Document Length Normalization

Amit Singhal*
Gerard Salton
Mandar Mitra
Chris Buckley

Department of Computer Science
Cornell University, Ithaca, NY 14853
{singhal, gs, mitra, chrisb}@cs.cornell.edu

Abstract

In the TREC collection – a large full-text experimental text collection with widely varying document lengths – we observe that the likelihood of a document being judged relevant by a user increases with the document length. We show that a retrieval strategy, such as the vector-space cosine match, that retrieves documents of different lengths with roughly equal probability, will not optimally retrieve useful documents from such a collection. We present a modified technique that attempts to match the likelihood of retrieving a document of a certain length to the likelihood of documents of that length being judged relevant, and show that this technique yields significant improvements in retrieval effectiveness.

1 Introduction

Text REtrieval Conference, or TREC, is an ARPA and NIST co-sponsored effort that brings together information retrieval researchers from around the world to discuss their systems, and to develop a large test-bed for automatic retrieval systems in the process. [9, 10, 11] The third in this series of conferences, TREC–3, attracted thirty three different participants — nineteen universities and fourteen commercial organizations. With such a large participation of various IR researchers, a large and varied collection of full-text documents, a large number of user queries, and a superior set of independent relevance judgments, TREC has rightfully become the standard test collection for current information retrieval research. TREC has facilitated an objective and comparative

*This study was supported in part by the National Science Foundation under grant IRI-9300124.

evaluation of various information retrieval techniques on a realistic database, that were previously never evaluated under one umbrella.

Term weighting is one of the most important parts of a retrieval system. Different term weighting approaches have shown promise in the TREC environment. In TREC–3, the three systems with the best base performance (when only term weights and query–document matching is used) were Okapi [14, 15], INQUERY [8] and Cornell's Smart[1]. [19, 2] These three systems are statistical systems whose term weighting approaches reduce to variations of the $tf \times idf$ weighting approach, [24, 20] each working from a very different basis. Okapi uses a probabilistic technique, based on approximations to the 2–Poisson model, to get the document and the query term weights. INQUERY is based on the inference network model and uses another probabilistic technique to estimate the importance of a term in a document. [29] Smart, on the other hand, uses the classical $tf \times idf$ approach to term weighting. In the TREC–3 task, Okapi had the highest performance, followed by INQUERY, and then by Smart. Since all three systems use some variation of the $tf \times idf$ term weights, one asks what is causing the noticeable differences in their performance? In this study, we focus on the most significant difference between Smart's term weighting scheme and the term weights used by Okapi and INQUERY, namely Smart's use of the cosine normalization function [22, 23] for term weight normalization.

TREC is very different from the previous test collections used in information retrieval. The previously used collections had few documents, mostly small, with very little variation in the document lengths. The TREC collection, however, has a large number of full-text documents with widely varying lengths. The smallest non-empty document in TREC disks one and two is document `DOE1-49-1252`, which is seventy bytes long, and has only one non-stop word. The largest document in this collection is document `FR89119-0111`, which is 2,637,276 bytes long and has 207,459 non-stop words. With such massive variation among the document lengths, the document length normalization aspect of document term weighting becomes extremely important in retrieval from the TREC collection.

Term weight normalization is used to remove the advantage that long documents have, in retrieval, over short documents. For years, researchers have worked with the implicit assumption that *relevance of documents is independent of document lengths*. Based on this assumption, researchers have attempted to remove length preferences in retrieval that might exist due to the variations

[1] Various other TREC participants have also used the Smart system for their experiments. In the present study, the term *Smart* invariably refers to Cornell's participation in TREC. [6, 4, 5]

in the document lengths. A study of the TREC collection reveals that longer documents have a higher chance of being judged relevant to a user-query, compared to shorter documents. Whether this characteristic of TREC holds across different information bases is an open question. It is plausible that the longer documents contain more information, and are, thereby, potentially more useful to a user, but this phenomenon has never been observed earlier. In this study, we show that instead of retrieving documents of all lengths with equal chances, we would like to retrieve documents of a given length with the same probability as the probability of finding a relevant document of that length.

We have observed that the weighting schemes of Okapi and INQUERY do, in fact, retrieve a greater number of long documents, as opposed to short documents. Results show that if the Smart system is modified to favor the longer documents in retrieval, its retrieval effectiveness improves significantly, and becomes comparable to the retrieval effectiveness of the best system in TREC. The main results of this study are:

- Contrary to the general assumption that the probability of relevance of a document to a query is independent of the document length, in the TREC collection, probability of relevance of a document increases with its length.
- A system that retrieves documents of a certain length with a probability similar to that of finding a relevant document of that length, will outperform other systems that retrieve documents with very different probabilities from their probability of relevance.

The rest of this study is organized as follows. Section two introduces term weighting and document ranking. Section three explains how the document length study was performed. Section four introduces pivoted-cosine normalization, a variant of cosine normalization used to favor longer documents in retrieval. Section five contains the results. Section six discusses the main observation from this study. Section seven concludes the study.

2 Term Weighting and Document Ranking

In automatic information retrieval, documents in a text collection are usually indexed by their terms. Terms are words, phrases, or any other indexing units used to identify the contents of a text. Since different terms have different importance in a text, an importance indicator – the *term weight* – is associated with every term. [27, 22] Terms that are less important for content identification are assigned lower weights, whereas terms that are more important in a text are assigned higher weights. Judicious assignment of term weights can substantially improve the search effectiveness of a system. [3] Several term weighting schemes have been proposed over years. Most of the schemes use the following factors to assign weight to a document term:

- the term frequency in the document,
- prevalence of the term in the entire collection, and
- the length of the document.

Typically, a term that occurs frequently in a text is more important in the text than an infrequent term. Therefore, the number of occurrences of a term (in a text), often called the *term frequency* or *tf*, is used as the term weight. Common words tend to occur in numerous documents in a collection, and are poor indicators of a document's content. The more documents a term occurs in, the less important it may be. Therefore, the weight of a term should be inversely related to the number of documents in which the term occurs, or the *document frequency* of the term. An *inverse document frequency*, or *idf*, factor is commonly used to incorporate this effect. [17]

Long and verbose documents usually use the same terms repeatedly. As a result, the term frequency factors may be large for long documents. Long documents also have numerous different terms. This increases the number of word matches between a query and a long document, increasing its chances of retrieval over shorter documents. To compensate for these effects, *normalization* of term weights is often used. Normalization is a way of penalizing the term weights for longer documents, thereby reducing, if not removing completely, the advantage that long documents have in retrieval. Different systems use different term weight normalization techniques.

When a user poses a query to the system, the query is indexed by its terms and weights are associated with the query terms. A numerical similarity is computed between the user query and all the documents in the collection. [20, 23] This numerical similarity is based on individual contributions from the various matching terms between the query and the document. Such a similarity supposedly measures the potential usefulness of a document for the user-query. The documents in the information base are ranked by their decreasing similarity to the query and are presented to the user in this order. Documents presented earlier in a search have a higher degree of vocabulary overlap to the user-query, and are potentially more relevant to the user's information need.

3 Studying Document Length Effects

Analysis of the probability of retrieval and the probability of relevance of documents based on the document lengths was the main tool used in this study. We started the document length analysis with a list of 741,856 TREC documents, sorted by increasing byte-lengths of the documents. We divided this sorted list into bins of one thousand documents each, yielding 742 different bins: the first 741 bins containing one thousand documents each, and the last bin containing the longest 856 documents. We selected the median document length in each bin to represent the bin on the graphs used in this study.

To study the lengths of the documents retrieved using a particular term weighting scheme, for each of the fifty TREC queries used in the ad-hoc task at TREC–3, we retrieved the top one thousand documents using the scheme. If the same document was retrieved more than once (for different queries), it was used more than once in the length analysis. This yielded 50,000 ⟨query, retrieved-document⟩ pairs for the document length analysis. We counted the number of documents retrieved from each of the 742 bins obtained above, in the 50,000 ⟨query, retrieved-document⟩ pairs. We then computed the probability of retrieving a document from a bin, which is the ratio of the number of documents retrieved from a particular bin and the total number of documents retrieved (50,000). In terms of conditional probability, given a document D, this ratio for the ith bin can be represented by

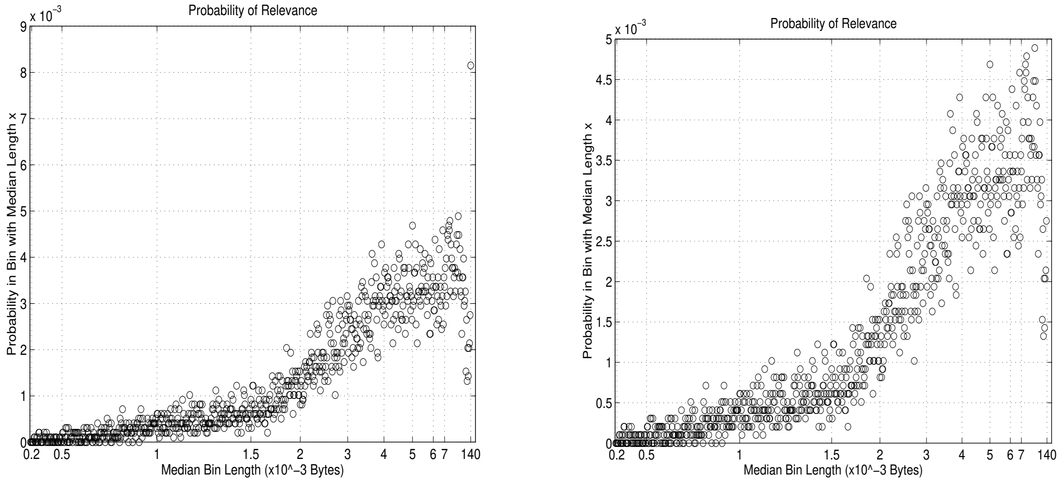
$$P(D \in Bin_i \mid D \ is \ Retrieved)$$

To study the document length dependence of relevance, we repeated the above analysis for the 9,805 ⟨query, relevant-document⟩ pairs for the fifty TREC–3 queries. In terms of conditional probability, given a document D, the probability obtained from this analysis for the ith bin can be represented by

$$P(D \in Bin_i \mid D \ is \ Relevant)$$

If the general assumption that the relevance of a document is independent of its length is true in the TREC collection, the plot obtained from the above analysis of probability of relevance should be mostly free of any biases in favor of documents of any particular length, or “flat”.

Using such document length analysis, we can study the document length retrieval pattern for various term weighting approaches. Comparing these patterns to the pattern for the relevant



(a) (b)

Figure 1: Probability of finding a relevant document in a bin, plotted against the median bin length. Figure (a) shows all the 742 bins; Figure (b) is a zoomed–in view of the majority of the bins. It is evident from this plot that the general assumption of document length independence of relevance is not true in the TREC collection. In TREC, the probability of relevance increases as the documents grow longer. TREC queries 151–200 were used in these experiments.

documents, we can observe how closely the probability of retrieval and the probability of relevance are related for a given term weighting scheme. Schemes for which these two probabilities are closely related should perform better than schemes for which these probabilities are very different.

3.1 Document Length Dependence of Relevance

Figure 1 is the plot obtained from the length analysis of the relevant documents. The most important observation we can make from Figure 1 is that *the general assumption of document length independence of relevance is not valid in the TREC collection.* In TREC, the probability of relevance of a document increases as the documents grow longer. The increase in the probability of relevance of a document with document length suggests that any scheme that biases its retrieval in favor of the longer documents, and does not retrieve documents of all lengths with equal probability, has a better chance of retrieving more relevant documents, and would have a better retrieval effectiveness on the TREC collection. Consider a query that has ten relevant documents, one of length l_1, two of length l_2, three of length l_3, and four of length l_4. If a system retrieves ten documents for that query, the only way it can retrieve all the ten relevant documents is by retrieving exactly one, two, three, and four documents of length l_1, l_2, l_3, and l_4, respectively. In general, a scheme whose probability of retrieval, for the documents of a given length, is very close to the probability of finding a relevant document of that length, should perform better than another scheme which retrieves documents with a very different probability from their relevance probability.

3.2 Retrieval by Smart

The Smart system [19] is a widely used text processing system based on the vector space model [26], developed over the last thirty-five years. In the vector space model, every information item – including the stored texts and any natural language information request – is stored as a set, or vector, of terms. Smart automatically assigns weights to the terms in a vector. A natural language query entered by a user is converted into a weighted term vector, and using the vector inner-product function, a numeric similarity is computed between the query vector and the vector for every document in the collection.

In TREC–3, the Smart group at Cornell used the following term weighting schemes: [5]

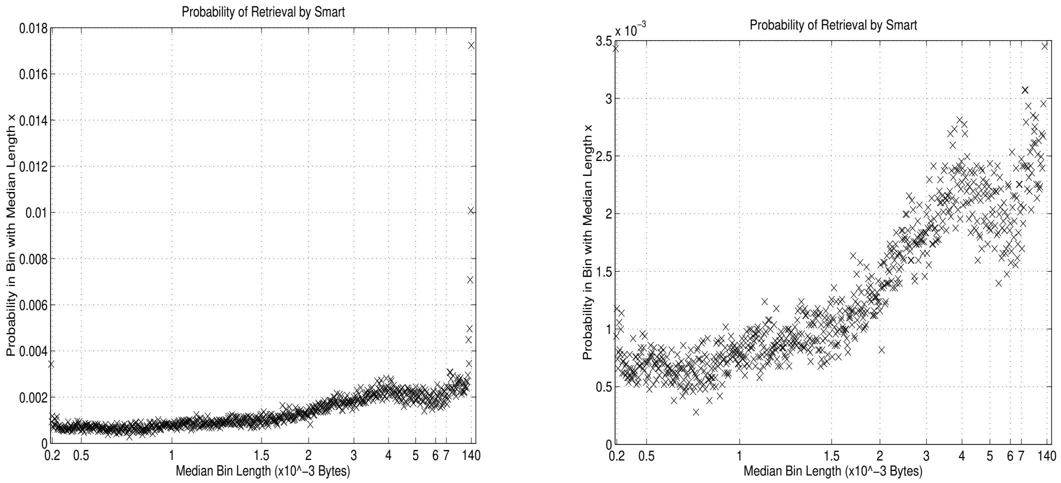
Document term weights (W_{di}):	$\frac{w_{di}}{\sqrt{{w_{d1}}^2+{w_{d2}}^2+\ldots+{w_{dT}}^2}}$,	$w_{di} = 1 + log(tf_i)$
Query term weights (W_{qi}):	$\frac{w_{qi}}{\sqrt{{w_{q1}}^2+{w_{q2}}^2+\ldots+{w_{qT}}^2}}$,	$w_{qi} = (1 + log(tf_i)) \times log(\frac{N}{n})$
Query document similarity:	$\sum_{matching\ terms} W_{dt} \times W_{qt}$	

where,

tf_i	is the term frequency of the ith term in the document/query text,
T	is the number of unique terms in the document/query,
N	is the total number of documents in the collection,
n	is the number of documents within the collection in which the term under consideration is present,
W_{dt}	is the weight of matching term t in the document, and
W_{qt}	is the weight of matching term t in the query.

The expression $\sqrt{{w_1}^2 + {w_2}^2 + \ldots + {w_T}^2}$ is the cosine normalization factor.

To study the document length dependence of the probability of retrieval by the Smart system, for any of the fifty TREC–3 queries, within the top one thousand documents retrieved, we plotted the retrieval probability from a bin against the median length of the bin. Figure 2 is the plot obtained by this length analysis. From Figures 1 and 2 we observe that with increasing document length, the increase in the probability of relevance is steeper than the increase in the probability of retrieval by the Smart system.



(a) (b)

Figure 2: Probability of retrieval from a bin by the Smart system. Figure (a) shows all the 742 bins; Figure (b) is a zoomed–in view of the majority of the bins. TREC queries 151–200 were used in these experiments.

	Original Results	Our Approximation	Our Approximation Compared to the Original Results
	Non-Interpolated Avg. Precision		
Smart	0.2842	0.2842	+ 2.6%
Okapi	0.3370	0.3456	
	+18.6%	+21.6%	

Table 1: Results from our approximation to Okapi's term weighting in the Smart system. The results obtained by our approximation are 2.6% better than the original results reported by Okapi. TREC queries 151–200 were used in these experiments.

3.3 Term Weighting of Okapi and INQUERY

To test the document length retrieval pattern of the term weighting strategies of the Okapi system and the INQUERY system, we approximated their term weighting schemes within the Smart system. We experimentally selected appropriate values for the parameters involved in Okapi's and INQUERY's term weighting schemes to yield good results.

Our approximation to Okapi's term weighting scheme is:

Document term weights (W_d): $\frac{tf \times log(\frac{N-n+0.5}{n+0.5})}{2 \times (0.25 + 0.75 \times \frac{dl}{avdl}) + tf}$

Query term weights (W_q): tf

Query document similarity: $\sum_{matching\ terms} W_{dt} \times W_{qt}$

	Original Results	Our Approximation	Our Approximation Compared to the Original Results
	11-pt. Avg. Precision		
Smart	0.3057	0.3057	
INQUERY	0.3180	0.3330	+ 4.7%
	+ 4.0%	+ 8.9%	
	Non-Interpolated Avg. Precision		
Smart	0.2842	0.2842	
INQUERY	–	0.3118	
		+ 9.7%	

Table 2: Results from our approximation to INQUERY's term weighting in the Smart system. The results obtained by our approximation are 4.7% better than the original results reported by the INQUERY group. TREC queries 151–200 were used in these experiments.

where,

tf is the term frequency of a term in the document/query text,
N is the total number of documents in the collection,
n is the number of documents in the collection in which the term under consideration is present,
dl is the length of the document (in bytes), and
avdl is the average document length in the collection (in bytes).

Table 1 shows that this approximation to Okapi's term weighting scheme is good. It yields results that are 2.6% better than the original results reported by Robertson et al., in [16].

Our approximation to INQUERY's term weighting scheme is:

Document term weights (W_{di}): $0.4 + 0.6 \times (0.4 \times H + 0.6 \times \frac{log(tf+0.5)}{log(max_{tf}+1.0)}) \times \frac{log(\frac{N}{n})}{log(N)}$
Query term weights (W_q): tf
Query document similarity: $\sum_{matching\ terms} W_{dt} \times W_{qt}$

where,

tf is the term frequency of a term in the document/query text,
max_{tf} is the maximum term frequency for any term in that document,
N is the total number of documents in the collection,
n is the number of documents within the collection in which the term under consideration is present, and
H $= 1.0$ if $max_{tf} \leq 25$
$= \frac{25}{max_{tf}}$ otherwise.

Table 2 shows that this approximation to INQUERY's term weighting is also reasonable. In fact, it yields results that are 4.7% better than the original results reported by Broglio et al., in [1]. As TREC evaluations use the non-interpolated precision values, which were not reported for IN-

	Smart	INQUERY	Okapi
Original Results	0.2842	–	0.3370 (+18.6%)
Original Rank	III	II	I
Our Approximation	0.2842	0.3118 (+ 9.7%)	0.3456 (+21.6%)
Our Rank	III	II	I

Table 3: A comparison of the original Okapi and INQUERY results and results obtained from our approximations to the weighting schemes of these systems. Our approximations preserve the ranking of the systems as observed in TREC–3. TREC queries 151–200 were used in these experiments.

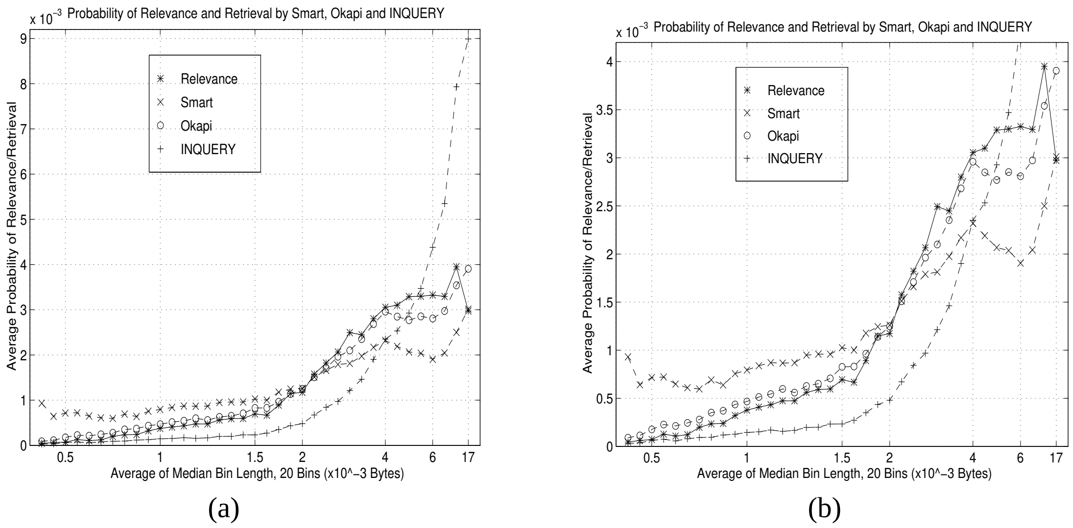
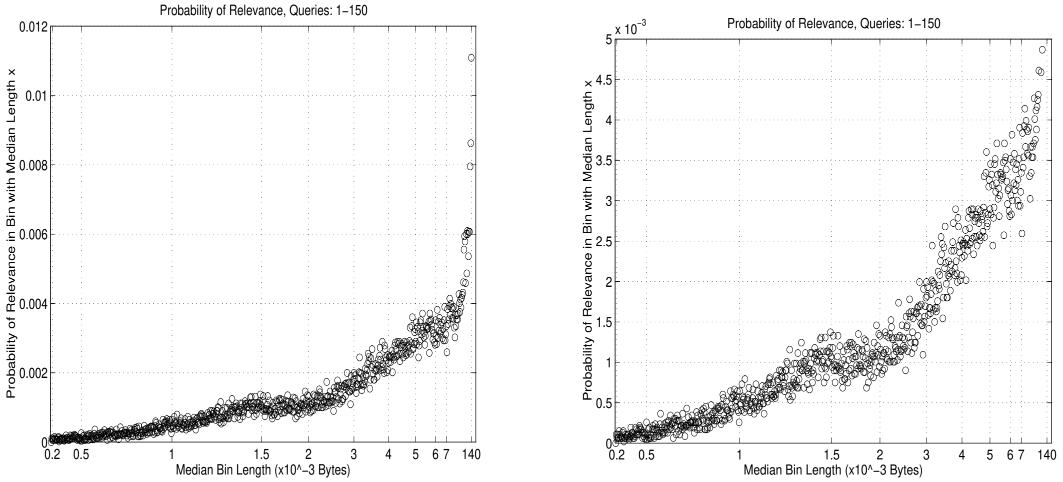


Figure 3: Smoothed plots for the probability of retrieval using the weighting schemes of three systems along with the smoothed plot for relevance. Figure (a) shows all the points; Figure (b) is a zoomed–in view of the majority of the collection. Notice the strong correlation between the probability of relevance and the probability of retrieval using Okapi's term weighting scheme.

QUERY's base run in [1], we have also listed this value for our approximation of INQUERY's weighting scheme in Table 2.

Table 3 compares the results from our approximations to the weighting schemes of the other two systems to the original results reported by Okapi and INQUERY. These results show that our results do follow the TREC ranking of the three systems – Okapi followed by INQUERY followed by Smart. The results from Tables 1, 2, and 3 confirm that the length analysis obtained from these approximations correctly reflects the document length retrieval patterns for the Okapi system and the INQUERY system.

By performing the document length analysis for documents retrieved by our approximation of the weighting schemes of Okapi and INQUERY, we observed that for documents of a certain length, the probability of retrieval by these two systems was closer to the relevance probability, as



(a) (b)

Figure 4: Probability of finding a relevant document in a bin for TREC queries 1–150, plotted against the median bin length. Figure (a) shows all the 742 bins; Figure (b) is a zoomed–in view of the majority of the bins. These queries also show a strong bias of relevance in favor of the longer documents.

compared to Smart. This effect is illustrated in Figure 3, where the smoothed plots[2] for retrieval using weighting schemes of all the three systems are plotted together with the smoothed plot for relevance, obtained from Figure 1. Figure 3 shows that the probability of retrieving a document of a certain length using the weighting scheme of the Okapi system has a very strong correlation with the probability of finding a relevant document of about the same length. We have also seen that the performance of Okapi's term weighting is significantly better than that of the other two weighting schemes. We believe that this strong correlation is the main reason behind the superior retrieval effectiveness of Okapi's term weighting scheme.

4 Training Cosine Normalization

Participants of the TREC–3 conference [11] had access to one hundred and fifty user queries and their relevance judgments, accumulated over the previous TREC conferences [9, 10], to train their systems. Figure 4 shows the document length analysis for the relevant documents for the one hundred and fifty training queries. The relevance judgments for this set of training queries, very much like the ones for the fifty test queries for the TREC–3 ad-hoc task (see Figure 1), exhibit a

[2]We generated smooth plots for various figures by representing a sequence of twenty bins by a single point and connecting these points by a curve. The representative point for a group of twenty bins was obtained by taking the averages of both the median lengths, and the probabilities of retrieval for the twenty consecutive bins.

strong bias in favor of longer documents. If a term weighting scheme is trained to duplicate this document length dependence of relevance in its retrieval, its performance should improve on the training queries, as well as the test queries.

The term weighting function that was used by the Okapi group in their TREC–3 participation was obtained by using four different parameter values, called k_1, k_2, k_3, and b, in their general parameterized term weighting formula BM25. The values 2.0, 0.0, ∞, and 0.75 were used for k_1, k_2, k_3, and b, respectively. [16, 18] This set of numbers was obtained by trying different values using the training queries, and selecting the set that yielded the best results. The term weighting scheme of the INQUERY system involves just one parameter, the H factor. A good H value was learned by the INQUERY group using the 150 training queries, and was used in INQUERY's TREC–3 participation. [1]

The term weighting scheme of the Smart system, on the other hand, does not involve any special parameters. Therefore, in TREC–3, the Smart system did not utilize the previously available training data to adjust its term weighting strategy, beyond the choice of which tf and idf variants were to be used. In this study, we have developed a trainable variant of the cosine normalization function, which we call the *pivoted-cosine normalization* function.

4.1 The Pivoted-Cosine Normalization

Any normalization factor has an effect of decreasing weights of document terms in proportion to the document length. Higher values of the normalization factor, usually used for longer documents, result in lower individual term weights. Lower individual term weights in a document, in turn, lower the query–document similarity, thereby lowering the chances of retrieval of the document. Therefore, the higher the value of the normalization factor for a document is, the lower are the chances of retrieval for that document. In effect, the probability of retrieval of a document is inversely proportional to the normalization factor used in the term weight estimation for that document.

$$P(retrieval) \propto \frac{1}{Normalization\ Factor}$$

This relationship suggests that if we want to boost the chances of retrieval for certain documents, we should lower the value of the normalization factor, and vice-versa.

Our aim is to match the retrieval probability and the relevance probability of documents of a certain length. To do this, we would like to promote the retrieval of the documents that are retrieved

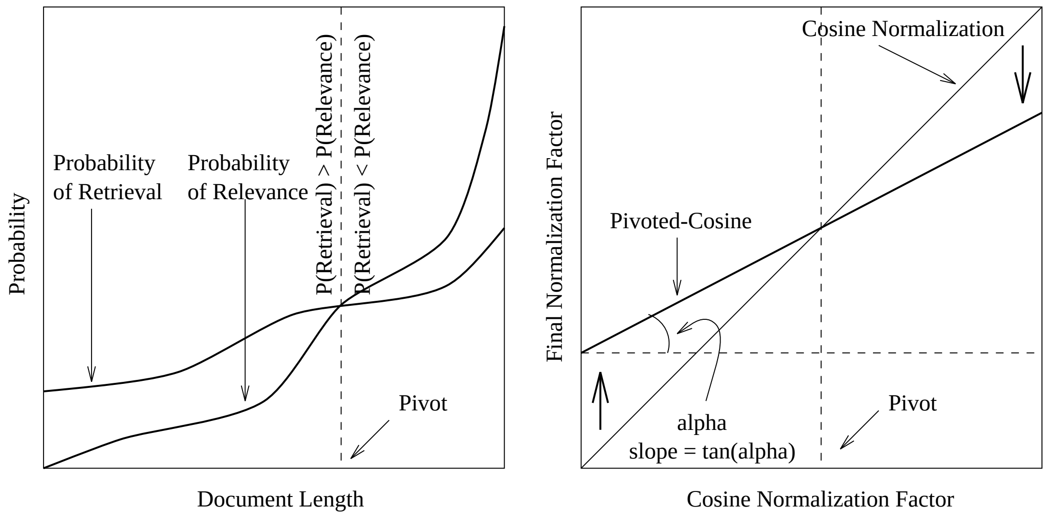
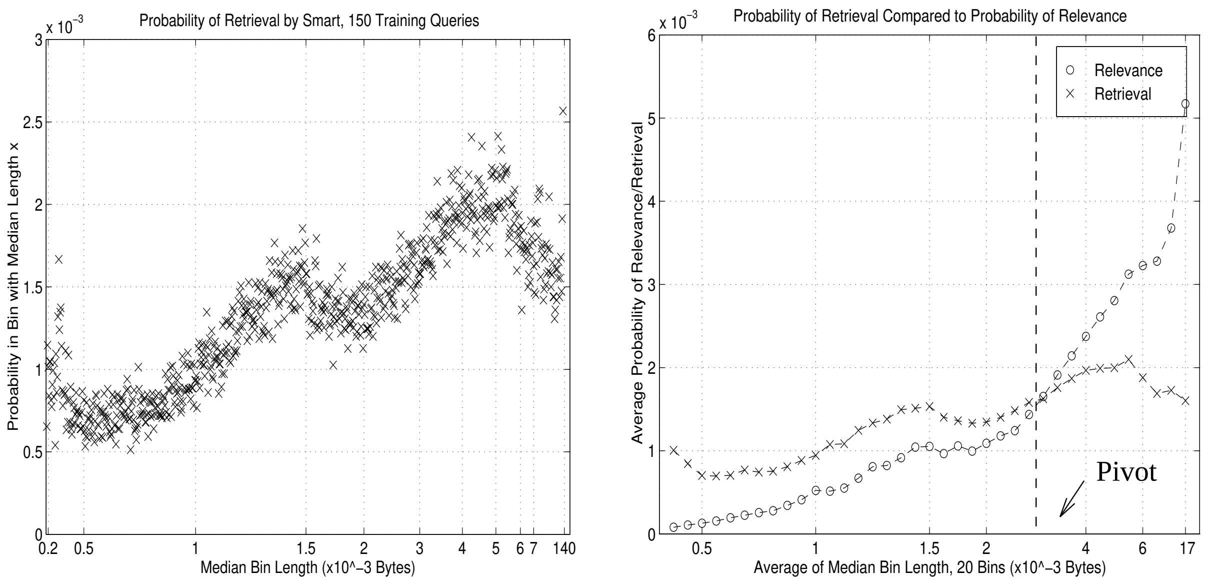


Figure 5: **Pivoted-Cosine Normalization.** Using the training queries, an initial set of documents is retrieved and the probability of relevance and retrieval based on document lengths is computed. The point where these curves intersect is called the pivot. The normalization factor for documents for which $P(retrieval) > P(relevance)$ is increased, whereas the normalization factor for documents for which $P(retrieval) < P(relevance)$ is decreased.

with a lower probability than their probability of relevance, and reduce the chances of retrieval of the documents that are retrieved with a higher probability than their probability of relevance. The pivoted-cosine normalization scheme is based on this principle. Using the training queries, and a cosine normalized weighting scheme, a set of documents is initially retrieved. The length of the retrieved documents is compared to that of the relevant documents. The normalization factor for the document length range in which documents were retrieved with a higher probability than their probability of relevance is increased, thereby reducing the chances of their retrieval; whereas the normalization factor for the document length range in which documents were retrieved with a lower probability than their probability of relevance is decreased, thereby increasing the chances of their retrieval.

This process is illustrated in Figure 5. The retrieval and the relevance curves for the training set are plotted. The point where these two curves cross each other is called the *pivot*. The documents on one side of the pivot are generally retrieved with a higher probability than their relevance probability, whereas the documents on the other side of the pivot are retrieved with a lower probability than their probability of relevance. The normalization function can now be "pivoted" around the pivot and "tilted" so as to increase the value of the normalization factor, as compared to the original normalization factor, on one side of the pivot. This also decreases the value of the normalization factor on the other side of the pivot. The amount of "tilting" needed



(a) (b)

Figure 6: Initial estimation of the pivot using the 150 training queries. Figure (a) shows the probability of retrieval for different bins, using cosine normalization. Figure (b) shows the smoothed plots for the probability of relevance and the retrieval probability. It can be observed from the smoothed plot that the pivot lies around 3,000 bytes.

becomes a parameter of the weighting scheme, and is called the *slope*. Also, the location of the pivot can be fine tuned to yield good performance.

4.2 Training for TREC–3

Figure 6 shows the initial pivot estimation using the one hundred and fifty training queries[3]. Figure 6 indicates that the pivot for the training queries lies around the document length of 3,000 bytes. In a previous study, we sampled several documents of different lengths from the TREC collection and computed the cosine normalization factors for the documents. [28] It was found that the cosine normalization factor for documents which were approximately the same length as the pivot was in the range of sixteen to eighteen. Using these values for the pivot, we trained for a good value of the slope. Some retrospective results obtained during this training session are shown in Table 4.

We observed in our experiments that, when used with cosine normalization in document term

[3] For this experiment, we removed some fields from the training queries that are not present in the testing queries, specially the *concepts* field.

Cosine Normalization 23,664 0.2280	Pivoted-Cosine Normalization						
	Slope						
Pivot	0.40	0.50	0.60	0.65	0.70	0.80	0.90
16	24,889 0.2581 +13.2%	25,882 0.2722 +19.4%	26,285 0.2767 +21.3%	26,277 0.2760 +21.0%	26,133 0.2731 +19.8%	25,524 0.2619 +14.8%	24,384 0.2432 + 6.3%
17	24,690 0.2552 +11.9%	25,779 0.2707 +18.7%	26,245 0.2764 +21.2%	26,306 0.2764 +21.2%	26,181 0.2740 +20.2%	25,612 0.2633 +15.5%	24,458 0.2436 + 6.8%
18	24,473 0.2532 +10.6%	25,642 0.2689 +17.9%	26,188 0.2761 +21.1%	26,303 0.2766 +21.3%	26,210 0.2749 +20.5%	25,703 0.2646 +16.0%	24,522 0.2448 + 7.4%

Table 4: Retrospective results of using pivoted-cosine normalization with the 150 training queries. Each entry indicates the total number of relevant documents retrieved for all the 150 queries, the non-interpolated average precision, and the percentage improvement over using full cosine normalization.

weights, a term frequency factor of 1+log(tf) in the query term weights yields significant improvements over using pure tf; but with pivoted-cosine normalization in document term weights, using 1+log(tf) is slightly worse than using raw tf in the query term weights. It is possible that using the logarithmic factor on the query term frequencies is partially having the same effect as pivoted-cosine normalization. The baseline average precision (0.2280) for cosine normalized documents in Table 4 was obtained using 1+log(tf) in the query term weights, whereas we used pure tf in the query term weights with pivoted-cosine normalized documents.

To verify that the pivoted-cosine normalization scheme is working well, *i.e.*, it is changing the retrieval characteristics for documents of different lengths to match the relevance characteristics for documents of different lengths, we performed the document length analysis on the documents retrieved using the pivoted-cosine normalization scheme, and compared it to the document length analysis of the relevant documents for the 150 training queries. Figure 7 shows the smoothed plots for relevance and retrieval using pivoted-cosine normalization. It is evident from Figure 7 that the use of pivoted-cosine normalization enables us to retrieve documents of a given length with about the same probability as their probability of relevance. Also the improvements obtained in Table 4 strongly support our hypothesis that if a retrieval technique retrieves documents of a given length in accordance with their probability of relevance, it would perform substantially better.

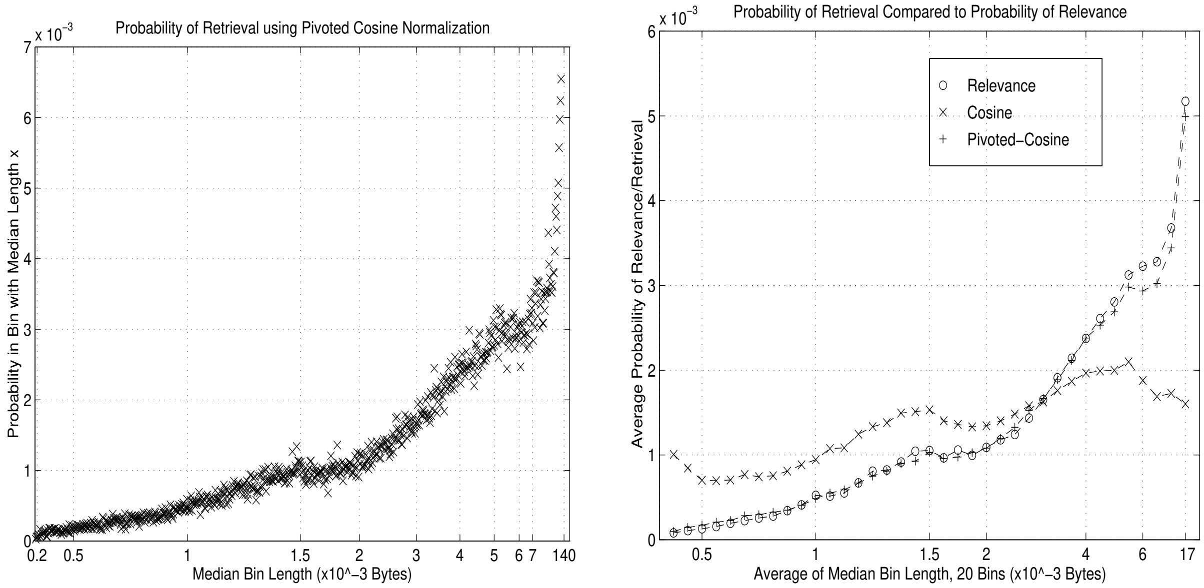


Figure 7: Document length analysis for documents retrieved using pivoted-cosine normalization for the 150 training queries. A pivot value of 17 and a slope value of 0.65 was used in these experiments. It is clear from the smoothed retrieval and relevance plots that pivoted-cosine normalization enables us to retrieve documents in accordance with their probability of relevance.

5 Results

We tested the new normalization scheme on the fifty test queries used in TREC–3. In selecting a value for the pivot and the slope from Table 4, whenever two sets of ⟨pivot, slope⟩ values had negligible difference in average precision, we favored the set that retrieved more relevant documents. For example, the sets ⟨16, 0.60⟩ and ⟨17, 0.65⟩ yield an average precision of 0.2767 and 0.2764, respectively, but the set ⟨17, 0.65⟩ retrieves more relevant documents than the set ⟨16, 0.60⟩ (26,306 in place of 26,285) and was preferred over the latter.

We selected the ⟨pivot, slope⟩ set ⟨17, 0.65⟩ and used pivoted-cosine normalization to perform retrieval using the fifty test queries of TREC–3. A comparison between the use of cosine normalization and pivoted-cosine normalization is listed in Table 5. Table 5 shows that the use of pivoted-cosine normalization yields a significant 13.7% improvement over using cosine normalization for the fifty TREC–3 queries. The recall-precision graph corresponding to Table 5 is shown in Figure 8. This graph illustrates that the pivoted-cosine normalization performs better than cosine normalization at every recall point.

We compared the performance of the Smart system using pivoted-cosine normalization to the

Run 1. Cosine Normalization Run 2. Pivoted-Cosine Normalization		
Run	1	2
# Queries	50	50
Total number of documents over all queries		
Retrieved	50000	50000
Relevant	9805	9805
Rel. Ret.	6531	6679
Interpolated Recall - Precision Averages		
at 0.00	0.7884	0.8565
at 0.10	0.5697	0.6354
at 0.20	0.4786	0.5290
at 0.30	0.4137	0.4565
at 0.40	0.3492	0.3854
at 0.50	0.2859	0.3150
at 0.60	0.2196	0.2526
at 0.70	0.1478	0.1805
at 0.80	0.0848	0.1224
at 0.90	0.0245	0.0339
at 1.00	0.0000	0.0000
Avg. precision (non-interpolated)		
	0.2842	0.3233 (+13.7%)

Table 5: Performance of pivoted-cosine normalization compared to that of cosine normalization on fifty TREC–3 queries. The pivot and the slope values are 17 and 0.65, respectively, and are learned using the 150 training queries available for TREC–3.

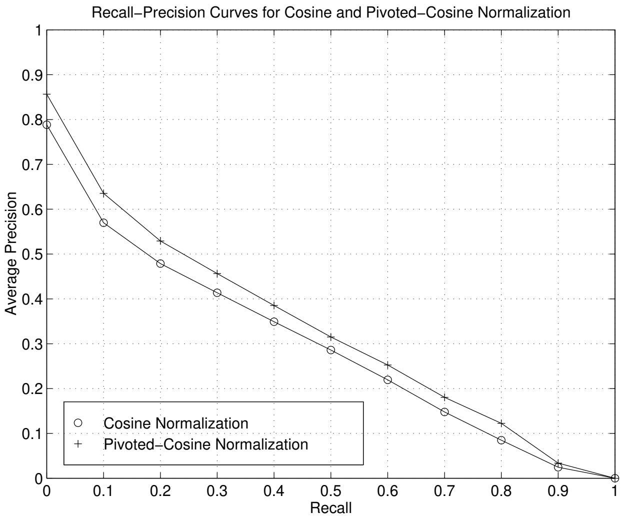


Figure 8: Recall-precision graph for pivoted-cosine normalization compared to cosine normalization. It is evident that the use of pivoted-cosine normalization is better than using cosine normalization at every recall point.

	Cosine Normalization	Pivoted-Cosine Normalization
	Non-Interpolated Average Precision	
Smart	0.2842	0.3233
Okapi	0.3370	0.3370
	+18.6%	+4.2%
	11-pt. Average Precision	
Smart	0.3057	0.3425
INQUERY	0.3180	0.3180
	+ 4.0%	- 7.2%

Table 6: A comparison of the results obtained using pivoted-cosine normalization to the TREC–3 results reported by Okapi and INQUERY. Using pivoted-cosine normalization makes the retrieval effectiveness of the term weighting scheme of the INQUERY system, which was initially 4.0% better than that of the Smart system, 7.2% worse. It also reduces the performance gap between Smart and Okapi from 18.6% to 4.2%.

results reported by Okapi and INQUERY in their TREC–3 participation. Table 6 shows that when Smart's term weighting scheme is trained to incorporate document length biases in relevance, its retrieval effectiveness improves substantially, and is now at par with these two systems (a bit better than INQUERY, and slightly worse than Okapi).

These results show that the earlier performance differences between the Smart system and the other two systems may not be due to any inherent virtue of probabilistic term weighting used by Okapi and INQUERY. Instead, these systems may have used the training queries to incorporate the document length pattern of relevance in their retrieval strategies, which resulted in superior performance at TREC–3. When the Smart system is trained to incorporate similar biases in its retrieval strategy, its performance becomes comparable with both the systems.

6 Discussion

It is plausible that very small documents containing very few words have too little information to be useful to a user. A text should have some minimum length to be informative. This hypothesis is strongly supported by the study done by Callan on passages. [7] In Table 7 of [7], one notices that the average precision of a passage-based search increases as the size of passages used in the search increases from twenty five words to around three hundred words, suggesting that the twenty five or fifty word passages are not as informative as the two or three hundred word passages. Also, in Kwok et al. large documents are segmented into chunks of five hundred and fifty words each to obtain good performance from their system in the TREC collection. [13] Very small chunks did not work as well as the larger chunks. These results support our hypothesis that very small texts are not as informative as longer texts and could, therefore, be less useful to a user. The minimum amount of information needed to satisfy an information need can be termed as an "information unit". To be useful, a text should contain at least this much information.

Several researchers have pointed out that long documents consist of several information units (called *segments*, or *themes*, or *topics*, or *text–tiles*). [21, 25, 12] Since a single information unit can potentially satisfy a query, the chances of a document being useful to a user-query increase as the number of information units present in a document increase. This can explain the higher probability of relevance that we observe for the longer documents in the TREC collection. If a document addresses several topics, or it addresses several aspects of the same topic, it is potentially useful for a large set of queries, as opposed to a small document that deals with just one topic and can be potentially useful to only the queries aimed at that particular topic. These observations suggest that long documents might be judged more useful by users in other full-text collections as well. If this hypothesis is true, it would be possible to use the pivoted-cosine normalization in retrieval for other text collections, without any training. We are currently studying the stability of the pivot and the slope values across collections for the pivoted-cosine normalization scheme.

7 Conclusions

This study shows that the expectation of mutual independence of relevance and document lengths is largely violated in the TREC collection. In the TREC collection, long documents are more likely to be found useful by a user. A retrieval system that retrieves documents of a certain length with the same chances as the chance of finding a relevant document of that length, would perform better than another system which retrieves documents with probabilities different from their probability of relevance. This bias in relevance was a major reason behind the discrepancy in the performances of the Smart system and the Okapi and the INQUERY systems at TREC–3. If the Smart system is biased to retrieve more longer documents, its performance becomes comparable to the best system at TREC–3.

References

[1] J. Broglio, J.P. Callan, W.B. Croft, and D.W. Nachbar. Document retrieval and routing using the INQUERY system. In D. K. Harman, editor, *Proceedings of the Third Text REtrieval*

Conference (TREC-3), pages 29–38. NIST Special Publication 500-225, April 1995.

[2] Chris Buckley. Implementation of the SMART information retrieval system. Technical Report TR85-686, Department of Computer Science, Cornell University, Ithaca, NY 14853, May 1985.

[3] Chris Buckley. The importance of proper weighting methods. In M. Bates, editor, *Human Language Technology*. Morgan Kaufman, 1993.

[4] Chris Buckley, James Allan, and Gerard Salton. Automatic routing and ad-hoc retrieval using SMART : TREC 2. In D. K. Harman, editor, *Proceedings of the Second Text REtrieval Conference (TREC-2)*, pages 45–56. NIST Special Publication 500-215, March 1994.

[5] Chris Buckley, James Allan, Gerard Salton, and Amit Singhal. Automatic query expansion using SMART : TREC 3. In D. K. Harman, editor, *Proceedings of the Third Text REtrieval Conference (TREC-3)*, pages 69–80. NIST Special Publication 500-225, April 1995.

[6] Chris Buckley, Gerard Salton, and James Allan. Automatic retrieval with locality information using SMART. In D. K. Harman, editor, *Proceedings of the First Text REtrieval Conference (TREC-1)*, pages 59–72. NIST Special Publication 500-207, March 1993.

[7] J.P. Callan. Passage–level evidence in document retrieval. In W. Bruce Croft and C.J. van Rijsbergen, editors, *Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 302–310. Springer-Verlag, New York, July 1994.

[8] J.P. Callan, W.B. Croft, and S.M. Harding. The INQUERY retrieval system. In *Proceedings of the Third International Conference on Database and Expert Systems Applications*, pages 78–83. Springer-Verlag, 1992.

[9] D. K. Harman. Overview of the first Text REtrieval Conference (TREC-1) . In D. K. Harman, editor, *Proceedings of the First Text REtrieval Conference (TREC-1)*, pages 1–20. NIST Special Publication 500-207, March 1993.

[10] D. K. Harman. Overview of the second Text REtrieval Conference (TREC-2). In D. K. Harman, editor, *Proceedings of the Second Text REtrieval Conference (TREC-2)*, pages 1–20. NIST Special Publication 500-215, March 1994.

[11] D. K. Harman. Overview of the third Text REtrieval Conference (TREC-3). In D. K. Harman, editor, *Proceedings of the Third Text REtrieval Conference (TREC-3)*, pages 1–19. NIST Special Publication 500-225, April 1995.

[12] Marti A. Hearst and Christian Plaunt. Subtopic structuring for full-length document access. In *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 59–68, June 1993.

[13] K.L. Kwok, L. Grunfeld, and D.D. Lewis. TREC-3 ad-hoc, routing retrieval and thresholding experiments using PIRCS. In D. K. Harman, editor, *Proceedings of the Third Text REtrieval Conference (TREC-3)*, pages 247–255. NIST Special Publication 500-225, April 1995.

[14] S.E. Robertson et al. Okapi at TREC. In D. K. Harman, editor, *Proceedings of the First Text REtrieval Conference (TREC-1)*, pages 21–30. NIST Special Publication 500-207, March 1993.

[15] S.E. Robertson et al. Okapi at TREC–2. In D. K. Harman, editor, *Proceedings of the Second Text REtrieval Conference (TREC-2)*, pages 21–34. NIST Special Publication 500-215, March 1994.

[16] S.E. Robertson et al. Okapi at TREC–3. In D. K. Harman, editor, *Proceedings of the Third Text REtrieval Conference (TREC-3)*, pages 109–126. NIST Special Publication 500-225, April 1995.

[17] S.E. Robertson and K. Sparck Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3):129–146, May-June 1976.

[18] S.E. Robertson and S. Walker. Some simple effective approximations to the 2–poisson model for probabilistic weighted retrieval. In W. Bruce Croft and C.J. van Rijsbergen, editors, *Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 232–241. Springer-Verlag, New York, July 1994.

[19] Gerard Salton, editor. *The SMART Retrieval System—Experiments in Automatic Document Retrieval.* Prentice Hall Inc., Englewood Cliffs, NJ, 1971.

[20] Gerard Salton. *Automatic text processing—the transformation, analysis and retrieval of information by computer.* Addison-Wesley Publishing Co., Reading, MA, 1989.

[21] Gerard Salton and James Allan. Automatic text decomposition and structuring. In *RIAO 94 Conference Proceedings*, pages 6–20, October 1994.

[22] Gerard Salton and Chris Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[23] Gerard Salton and Chris Buckley. A note on term weighting and text matching. Technical Report 90-1166, Department of Computer Science, Cornell University, Ithaca, NY 14853, October 1990.

[24] Gerard Salton and M.J. McGill. *Introduction to Modern Information Retrieval.* McGraw Hill Book Co., New York, 1983.

[25] Gerard Salton and Amit Singhal. Automatic text theme generation and the analysis of text structure. Technical Report 94-1438, Department of Computer Science, Cornell University, Ithaca, NY 14853, July 1994.

[26] Gerard Salton, A. Wong, and C.S. Yang. A vector space model for information retrieval. *Journal of the American Society for Information Science*, 18(11):613–620, November 1975.

[27] Gerard 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, January–February 1975.

[28] Amit Singhal, Gerard Salton, and Chris Buckley. Length normalization in degraded text collections. Technical Report TR95-1507, Department of Computer Science, Cornell University, Ithaca, NY 14853, April 1995.

[29] Howard Turtle. *Inference Networks for Document Retrieval.* PhD thesis, Department of Computer Science, University of Massachusetts, Amherst, MA 01003, 1990. Available as COINS Technical Report 90-92.