.

Figure 1: Precision and Recall Comparison at top ranks for TREC Legal 2007

Table 2: Recall Comparison for TREC Legal 2007

Description	Recall	Rank	Precision
BM25	.916	4990	.019
Cosine	.959	800	.122
Log	.959	770	.126
SquareRt	.959	830	.117
CubeRt	.959	770	.127
FourthRt	.959	760	.128
Optimal ($p = .28$)	.959	760	.128

Table 3: Recall Comparison for TREC Legal 2006

Description	Recall	Rank	Precision
BM25	.928	4990	.021
Cosine	.946	900	.115
Log	.946	900	.115
SquareRt	.946	900	.115
CubeRt	.946	900	.115
FourthRt	.946	900	.115
Optimal ($p = .36$)	.946	900	.115

GAL06 was .36, for LEGAL07 was .28, and for TREC1 was .32. [3].

4.2 Results

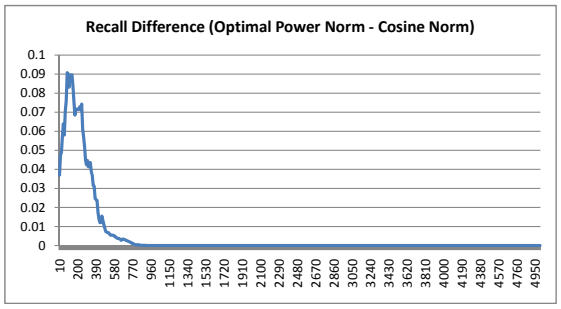
As expected, BM25 provided better recall and precision at top ranks. A comparison of optimal BM25 to optimal power normalization at top ranks for TREC 2007 appears in Figure 1. However, the maximum value for recall presents a different picture. Table 2 shows a comparison between BM25 and various power normalization schemes for TREC Legal 2007. The first column of the table describes the weighting/normalization scheme. The next column shows the maximum recall achieved at any rank (10 to 5000). The third column indicates the lowest rank where the maximum recall is reached. The final column is the precision at this rank. It is clear from this data that power normalization retrieves more relevant documents overall, retrieves them at better ranks, and does so with higher precision. Furthermore, the choice of power normalization parameter does not seem to impact this result. For comparison, BM25 recall at rank 760 is .903, precision is .127.

In order to determine if these results were specific to these queries we also ran the 2006 Legal track queries and the results appear in Table 3. Oddly, the power norm schemes all produced the maximal recall value of .946 at rank 900, with corresponding precision of .115. The BM25 recall at rank 900 is .923 and precision is .114.

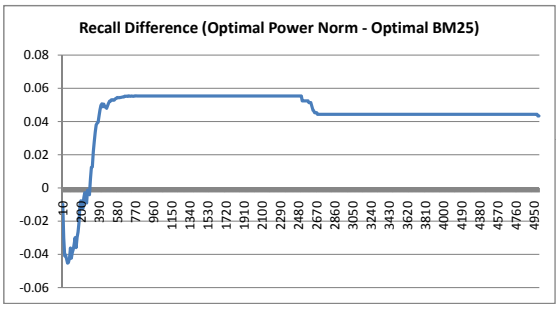
Table 4: Recall Comparison for TREC Disks 1 & 2

Description	Recall	Rank	Precision
BM25	.942	5000	.019
Cosine	.989	1720	.134
Log	.989	1730	.133
SquareRt	.989	1730	.133
CubeRt	.989	1720	.134
FourthRt	.989	1720	.134
Optimal ($p = .32$)	.989	1730	.133

To ensure the that results did not occur due to some characteristic inherent in the IIT CDIP 1.0 data set, we also tested using queries 101-150 for TREC Disks 1 & 2. The results for this data set appear in Table 4. The results are similar to the results for the Legal data sets. The only difference is that more relevant documents are retrieved overall by both BM25 and the power normalization techniques. Once again there is very little difference among the different power normalization schemes. BM25 recall at 1720 is .937, precision at this rank is .127.



(a) Power vs Cosine.



(b) Power vs BM25.

Figure 2: Comparing Power Norm to Cosine and BM25 - Recall

4.3 Analysis

Tables 2-4 show little difference between cosine normalization and the power normalization functions, so which would be preferred? If a user is concerned solely about maximizing recall, it does not matter which technique is used; however, as Figures 2 and 3 show, there are some advantages to using the power normalization algorithm in some situations.

These figures are charting the difference between power normalization at the optimal setting for p and cosine normalization (subfigure a) and the BM25 algorithm at the optimal values for b, $k1$, and $k3$ (subfigure b) for both recall (Figure 2) and precision (Figure 3). Precision and recall were measured at ranks 10-5000 (incrementing by 10), and the cosine (resp. BM25) values were subtracted from the Power normalization value. The TREC 2007 data set was used.

Comparing Power to Cosine shows the superiority of Power Normalization at the top ranks. Both recall and precision start out much higher when using power normalization over cosine normalization.

As already noted, BM25 outperforms power normalization in the top ranks and this trend can be seen in Figures 2 and 3. However, these figures can be used to identify the point at which power normalization surpasses BM25, which is at rank 300 for recall and around rank 430 for precision.

These studies can be used to identify the proper normalization technique for a given application. If users will look at results only in the very top ranks, BM25 is definitely the optimal choice. If users are willing to dig further to examine, for example, the top 1000 documents, there is not much difference between the various techniques. Identifying as many relevant documents as possible clearly requires the use of power or cosine normalization instead of BM25.

5 Conclusions

Often when several search and retrieval systems are compared, researchers focus on retrieval performance at top ranks. This is appropriate for a variety of applications, including web search, but some applications, particularly those involving legal, medical or patent data are more concerned with recall. Achieving higher recall rates at better (but not necessarily top) ranks would uncover important documents more efficiently.

We have shown that BM25 does not consistently provide the highest possible recall rates, and, in fact, simpler techniques can be used to improve recall. Furthermore, higher recall rates can be achieved at better ranks. Reading (or skimming 770) documents, knowing that almost 96% of the documents you want to find are in the results, is clearly preferrable to reading 4990 documents to identify the 91% that are available.

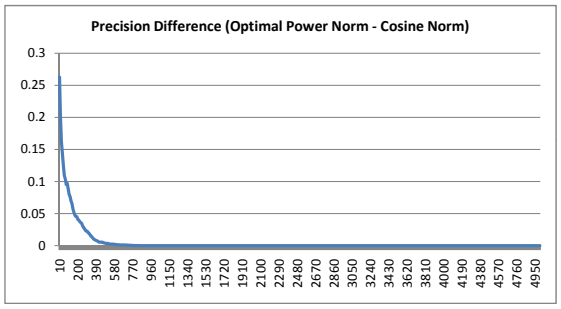
References

[1] Jason R. Baron, David D. Lewis, and Douglas W. Oard. TREC-2006 Legal Track Overview. In *Proceedings of the Fifteenth Text REtrieval Conference (TREC2006). NIST Special Publication 500-272.*

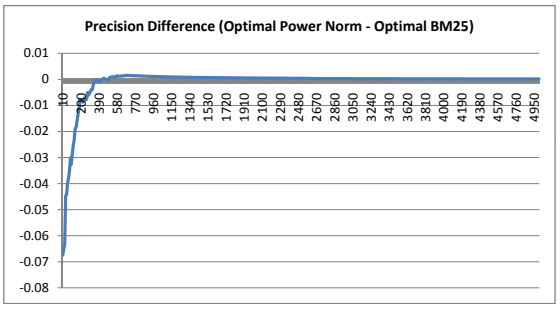
[2] Donna Harman. Overview of the First Text REtrieval Conference (TREC-1). In *Proceedings of the First Text REtrieval Conference (TREC-1)*, pages 1–20, 1992.

[3] Scott Kulp. Improving Search and Retrieval Performance through Shortening Documents, Detecting Garbage, and Throwing Out Jargon. Technical report, Ursinus College, Collegeville, PA, USA, http://webpages.ursinus.edu/akontostathis, 2007.

[4] Scott Kulp and April Kontostathis. On Retrieving Legal Files: Shortening Documents and Weeding Out



(a) Power vs Cosine.



(b) Power vs BM25.

Figure 3: Comparing Power Norm to Cosine and BM25 - Precision

Garbage. In *Proceedings of the Sixteenth Text REtrieval Conference (TREC2007). NIST Special Publication 500-274.*

[5] Stephen E. Robertson, Steve Walker, Micheline Hancock-Beaulieu, Aarron Gull, and Marianna Lau. Okapi at TREC-3. In *Proceedings of the Third Text REtrieval Conference (TREC-3). NIST Special Publication 500-226*, pages 109–126, 1994.

[6] Gerard Salton and Chris Buckley. Term Weighting Approaches in Automatic Text Retrieval. Technical report, Cornell University, Ithaca, NY, USA, 1987.

[7] C. J. Van Rijsbergen. *Information Retrieval, 2nd edition*. Dept. of Computer Science, University of Glasgow, 1979.