# MINING INDEX TERMS

A thesis submitted in partial fulfillment of
the requirements for the degree of

MASTER OF TECHNOLOGY

by

**KOTIKALAPUDI S V D PRASAD**
**(Roll No. 10410104)**

Under the guidance of
**Dr.SANASAM RANBIR SINGH**


DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY–GUWAHATI
2013

# Abstract

In general, traditional Web Search Engines index entire contents of a document, except possibly few stop words. The number of terms present in the document collection is huge and only a very small subset of the entire vocabulary set is used in query formation. For majority of the documents,only a very small set of terms are actually used by users to retrieve the document. Weighting document or request terms so that some are defined as more important than others is an appealing idea. Various approaches to weighting document terms have been investigated, and some experiments have shown it to be useful. These weighting schemes are designed to influence matching in retrieval. Document term weights have also been taken into account in other contexts, for example in the formation of term classifications. It concentrates on the main use of weights, for controlling searching. The units of any form of index description may be weighted.But term weighting covers a variety of schemes based on quite different principles. Thus the assertion that weighting is useful in retrieval when it refers to projects operating different schemes, is of limited value. The objective of this thesis is to examine the logic of term weighting, and to see what effects different forms of weighting have on retrieval. Weights are sometimes derived from human judgements about the importance of terms, but weights based on statistical information are intended to simulate these and have the advantage of being objective. This thesis study and investigate two indexing frameworks and compare their performance with BM25 based baseline system. It is evident from several experiments that we can reduce index size by indexing only representative terms, without compromising retrieval performance.

# Contents

**Abstract** **i**

**List of Tables** **iv**

**1 Introduction** **1**

1.1 Introduction . . . . . 1

1.2 Motivation of the Thesis Work . . . . . 2

1.3 Problem Definition . . . . . 2

1.4 Thesis Organization . . . . . 3

**2 Traditional Indexing Methods** **4**

2.1 TF-IDF . . . . . 4

2.2 BM25 . . . . . 6

**3 Proposed Framework and Experimental Setup** **8**

3.1 Baseline Index . . . . . 8

3.2 Highest Index Terms (HIT) . . . . . 9

3.3 Term with the highest document's rank (HDR) . . . . . 9

3.4 Queries from Query log . . . . . 10

**4 Experiments and Results** **11**

4.1 Characteristics of Query Log . . . . . 11

4.2 Characteristics of Dataset . . . . . 14

4.3 Evaluation and Discussion . . . . . 14

4.3.1 Comparison of index sizes . . . . . 14

4.3.2 Retrieval time for queries . . . . . 15

4.3.3 Retrieval quality . . . . . 15

**5 Conclusions** **17**

5.1 Future Works . . . . . 17

# List of Tables

4.1 Queries used in Evaluation from Query Log . . . . . 13
4.2 Comparison of Index sizes of Baseline and BM25(in Giga Bytes) . . . . . 14
4.3 Comparison of Index Sizes of Baseline and HDR(in Giga Bytes) . . . . . 14
4.4 Comparison of Retrieval times(for 1000 queries in seconds) . . . . . 15
4.5 Comparison of average positional ranking of relevant documents for a query . 15
4.6 Comparison of average recall at top 20 for listed queries . . . . . 16

# Chapter 1

# Introduction

## 1.1 Introduction

An Information Retrieval (IR) system is a system that indexes a collection of documents such as Web collection and returns supposedly relevant documents against a query submitted by a user [5]. Present, traditional (and commercial) IR systems index entire content of the documents (excluding html tags and stop words). The primary reason of indexing the entire content of the document is to retain complete information of the documents.Text documents, HTML documents in particular, are semi-structure in nature containing free form texts. Though there are large number of terms(words) in a document, only a few of them are representative of the document. Some of the words are never used by the users in their query formation or never likely to use them. Our preliminary investigation using AOL query log [6] shows that only about 3.72% percent of the indexed terms are actually used by the users in query formation. It indicates that only a very few of the Web contents are actually contributing in retrieving the documents. Indexing entire content of the document has the following issues:

- **Index Space**: Since only few indexed terms are used in query formation, a large part of the index (memory space) is not effectively used. It results in huge waste in memory space i.e., waste in storage cost.

- **Retrieval quality**: The quality of retrieved documents are often affected by the noisy content of document. Though a term is not representative term of document, the document might be retrieved by system for a query containing the term. It will result poor quality of the retrieved results. If we can identify such non-representative terms and index only

the representative term, we can improve the retrieval quality.

- **Retrieval Time**:It is a measure of time elapsed between the submitting of query and getting results. The retrieval time need to be minimized for efficient performance.

Therefore, it is important to investigate feasibility of reducing index size without compromising retrieval performance. This is the main objective of this thesis. In this thesis, we propose to investigate several frameworks to identify potential index terms and further investigate their retrieval performance.

## 1.2 Motivation of the Thesis Work

A studying using AOL click-through data shows only 3% of the terms present in a document are used by users to retrieve the same document. It means that almost 97% of the terms are not useful for retrieving purposed and hence not necessary to index [6]. In this thesis, we propose to determine indexing term that are likely to be good representative of the documents. The motivation is to identify such representative terms and index only such terms to retrieve the documents without compromising the retrieval performance.

## 1.3 Problem Definition

In general, traditional Web Search Engines index entire contents of a document, except possibly few stop words. The number of terms present in the document collect is huge and only a very small subset of the entire vocabulary set is used in query formation. For majority of the documents,only a very small set of terms are actually used by users to retrieve the document. In this work we implement frameworks HIT and HDR,to indentify the representative terms for each document in document collection. This thesis study and investigate two indexing framework and compare their performance with TF-IDF based baseline system. It is evident from several experiments that we can reduce index size by indexing only representative terms, without compromising retrieval performance.

## 1.4 Thesis Organization

The outline of the work is as follows: Chapter 2 presents the traditinal indexing methods in Informational Retrieval. Chapter 3 explains our proposed framework of HDT and HDR, in detail and explian how we obtain queries for the evaluation of proposed mechanisms. Chapter 4 presents experiments performed on data set and the results. Finally chapter 5 concludes the work and proposes the scope for future work.

# Chapter 2

# Traditional Indexing Methods

Term weighting is an important aspect of modern text retrieval systems. 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 each term. Three main components that affect the importance of a term in a text are the term frequency factor(tf), the inverse document frequency factor(idf), and document length normalization [2]. This section presents in detail about the traditional indexing methods used in Information Retrieval.

## 2.1 TF-IDF

A crucial issue underlying an IR system is to rank the returned documents by decreasing order of relevance [4]. For example,recent surveys on the Web show that users rarely look beyond the top returned documents. Usually, ranking is based on a weighting model. Almost all weighting models take the within document term frequency(tf),the number of occurrences of the given query term in the given document,into consideration as a basic factor for weighting documents. For example, the classical tf-idf weighting formula is the following:

$$w(t,d) = tf * log(\frac{N}{df})$$

where w(t, d) is the weight of document d for term t, N is the number of documents in the collection and df is the document frequency, which is the number of documents containing the term t.

The above tf-idf formula is based on two basic principles of weighting:

- For a given term, the higher its frequency in the collection the less likely it is that it reflects much content.
- For a given term in a given document, the higher the within document term frequency(tf) is, the more information the term carries within the document.

However, the term frequency is dependent on the document length, i.e. the number of tokens in a document, and needs to be normalized by using a technique called term frequency normalization. Term weighting is an important aspect of modern text retrieval systems.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. Three main components that affect the importance of a term in a text are the term frequency factor(tf), the inverse document frequency(idf) and document length normalization. Document length normalization of term weight is used to remove the advantage that the long documents have in retrieval over the short documents. In Pivoted Document Length Normalization, Singhal et. al. gave the following two reasons for the need of the tf normalization:

- Higher term frequencies: Long documents usually use the same terms repeatedly. As a result, the term frequency factors may be large for long documents, increasing the average contribution of its terms towards the query-document similarity.
- More terms: Long documents also have numerous different terms.This increases the number of matches between a query and a long document, increasing the query document similarity, and the chances of retrieval of long documents in preference over shorter documents.

The two reasons above are based on the observation of term occurrences in the documents. As a consequence, a weighting model without employing a normalization method, such as tf-idf , could produce biased weights with respect to the document length, favouring long documents. Document length normalization is a way of penalizing the term weights for a document in accordance with its length. Various normalization techniques are used in information retrieval systems.

## 2.2 BM25

In this section, we propose a collection independent method for automatically tuning the parameters of a term frequency normalization approach. Our approach is based on the notion of normalization effect. A classical method for tuning the term frequency normalization parameters is the pivoted normalization [2]. In 1996,Singhal et. al. studied a set of collections and proposed a heuristic normalization method by pivoting the normalization factor to fit the relevance judgement information:

$$(1.0 - slope) + slope * \frac{factor}{factor_{avg}} \tag{2.2.1}$$

where factor and $\text{factor}_{\text{avg}}$ are the normalization factor and the average normalization factor respectively. The normalization factor is obtained by using a specific normalization method, e.g. the Cosine normalization where slope is a parameter. They claimed that slope = 0.2 is effective across collections.

As one of the most well established IR systems, Okapis normalization method is similar to the pivoted normalization. In the Okapi BM25 document weighting function, the idf factor $w^{(1)}$ is normalized as follows:

$$w = w^{(1)} \frac{(k_1 + 1)tf}{K + tf} \frac{(k_3 + 1)qtf}{k_3 + qtf} \tag{2.2.2}$$

where w is the final weight and K is:

$$k_1 * ((1 - b) + \frac{l}{b * avgl})$$

l and avgl are the document length and the average document length in the collection respectively; k1 , k3 and b are parameters; qtf is the number of occurrences of a given term in the query; tf is the within document frequency of the given term.

Let both the numerator and the denominator of

$$\frac{(k_1 + 1)tf}{K + tf}$$

be divided by tf , the equation (2.0.1) and (2.0.2) becomes:

$$w = w^1 \frac{k_1 + 1}{\frac{k_1((1-b)+b\frac{l}{avgl})}{tf} + 1} \frac{(k_3 + 1)qtf}{k_3 + qtf}$$

Let

$$tfn = \frac{tf}{k_1((1 - b) + b\frac{l}{avgl})}$$

$$w = w^1 \frac{k_1 + 1}{\frac{1}{tfn} + 1} \frac{(k_3 + 1)qtf}{k_3 + qtf}$$
$$= \frac{(k_1 + 1)tfn}{tfn + 1} \frac{(k_3 + 1)qtf}{k_3 + qtf}$$

Hence the term frequency normalization component of the BM25 formula can be seen as:

$$tfn = \frac{tf}{k_1((1 - b) + b\frac{l}{avgl})}$$

This is quite similar to the pivoted normalization. Both formulas assume an interpolated form of function for the factor. The default setting of the Okapi system is k1 = 1.2,k3 = 1000 and b = 0.75. The setting was obtained by taking into account the relevance judgement in experiments on a merged data set of TREC collections.

# Chapter 3

# Proposed Framework and Experimental Setup

This section presents a detail discussion about the proposed framework named HIT:Highest Index Terms and HDR:Highest Document Rank.The design and implementation details are discussed in details. As mentioned in Chapter 1, we identify the representative terms in a document by using BM25 weighting function in HIT and terms having the highest document rank in HDR. Query log contains the user-clicked URL's for each query. We compare the performance of proposed frameworks to BM25 based Baseline indexer by using queries from query log.

## 3.1 Baseline Index

Okapi BM25 weighting mechanism is used for ranking the documents in the collection. For the purpose of ranking the documents matching this query, we are really interested in the relative(rather than absolute) scores of the documents in the collection. To this end, it suffices to compute the BM25 score from each document D to query Q. For all the postings in the inverted index for the terms in q, accumulating the total score for each document. This scheme computes a score for every document in the postings of any of the query terms; the total number of such documents may be considerably smaller than N.Given these scores, the final step before presenting results to a user is to pick out the K highest-scoring documents.We use our Web sample as a documents collection and index it in a search engine.

**Algorithm 1** HIT

1: **for all** documents $d$ in collection **do**
2: **for all** terms $t \in d$ **do**
3: Determine $rel(t, d)$
4: **end for**
5: Rank the terms based on their $rel(t, d)$ weight
6: Select top $k$ terms as representative of $d$
7: Index the representative terms
8: **end for**

## 3.2 Highest Index Terms (HIT)

For every term $t \in d$, we estimate the relevance weight of the term $t$ with document $d$ using an estimator such as Okapi-BM25 or TF-IDF. We denote relevance score by $rel(t, d)$. We then sort the terms in descending order of their relevance weights and select the top terms as representative terms of the document. Formal procedure shown in Algorithm 1.

## 3.3 Term with the highest document's rank (HDR)

Given a document term $t$, let $rank(t, d)$ be the rank of the document $d$ against query $t$. For instance, if $d$ is ranked at the top for query $t$, then $rank(t, d)$ is 1, if it is ranked second, then

**Algorithm 2** HDR

1: **for all** documents $d$ in collection **do**
2: **for all** terms $t \in d$ **do**
3: Determine $rank(t, d)$
4: Find normalize rank $Nrank(t, d)$
5: **end for**
6: Sort the terms based on their $Nrank(t, d)$
7: Select top $k$ terms as representative of $d$
8: Index the representative terms
9: **end for**

$rank(t, d)$ is 2 and so on. Now, we normalize rank of a document $d$ against a query $t$ as follows:

$$Nrank(t, d) = \frac{rank(t, d)}{|R_t|} \tag{3.3.1}$$

where $|R_t|$ is the number of documents found for query $t$. For all the terms $t \in d$, we sort terms in ascending order of their $Nrank(t, d)$ (document at the top has least $rank(t, d)$ weight) and select top terms as representative terms. Formal procedure shown in Algorithm 2.

## 3.4 Queries from Query log

The objective of this thesis is to examine the logic of term weighting, and to see what effects different forms of weighting have on retrieval. Weights are sometimes derived from human judgements about the importance of terms, but weights based on statistical information are intended to simulate these and have the advantage of being objective. Query logs store users activities including clickthrough i.e., query, click URL and rank of the clicked results etc. Exploring query logs, we can easily determine users search vocabulary. Query logs are used in many applications such as indent mining, query expansion, document clustering etc. We can use query log to determine index terms. The queries used in evaluation are listed in Table 4.1.

# Chapter 4

# Experiments and Results

The main goal of the experiments is to show that our method reduces index space without compromising retrieval quality. In Chapter 3,we discuss the algorithms of proposed framework. In this section, we present experimental results and queries used in the evaluation in performance. We also discuss the retrieval performance of the proposed frameworks. The main focus of analysis present in this chapter is retrieval quality,index space and retrieval time with respect to the queries submitted.

## 4.1 Characteristics of Query Log

In this study, a large query logs collected from AOL search engine which consists of approximately 20 millions of queries submitted by 650000 users was used to verify the performance of proposed mechanisms. The query log was surrounded by controversies when it was first published to public because of its privacy concerns [7]. There were some debates about the ethics of using this data [9, 8]. According to Anderson [9], using AOL query log for research cannot be considered unethical as long as the aim is not to determine the identity of actual user. This claim had been well accepted by research communities and this data had been used in several studies in recent past including [10, 11]. We also declare that the analysis described in this paper is strictly anonymous; data was never used to identify any identity. The results reported in this paper rely strictly on the aggregated statistics.

We use query log to generate the testing data. To evaluate the proposed framework, we need a set of queries and their relevant URLs. In general, such datasets are created manually. One problem with this approach is that, such dataset could be biased towards the labeller. To

avoid biasness, we use AOL query log. We identify the most frequently submitted query and the URLs that users click against these query. However, we have reported the peformance for the top 128 queries only in this report.The queries are listed in Table 4.1. In IR research, it is well accepted that majority of the users click on relevant URLs.

Table 4.1: Queries used in Evaluation from Query Log

| | | | |
|---|---|---|---|
| adam4adam(4) | adultfriendfinder.com(7) | airtran(7) | 89.com(4) |
| abc.com(2) | amazon.com(7) | americanidol.com(4) | americanidol(3) |
| anywho(1) | ask.com(6) | autotrader(8) | autotrader.com(2) |
| bet.com(1) | blank planet(4) | bofa(1) | buddylist(2) |
| burlingtom cost factory(3) | carlifornia lottery(8) | carmax(3) | cheater planet(4) |
| citibank(5) | cnn(3) | cnn.com(4) | continental airlines(4) |
| craigslist(9) | craigslist.com(1) | delta airlines(4) | dictionary(10) |
| dictionary.com(3) | dillards(2) | disney(4) | disney channel(2) |
| disney land(4) | dogpile(4) | drudgereport(3) | ebay.com(4) |
| eharmony(8) | espn(5) | expedia(3) | facebook.com(2) |
| fafsa(6) | fedex(4) | fidelity.com(3) | florida lottery(12) |
| florida lotto(3) | food network(3) | food network.com(4) | free ones(4) |
| gamesspot(4) | gay.com(2) | geico.com(3) | google.com(6) |
| guitar tabs(11) | hi5(4) | homedepot(6) | horoscope(6) |
| horoscopes(8) | hostels.com(4) | hotmail(1) | houston chronicle(3) |
| hsn(3) | ikea(3) | illinois lottery(2) | imdb(5) |
| jokes(6) | kazaa(4) | limewire(5) | literotica(3) |
| lotto(5) | love poems(8) | lyrics(9) | lyrics.com(1) |
| mapquest(12) | mapquest.com(8) | mass lottery(2) | match.com(4) |
| mayo clinic(2) | mega millions(8) | michigan lottery(4) | mini clip(4) |
| netzero(5) | newyork lottery(4) | ntl.com(4) | nick.com(4) |
| njlottery(3) | north west airlines(4) | nydailynews.com(4) | palottery(1) |
| photo bucket(5) | picchunter(3) | pizza hut(3) | play boy(5) |
| plenty of fish(3) | pogo.com(9) | powerball(8) | qvc(4) |
| qvc.com(3) | rand mc nally(1) | realter.com(5) | rotters.com(1) |
| runescape(8) | south west airline(2) | south west airlines(2) | sprint(2) |
| sudoku(5) | suntrust bank(3) | tinus lottery(5) | tiawa(5) |
| ticketmaster.com(3) | tmobile(3) | uvision.com(5) | victoria secret(3) |
| wachovia(4) | walmart.com(7) | wama(13) | watchersweb(5) |
| webmd(4) | webshots(3) | www.msn.com(5) | wikipedia(8) |
| wwe.com(5) | www.ask.com(5) | www.yahoo.com(4) | xanga(7) |
| zillow.com(3) | aol(7) | | |

## 4.2 Characteristics of Dataset

AOL logs have approximately 1.62 million URLs. Out of which approximately 1.61 million URLs are html documents totalling of 33GB. These URLs are collected by crawling locally. In this thesis, we use this dataset.

## 4.3 Evaluation and Discussion

The contents of the documents in the collection are indexed after elimination of stop words and stemming in baseline indexer. In the proposed mechanisms only the representative terms are indexed. The representative terms are identified by the weighting factor calculated for each term w.r.t document in collection. Only the top representative terms are indexed for each document.After indexing,performance is evaluated against BM25 based baseline indexer. The retrieval quality, retrieval time and index spaces for each indexer is tabulated.

### 4.3.1 Comparison of index sizes

Table 4.2 and 4.3 shows the comparison of size between indexers. BaselineIndex represents the baseline system where entire content is indexed,BM25Index represents the system where top 20% relevant terms are indexed,HDR represents the system where highest document terms are indexed. There is a reduction of about 15% in memory requirement from baseline index to bm25index and 25% in memory requirement from baseline to HDR index.It is evident from above experiments that we can reduce index size by indexing only representative terms, without compromising retrieval performance.The average positional ranking and recall at 20 for listed queries is summarized in following subsections.

Table 4.2: Comparison of Index sizes of Baseline and BM25(in Giga Bytes)

| BaselineIndex | BM25Index | Reduction |
|---|---|---|
| 14.4GB | 12.6GB | 12.5% |

Table 4.3: Comparison of Index Sizes of Baseline and HDR(in Giga Bytes)

| BaselineIndex | HDR | Reduction |
|---|---|---|
| 14.4GB | 12GB | 17.1% |

### 4.3.2 Retrieval time for queries

The comparison of retrieval times for 1000 most frequent queries from query log are in Table 4.4. It is evident that proposed frameworks is faster than the baseline system. It is due to smaller post list for each index term in the proposed framework.

Table 4.4: Comparison of Retrieval times(for 1000 queries in seconds)

| BaselineIndex | BM25Index | HDR |
|---|---|---|
| 1873.17s | 951.82s | 910.32s |

### 4.3.3 Retrieval quality

We used the following methods to evaluate the systems.

- Average rank: Average rank of the relevant URLs for a given query $q$. It is formally defined as follows.

$$avgRank(q) = \frac{\sum_{d \in R^q} rank(d, q)}{|R^q|}$$

  where $R^q$ is the set of relevant document for the query $q$ and $rank(d, q)$ is the rank position of $d$ in the retrieved list for the query $q$. Lower the $avgRank$, better is the system performance.

- Recall@$k$: It counts the number of relevant results among top $k$ results.

$$recall@k(q) = \frac{R^q \cap ret(q, k)}{|R^q|}$$

  where $ret(q, k)$ represents the top $k$ retrieved results for query $q$.

Table 4.5 and 4.6 present the comparison of $avgRank$ and recall@20 scores between baseline and proposed systems. It clearly shows that there is an improvement in retrieval performance of proposed system compared to the baseline.

Table 4.5: Comparison of average positional ranking of relevant documents for a query

| | BaselineIndex | BM25Index | HDR |
|---|---|---|---|
| Average | 32.46 | 22.96 | 22.60 |

Table 4.6: Comparison of average recall at top 20 for listed queries

| | BaselineIndex | BM25Index | HDR |
|---|---|---|---|
| Average | 0.48 | 0.54 | 0.59 |

# Chapter 5

# Conclusions

The retrieval times, recall@20, average positional ranking shows that the BM25 indexer,HDR indexer yields superior performance for Base line indexer on the document collection with reduction of index space and thereby minimizing memory requirement. This is probably due to the document length dependent smoothing constant of BM25 indexer and indexing all representative terms of normalized rank calculated on BM25 based baseline indexer.

## 5.1 Future Works

The proposed mechanism can be used for including the terms in document which are not contained within it,but retrieves the document among top results.Using the proposed mechanisms we can identify the non-representative and representative terms of document which are used in query formulation by users.

# Bibliography


[1] Ben HE,Ladh Ounis *A study of Parameter Tuning for Term Frequency Normalization. CIKM 2003*,(10-16)

[2] Amit Singhal.Chris Buckley.Mandir Mitra. Pivoted Document Length Normalization. SIGIR 1996: 21-29

[3] S.E.Robertson.C.J.van Rijsbergen and M.F.Porter.Probablistic models of indexing and searchingThree main components that a ect the im- portance of a term in a text are the term frequency factor (tf ), the inverse document frequency factor (idf ), and doc- ument length normalization. 9]

In *Information Processing and Management*, 2000

[4] Ben HE.Ladh Ounis.Term Frequency Normalization Tuning for BM25 and DFR Models. ECIR 2005: 200-214

[5] Thijs Westerveld.Wessel Kraaij.Djoerd Hiemstra.Retrieving Web pages using Content,Links,URLs and Anchors. In: TREC, 2001

[6] David J.Brenes.Daniel Gayo-Avello.Startified Analysis of AOl query log. Inf. Sci. 179(12): 1844-1858 (2009)

[7] Barbaro, M., Jr,T. Z. (2006) A face is exposed for aol searcher no. 4417749. `http://www.nytimes.com/2006/08/09/technology/09aol.html`.

[8] Hafner,K. (2006) Researchers yearn to use aol logs, but they hesitate. `www.nytimes.com/2006/08/23/technology/23search.html`.

[9] Anderson, N. (2006). The ethics of using aol search data. `http://arstechnica.com/news.ars/post/20060823-7578.html`.

[10] Brenes, D. J., Gayo-Avello, D. (2009). Stratified analysis of aol query log. *Information Science*, 179(12),1844–1858.

[11] Carman, M. J., Gwadera, R. t, Crestani, F., Baillie, M. (2009). A statistical comparison of tag and query logs. In *SIGIR'09: Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY:ACM.