Ranked Information Retrieval using Weighted TF IDF

A dissertation Presented by

Saleem Anwar

(2005-NUST-MS PhD-CSE(E)-07)

Submitted to the Department of Computer Engineering in partial fulfillment of the requirements for the degree of

Master of Science

in

Computer Software Engineering

Advisor:

Lt. Col. Dr. Shoab Ahmad Khan

MSC-CSE-5

College of Electrical & Mechanical Engineering

National University of Sciences and Technology

2008

Dr. Younis
Dr. Shoaib
Dr. Farooque Azam
Dr. Asia khanam

THE COMMITTEE

Ranked Information Retrieval using Weighted TF IDF

A dissertation Presented by

Saleem Anwar

Approved as to style and content by:

Lt. Col. Dr. Shoab Ahmad Khan, *Supervisor*

Dr. Farooq-e-Azam, *Member*

Dr. Assia Khanam, *Member*

Brig. Dr. Muhammad Younis Javeed

Department Chair, Computer Engineering

DEDICATIONS

Dedicated to my parents and family

ACKNOWLEDGEMENTS

Nothing worthwhile was ever achieved in isolation. I can not claim to have written this thesis without the significant influence of others. I would like to thank my supervisor **Dr. Shoab Ahmad Khan**. It was invaluable to have the benefit of his decades of experience in the field of Information Technology standing behind his advising of me. I am most grateful that he allowed me to work with him for this thesis, and to be a part of his research group. I would also like to thank **Brig. Dr. Muhammad Younis Javeed**, Department Chair for heading the committee. I would like to thank **Dr. Farooq e Azam Khan** and **Dr. Asia Khanam** for serving on my committee.

I thank my parents and family for standing by me in times of crisis and for being patient with my endless years of study.

I have to thank the large circle of friends I've had at EME and outside each of whom have been special and dear. I feel particularly thankful for having had excellent friends over the years, especially Shahab ud Din who has been a great support and a loyal friend throughout the years. Hassan Arif and Sultan Zia deserves a special mention as being close friends and a continuous source of support, always lending a nonjudgmental ear through good times and bad. I cannot imagine how I could have completed this work without, Muhammad Ahsan who at this point is a sounding board for every idea and thought in my head whether academic or non-academic. His infectious laughter, warmth and affection have made my stay in EME a more than pleasant experience. Muhammad Azhar with his level-headed sensibility, relaxing company, warmth and an ability to make me laugh in times when I have thought myself incapable of doing so, has also been a companion I cannot imagine what EME would have been like without him.

ABSTRACT

Ranked Information Retrieval using Weighted TF IDF

Document Retrieval is the task of retrieving a relevant Document in response to a query, a question, or a reference Document. Tasks such as question answering, summarization, novelty detection, and information provenance make use of a Document retrieval module as a preprocessing step. The performance of these systems is dependent on the quality of the Document-retrieval module. Other tasks such as information extraction and machine translation operate on Documents, either using them as training data, or as the unit of input or output (or both), and may benefit from Document retrieval to build a training corpus, or as a post-processing step.

In this thesis we begin by studying IR Model, then we build a through understanding of exiting IR algorithms like TFIDF, Okapi BM25 and Pivoted length normalization to name a few. During the study of the mentioned algorithms we come up with some deficiencies in retrieval algorithms and started working to eradicate those deficiencies. We proposed a better approach for scoring documents named Weighted TF IDF (WTF IDF) instead of TF IDF where terms are counted rather than weighted with respect to locality of documents and term order. More over we planned to cope with different writing styles by looking for synonym query along with original query, this increase the chances of retrieving some novel information from the corpus.

We have provided the implementation of exiting algorithms and compare the performance with proposed approach WTF IDF and presented the result. The proposed approach has better results than the exiting ones.

Table of Contents

THE COMMITTEE ..... **ii**

DEDICATIONS ..... **iii**

ACKNOWLEDGEMENTS ..... **iv**

ABSTRACT ..... **v**

List of Abbreviations ..... **viii**

List of Figures ..... **ix**

List of Tables ..... **x**

Chapter 1 ..... **1**

1 Introduction **1**

1.1 Contributions 2

1.2 IR Model 3

1.3 Thesis Overview 4

Chapter 2 ..... **5**

2 Related work **5**

2.1 Preprocessing 6

2.1.1 Stopwords Removal 6

2.1.2 Stemming 6

2.1.3 Removal of other useless items 7

2.2 Retrieval Algorithm 7

2.2.1 Boolean Weighting 8

2.2.2 Term Weighting 8

2.2.3 Term frequency 10

Chapter 3 ..... **13**

3 Proposed Approach **13**

3.1 Problems Statement 13

3.2 Suggested Improvements 13
3.2.1 Document Structure / Zones 14
3.2.2 Query Words Contiguity and Order 14
3.2.3 Stylistic Approach 15
3.3 Mathematical Explanation 16
3.3.1 Proposed IR Model 18
3.3.2 Algorithm 19
Chapter 4 22
4 Results and Discussion 22
4.1 Dataset 22
4.2 Performance Evaluation 22
4.2.1 Average Precision and MAP 23
4.2.2 Precision 28
4.2.3 Recall 32
4.2.4 F Measure 36
4.2.5 Fallout 40
4.2.6 Summary of Evaluation 44
4.3 Conclusion 46
Appendix A 47
A. WordNet 1.6 for .NET 47
Getting Started 47
Appendix B 50
B. Porter Stemmer 50
Appendix C 52
C. Stopwords 52
5 Bibliography 54

List of Abbreviations

IR	Information Retrieval
RIR	Ranked Information Retrieval
TF	Term Frequency
DF	Document Frequency
IDF	Inverse Document Frequency
TF IDF	Term Frequency Inverse Document Frequency
WTF IDF	Weighted TF IDF
MAP	Mean Average Precision

List of Figures

FIG. 2.1 IR MODEL 5
FIG. 3.1 REDEFINED QUERY TO Q' FROM Q 15
FIG. 3.2 WTFIDF IR MODEL 18
FIG. 3.3 FLOW CHART WTF IDF ALGORITHM 20
FIG. 4.1 COMPARISON OF AVERAGE PRECISION OF 100 CFQUERIES 25
FIG. 4.2 COMPARISON OF MAP OF 100 CFQUERIES 26
FIG. 4.3 COMPARISON OF MAP OF 100 CFQUERIES 27
FIG. 4.4 COMPARISON OF MAP OF 100 CFQUERIES 27
FIG. 4.5 COMPARISON OF AVG. OF PRECISION 32
FIG. 4.6 COMPARISON OF AVG. OF PRECISION @10 32
FIG. 4.7 COMPARISON OF AVG. OF RECALL 36
FIG. 4.8 COMPARISON OF AVG. OF RECALL @10 36
FIG. 4.9 COMPARISON OF AVG. OF F-MEASURE 40
FIG. 4.10 COMPARISON OF AVG. OF F-MEASURE @10 40
FIG. 4.11 COMPARISON OF AVG. OF FALLOUT 44
FIG. 4.12 COMPARISON OF AVG. OF FALLOUT @10 44
FIG. 4.13 COMPARISON OF ALL EVALUATION METRICS 45
FIG. 4.14 COMPARISON OF ALL EVALUATION METRICS @10 46

List of Tables

TABLE 2.1 STOPWORDS 6
TABLE 2.2 WORDS STEMMED TO ROOT WORDS USING PORTER STEMMER 7
TABLE 3.1 MATRIX REPRESENTING SAMPLE RUN OF PROPOSED ALGORITHM WITH DUMMY VALUES 19
TABLE 4.1 CFQUERY NO 00033 USING PROPOSED ALGORITHM ON THE CFC SHOWING IR EVALUATION METRICS 23
TABLE 4.2 COMPARISON OF AVERAGE PRECISION AND MAP OF 100 CFQUERIES 25
TABLE 4.3 COMPARISON OF MAP @ 25, 50, 75 & 100 CFQUERIES 26
TABLE 4.4 COMPARISON OF PRECISION AND PRECISION AT R=10 OF 100 CFQUERIES 31
TABLE 4.5 COMPARISON OF RECALL AND RECALL AT 10 OF 100 CFQUERIES 36
TABLE 4.6 COMPARISON OF F-MEASURE AND F-MEASURE AT R=10 OF 100 CFQUERIES 40
TABLE 4.7 COMPARISON OF FALLOUT AND FALLOUT AT R=10 OF 100 CFQUERIES 44
TABLE 4.8 COMPARISON OF ALGORITHMS USING ALL EVALUATION METRICS 45
TABLE 4.9 COMPARISON OF ALGORITHMS USING ALL EVALUATION METRICS @R = 10 45

Chapter 1

1 Introduction

Information Retrieval (IR) is the task of retrieving a relevant Document in response to a query, a question, or a reference Sentence. IR is the science of searching for information in documents or searching for documents themselves. IR covers many aspects of getting information such as ranking query result, question answering, novelty detection, and information provenance. The performance of these systems is dependent on the quality of the retrieval module. Other tasks such as information extraction and machine translation operate on documents, either using them as training data, or as the unit of input or output (or both), and may benefit from document retrieval to build a training corpus, or as a post-processing step. Automated IR systems are used to reduce information overload. Many universities and public libraries use IR systems to provide access to books, journals, and other documents[1]. Web search engines such as Google, Yahoo search and Live Search (formerly MSN Search) are the most visible IR applications.

An information retrieval process begins when a user enters a query into the system. Queries are formal statements of information needs, for example search strings in web search engines. In information retrieval a query does not uniquely identify a single object in the collection. Instead, several objects may match the query, perhaps with different degrees of relevancy. An object is an entity which keeps or stores information in a database. User queries are matched to objects stored in the database. Depending on the application the data objects may be, for example, text documents, images or videos. Often the documents themselves are not kept or stored directly in the IR system, but are instead represented in the system by document surrogates. Most IR systems compute a numeric score on how well each object in the database

1 http://en.wikipedia.org

match the query, and rank the objects according to this value. The top ranking objects are then shown to the user. The process may then be iterated if the user wishes to refine the query.

The notion of relevance is captured by the way the similarity between the query and the document is modeled. It follows that documents containing more query terms would be more relevant than documents containing fewer query terms.

The most popular family of document retrieval models is the vector space model, which typically uses the TF IDF family of term weighting schemes. Although the various TF IDF approaches differ in how the term frequency (the TF part) and the inverse document frequency (the IDF part) are combined, a common theme among the models in this family is that documents with a higher term frequency have a higher score. A second family of retrieval models is the language model approach, which also assigns a higher score to documents that have a higher term frequency.

In both cases, documents containing multiples of the same query term will be scored higher than documents containing singleton query terms. The assumption that documents containing more query terms are more relevant (or more similar to the query) extends to efforts to capture multiple ways of expressing the same concepts. Techniques for query expansion and pseudo-relevance feedback assume that by expanding the query (perhaps with synonyms or other related terms) we can capture documents that are relevant but using a slightly different vocabulary than the original query.

1.1 Contributions

In this thesis we address the question of how to retrieve documents (retrieval module), and demonstrate document retrieval in the context of query document retrieval, question answering, novelty detection, and information provenance. Our emphasis will be on refining the retrieval modal, typical called TF IDF is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a

document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. Variations of the TF IDF weighting scheme are often used by search engines to score and rank a document's relevance given a user query. In addition to TF IDF weighting, Internet search engines use link analysis based ranking to determine the order in which the scored documents are presented to the user. There are many different formulas used to calculate TF IDF but none of them is an exhaustive implementation of TF IDF. We have suggested refinements in TF IDF and made it some what comprehensive, incorporating different stylistic, term proximity and document structures. We have given generic implementation of our retrieval modal with out restricting it for specific domain.

1.2 IR Model

Although we discuss IR models in later chapters, it is helpful to understand the way the models are evaluated, and the corpora used for the evaluation of information retrieval systems. Information retrieval systems typically consist of an index of documents, and a query which is usually provided by the user. The task of the system is to retrieve documents that are relevant to the user's query. Most often, documents are presented in a ranked list, where the documents most likely to be relevant are presented at the top of the list, and documents less likely to be relevant are presented at the bottom of the list. Documents and queries can be any unit of information, but this thesis concerns sentence retrieval, which means that sentences are treated as documents. To avoid confusion with other types of documents the term document refers to newswire articles. In general a query is any information the system uses to rank documents, and may include terms the user has provided in addition to other information about the user, such as documents the user has previously viewed, or expansion terms the system has provided. In this work a query is a question, a set of terms, or a sentence. A gold standard Cystic Fibrosis Corpus (CFC) (1) is used for evaluating the algorithm and comparing the performance of existing and proposed algorithms. The Cystic Fibrosis Database (CF) consists of seven files: cf74 to cf79 containing 1,239 documents published from 1974 to 1979 discussing

Cystic Fibrosis aspects. CF query a set of 100 queries with the respective relevant documents as answers by 4 different domain experts at scale of 0, 1, 2 from irrelevant, slightly relevant and relevant. Various metrics are used to evaluate retrieval systems. Most of the work in this thesis is evaluated using precision at N. Precision at N is the number of relevant documents in the top N documents, averaged over all of the queries. A related metric is mean average precision which is the mean of the precision at the rank of each relevant document for a given query, averaged over all queries. In this thesis, the results reported for average precision refer to mean average precision. Recall is the number of relevant documents that have been retrieved, divided by the total number of relevant documents. A t-test is an appropriate significance test for precision at N and mean average precision because these metrics simply count the number of relevant sentences in a set of retrieved sentence, ignoring the rank of the sentences.

1.3 Thesis Overview

We begin the thesis with introduction to the IR in Chapter 1, later on we'll be discussing differ IR models, specially TF IDF and its variations and factors effecting it. Chapter 3 and later will discuss our improvements in TF IDF as proposed titled Weighted TFIDF; its mathematical explanation proposed algorithm, Results and discussion are discussed in last Chapter.

Chapter 2

2 Related work

This chapter entails history of IR systems, the basic algorithm and steps that comprise RIR system. Most of the steps are similar to that of any text processing application. Text processing tasks are driven by huge textual information. For relevant information retrieval the textual data must be first passed through preprocessing steps. The preprocessed data is then represented in numeric form using suitable representation. The main steps of the system are shown in Figure2.1

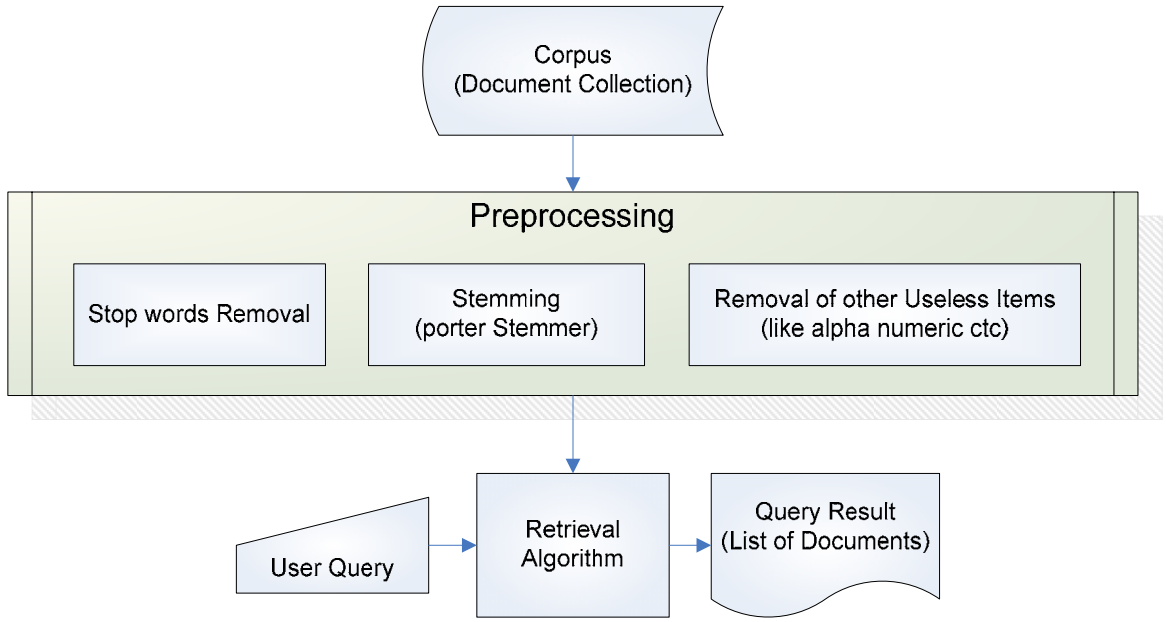

Fig. 2.1 IR Model

2.1 Preprocessing

In text retrieval tasks the preprocessing of the textual information is very critical and important. Main objective of text preprocessing is to remove data which do not give useful information regarding the class of the document. Furthermore we also want to remove data that is redundant. Most widely preprocessing steps in the textual retrieval tasks are removing of stop words and performing stemming to reduce the vocabulary. Some additional steps might also be required to remove other useless items, like images, alpha numeric words etc

2.1.1 Stopwords Removal

In textual information retrieval there are words that do not carry any useful information and hence are ignored during indexing and searching. Stop terms definition in context of internet search engines is 'words that is so common on the Internet that search engines ignore them. E.g. homepage, home page, www, Web, Web page, the, of, that, is and, to, etc[2]'. In terms of database searches it is defined as 'words that databases will not search for[3]'. There is a possibility of two types of stop words, one the generic stop word list can be used for all corpuses and the other may be corpus dependent. e.g.

A	about	above	across	after	afterwards	Any	anyhow	anyone	anything
Again	against	All	almost	alone	Along	Are	around	as	At

Table 2.1 Stopwords

2.1.2 Stemming

The second main preprocessing tasks applied in textual information retrieval teasks is the stemming. It can be defined as 'an algorithm developed to reduce a search query to its stem or root form, in other words, variations of particular words such as past tense and plural and

2 www.pro-seo.com/glossary.html

3 www.methodist.edu/library/guides/libraryvocab.htm

singular usage are taken into account when performing a search, For example, applies, applying & applied matches apply[4]". In the context of searching it can be defined as "expansion of searches to include plural forms and other word variations[5]". In the context of document classification we can define it to be a process of representing words and its variants with its root. We used the porter stemming algorithms described in (2)[6]. Figure 2.4 shows some examples of the words after being stemmed with porter's algorithm.

Words	ponies	caress	cats	feed	Agreed	plastered	Falling	tanned	troubling
Stem	poni	caress	cat	Fe	Agree	plaster	Fall	Tan	troubl

Table 2.2 Words stemmed to root words using porter stemmer

2.1.3 Removal of other useless items

Other useless items might be words with too small size or alpha numeric words. Investigation of English vocabulary shows that almost all such words whose length are lesser than or equal to two contains no useful information regarding class of the document. Examples includes a, is, an, of, to, as, on etc. though there are words which have length of three and are useless like the, for, was, etc but removing all such words will cost us loosing some words that are very useful in our domain, like sex, see, sir, fre. There might be many words found in the corpus that are alpha numeric. Removal of those terms was important.

2.2 Retrieval Algorithm

In full-text information retrieval users of for instance web search engines enter some vague ambiguous query for what they are looking for, and it is the search engine's task to return a list of links to documents that are relevant to the user's request. Simply returning an unordered list

4 www.pr3.co.uk/seo/seo-glossary.php

5 members.optusnet.com.au/~webindexing/Webbook2Ed/glossary.htm

6 http://www.tartarus.org/~martin/PorterStemmer

of documents that contain the words the user entered is insufficient. On any very large document collection like the World Wide Web, many thousands of documents might contain the words the user has entered, but only a few of those documents will actually be relevant to the user[7] (3).

2.2.1 Boolean Weighting

The Boolean model of information retrieval is an example of a model that does not use word statistics and does not rank the documents. A Boolean retrieval system is designed to retrieve an unordered list of documents that contain the precise combination of words included in the query. Mathematically it can be represented as

$$W(t_i|D_j) = \begin{bmatrix} 1 & if\ tf_{ij} > 0 \\ 0 & otherwise \end{bmatrix}$$

A term ti will get a weight of 0 in document j if it is not present otherwise it will get a weight of 1. User may use the standard Boolean operators between the query terms like AND, OR, NOT to refine the result. Many models of ranked retrieval have been proposed that try to tackle the Boolean model's inability rank documents (4).

2.2.2 Term Weighting

Other IR systems uses word statistics in a rather ad-hoc way: the so-called TF IDF weights. These weights use a product of the term frequency TF and the inverse document frequency IDF. The TF component is related to the number of times a term occurs in a document, whereas the IDF component is inversely related to the number of documents in which the term occurs. Salton and Buckley (1988) report experiments with a total of 1,800 different variations of TF IDF weights (5), and many more variations have been suggested since. The use of statistical language models for information retrieval was recently proposed by Hiemstra (1998) (6), Miller

[7] **http://www.glotinternational.com**

et al. (1999) (7) and by Ponte and Croft (1998) (8). These models do not rely on TF IDF weighting. Instead, they use simple, easy to understand, probability measures of the form: “If the word occurs three times in a document that contains 100 words in total, then the probability of that word given the document is 0.03.”; or “if the word occurs 2,000 times in a corpus of a million words, then the probability of that word in the English language (assuming that we are searching for English documents) is 0.002”. A linear combination of these two probability measures results in a language model that behaves like the TF IDF weights, outperforming the best-performing TF IDF variations (Hiemstra 2000) (9).

Mathematically, there exist many term weighting methods which will calculate the weight for term differently. These weighting approaches are based mostly on following observations (10).

- The relevance of a word to the class of a document is proportional to the number of times it appears in the document.
- The discriminating power of a word between documents is less, if it appears in most of the documents in the documents collection. In other words, terms which are present in lesser number of documents are more discriminative.

Comparative study of different term weighting approaches in automatic text retrieval is presented by Salton and Buckley in (4) . Before defining each of the term weighting methods individually we define few terms first to make the understanding easier.

tf_{ij} as the frequency of term i in document ***j***, ***N*** as the total number of documents or documents in the corpus, n_i as the number of documents in the corpus where term ***i*** appears and ***M*** as the number of terms in the document collection (after stop words removal and stemming).

2.2.3 Term frequency

Also widely know as bag of words weighting and vector space model. It is also relatively simple weighting which counts the number of occurrences of term in an document. Mathematically it can represented as

$$Trerm\ Frequency\_W_{ij} = tf_{ij}$$

We performed few experiments with this weighting until we discovered the TFIDF with lengths normalized later on.

2.2.3.1 Term Frequency with Lengths Normalized

In order to cope with documents of different lengths a variant of term frequency is introduced. Here every weight of a term will be divided by the total number of terms frequencies in the document instance. Mathematically it can be represented as

$$Trerm\ Frequency\ Normalized\_W_{ij} = \frac{tf_{ij}}{M}$$

2.2.3.2 Term Frequency inverse document frequency

This is the most widely used weighting scheme. Term frequency and Boolean weighting do not take the global statistics of the term into account. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. TFIDF representation takes this property coupled with term frequency to define a new weighting which can be expressed mathematically as

$$TFIDF\_W_{ij} = tf_{ij} \times \log(N/n_i)$$

Where n_i is number of documents containing term t_i.

2.2.3.3 Term Frequency inverse document frequency with lengths Normalized

To account for the documents of different lengths the weights obtained from the TFIDF are normalized. Mathematically the normalized version can be expressed as

$$TFIDF\_Normalized\_W_{ij} = \frac{tf_{ij}}{M} \times \log(N/n_i)$$

Two (TF IDF)-based relevance estimation techniques have become particularly dominant: Okapi BM25 (11) (12) and pivoted length normalization (13) are the other variations of TF.IDF

2.2.3.4 Okapi BM25 & BM25f

Okapi BM25 is a bag-of-words retrieval function that ranks a set of documents based on the query terms appearing in each document, regardless of the inter-relationship between the query terms within a document (e.g., their relative proximity). It is a ranking function used by search engines to rank matching documents according to their relevance to a given search query. The name of the actual ranking function is BM25. To set the right context, however, it usually referred to as "Okapi BM25", since the Okapi information retrieval system, implemented at London's City University in the 1980s and 1990s[8], was the first system to implement this function. BM25, and its newer variants, e.g. BM25F (a version of BM25 that can take document structure and anchor text into account), represent state-of-the-art retrieval functions used in document retrieval, such as Web search.

Mathematically Okapi BM25 is: Given a query Q, containing keywords $q_1,...,q_n$, the score of a document D is:

$$score(D,Q) = \sum_{i=1}^{n} \left(log \frac{n - df_i + .5}{df_i} \times \frac{tf_i}{tf_i + 0.5 + 1.5 \times \frac{dl}{avdl}} \right)$$

8 http://en.wikipedia.org/wiki/Okapi_BM25

2.2.3.5 Pivoted Length Normalization

One version of the pivoted normalization scheme is shown in Equation, where q_{tf} is the number of occurrences of t in Q rest of the formula is pretty similar to okapi BM25

$$score(D,Q) = \sum_{i=1}^{n} \left(\frac{1 + \log(1 + \log(tf_i))}{0.8 + 0.2 \times \frac{dl}{avdl}} \times qtf \times \log\left(\frac{N+1}{df_i}\right) \right)$$

Chapter 3

3 Proposed Approach

3.1 Problems Statement

Though a lot of work is done for term weighting methods, we saw different variations of TF IDF schemes, but none of them can be considered an exhaustive implementation of term weighting scheme. We identified following problems with existing TF IDF techniques.

- Though TF Is one of the best known indicators of relevancy but it considers document as bag of words, a few implementations give some importance to document structures Like Okapi BM25f (5) and CTR (14), yet none gives clear formula incorporating rich formatting and document structuring.
- Secondly, Term proximity, is an idea which has seen recent interest. In this context, term proximity refers to the lexical distance (15) between query terms, calculated as the number of words separating query terms in a document. Modern experimental retrieval systems support proximity query operators (16). Many researchers now use proximity enhanced approaches which has shown positive results (17) (18) (19). Term proximity does not give importance to order of query words.
- Lastly, none of retrieval model has incorporated different stylistic approaches in calculating term weights. They either ignore synonym words or give them importance equivalent to original query words.

3.2 Suggested Improvements

Following improvements are suggested in our WTF IDF.

3.2.1 Document Structure / Zones

A formatted document (either through rich text formatting or through mark up tags) can be divided into some zones. Each zone virtually may have certain associated significance, i.e. a term occurring in one zone or part of document may have different worth than the occurrence of same term in certain different zone. Like occurrence of a query term in title, headings or sub headings may have greater weight than same term occurring in body of document.

3.2.2 Query Words Contiguity and Order

We are going to redefine the term proximity, we are not only interested in nearness of query terms but also the same order of query terms in documents as in query. As changing order of query terms lead to inverse results of what you are looking for e.g. “cat killed rat” may seems weird if its order get shuffled e.g. “rat killed cat”. But enforcing only this strict order will lead the retrieval model to boolean retrieval model. So we suggested not only redefining the query but also redefining the term “term” in its original meaning. Previously a query word was called term, while term can be a word, phrase or expression. So we’ll be using term in its original meaning.

If Q be the query consisting of n words (q_1, q_2, q_3, ..., q_{n-1}, q_n). The redefined query would be Q’ of m terms where m = n (n+1) /2 (t_1, t_2, t_3, …., t_{m-1}, t_m). Where $t_1 = [q_1, q_2, q_3, ..., q_{n-1}, q_n]$, $t_2 = [q_2, q_3, ..., q_{n-1}, q_n]$, $t_3 = [q_1, q_2, q_3, ..., q_{n-1}]$, … $t_{m-1} = [q_{n-1}]$, $t_m = q_n$.

To understand more simply we take n=4, i.e. query with 4 words. Then m = 4(4+1)/2 = 10, i.e. the redefined query contains 10 query terms, as shown in tree like structure in fig 3.1 where Q $\{q_1, q_2, q_3, q_4\}$ → Q’ $\{t_1, t_2, t_3, ..., t_{10}\}$ and ts is the term size at each tree level

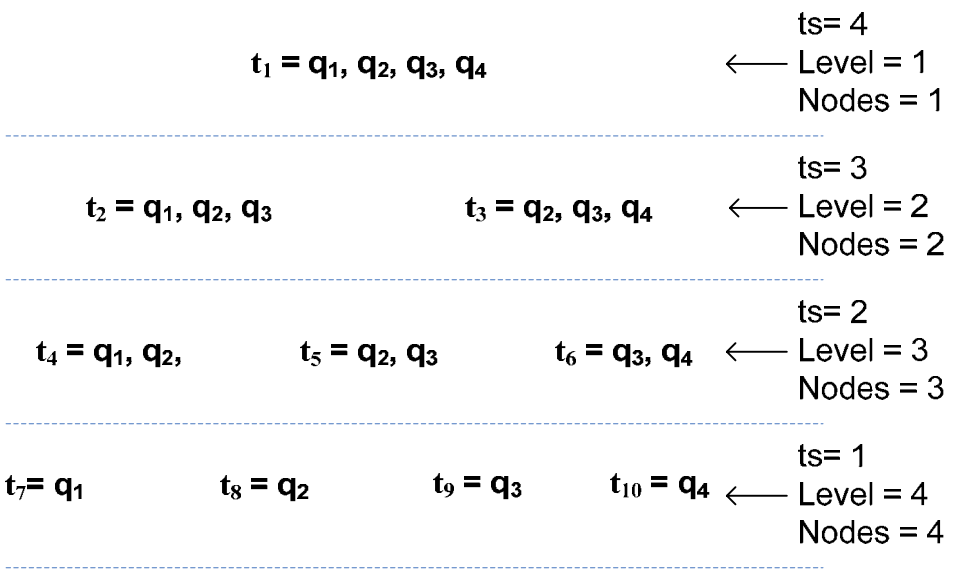

Fig. 3.1 Redefined query to Q' from Q

Now looking for the frequency of these query terms rather than looking for query words. More over a query term will have its influence in calculation of term weight equals to term size divided by query size, where term size is the number of query words in a term and query size is number of query words in original query. It automatically will give more importance to the occurrence of whole query rather than individual words.

3.2.3 Stylistic Approach

Each writer has its own writing style and may use different words and phrases to represent same concept, idea, topic or story while writing a document. Query words; if different from writing style of that document will not consider document as relevant, even if it belongs to the same concept for which a corpus is being queried but with different words. We've rephrased the query with its synonyms and found its weight. It adds up in previous weight calculated for the document. Here it is logical to give relatively less wait to synonym query from original

query. WordNet 1.69 is an online lexical reference system developed at the University of Princteon presents word and its synonym sets

3.3 Mathematical Explanation

Let we have a document collection D of |D| documents where each document is represented as d_i. Given a query Q consisting of n query words.

Q = {q1, … , qn}

Instead of treating query as bag of words, we gave importance to sequences words in query. We redefined the query words into query terms, where each term may consists of one or more query words such that order and proximity of query words remain intact. The redefined query is Q′

Q′= {t1, t2, …, tm}

Where m = n (n+1)/2 and t1 = [q1, …, qn], t2 = [q1, q2, …qn-1], t3=[q2, q3, …, qn] , … , tm-1 = [qn-1] , tm = [qn].

We simply don't count the occurrence of term in whole document but we'll be having weighted count of term at different formatting levels of document, where each formatting level have an associated value for its relative weight which can linearly or exponentially effect the count.

$$C(t|d) = \sum_{f=1}^{h} (C_{tf} \times V_f) \qquad (i)$$

[9] http://www.cogsci.princeton.edu/~wn/

Here C(t|d) is count of term in document linearly effected by formatting, which is equal to sum of count of that term at different formatting zones represented by Ctf multiplied by value of that formatting zone Vf.

$$W(t|d) = C(t|d) \times {}^{ts}/_{qs} \qquad (ii)$$

W(t|d) is weight of term for a given document, ts is term size and qs is the query size.

Similarly rephrasing the query Q to Q′ (synonym query) with set of synonyms for each query word Q″ = {s1, s2, … , sk}where s1, s2 , … , sk are synonyms of original query words where k is an arbitrary number depending upon the number of synonyms available for all words. It is logically not to look for the sequence of occurrence of Q″ words in this case ts = 1 and qs for Q″ is k so equation i & ii can be modified as

$$C(s|d) = \sum_{f=1}^{h} (C_{sf} \times V_f)$$

$$W(s|d) = C(s|d) \times {}^{1}/_{k}$$

Here subscript s is for synonym in place of t for term of eq i & ii.

Utilizing the same IDF formula mentioned by Okapi

$$IDF(x) = \log \frac{|D| - |D'| + 0.5}{|D'| + 0.5}$$

Here IDF(x) is inverse document frequency for a given query term t or synonym terms s, |D| is the number of documents in corpus, |D′| is number of document containing term.

Combining to calculate score of document for given query

$$\text{Score}(d \mid Q) = \sum_{i=1}^{m} \left[\text{IDF}(t_i) \times W(t_i \mid d) * \frac{\text{avgdl}}{\text{dl}} \right] + \alpha \sum_{j=1}^{k} \left[\text{IDF}(s_j) \times W(s_j \mid d) * \frac{\text{avgdl}}{\text{dl}} \right]$$

Here 0=< α <=1 influence of Stylistics on score.

3.3.1 Proposed IR Model

The proposed model is shown graphically in flow chart here as Fig. 3.2 elaborating the steps involved in any IR system, especially in this WTF IDF approach.

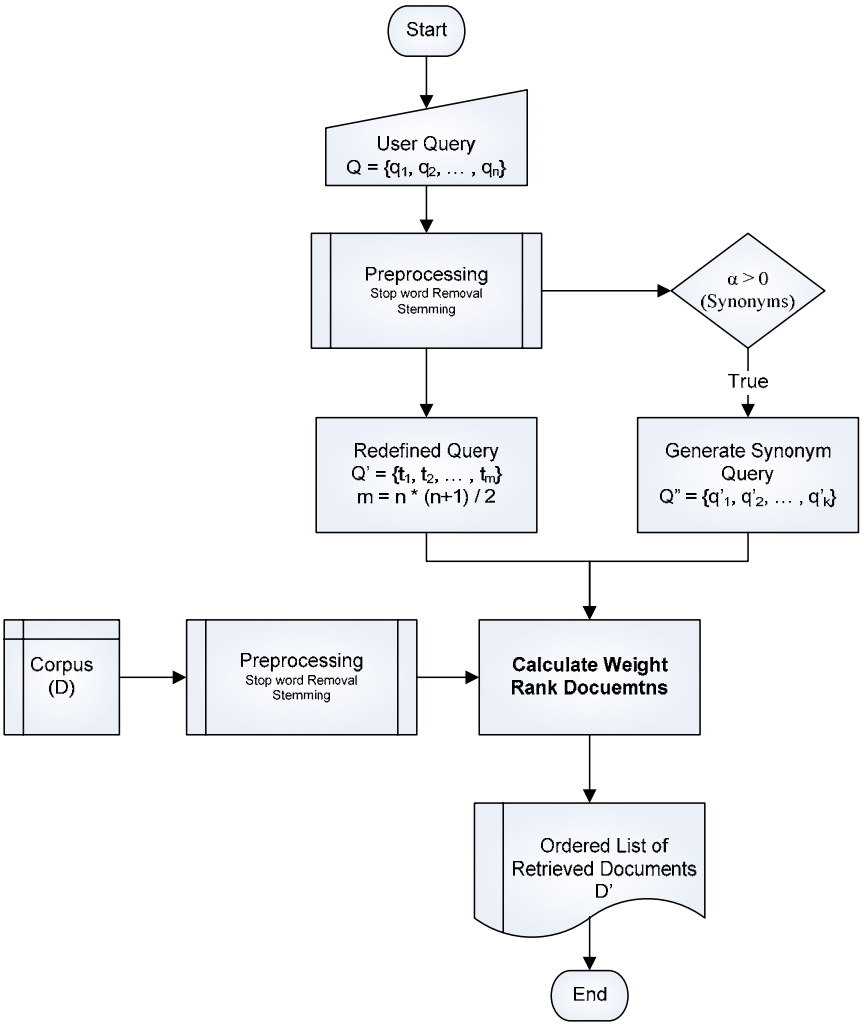

Fig. 3.2 WTFIDF IR Model

3.3.2 Algorithm

Algorithm of proposed approach WTF IDF explained in given six steps, show graphically in Fig.

3.3 Table: 1 represents the sample run of the same algorithm

Step 1: Pre-process the Corpus.

Step 2: Redefine Q to Q′ and to Q″ if α is not zero.

Step 3: look for the weight of tj in each document and construct a matrix M of |D|+1 rows and |m+k+1| or |m+1| columns, where each Mij will be containing weight of jth redefined term in ith document.

Step 4: Calculate the DF in last row against each term. Used for IDF.

Step 5: Calculate the score against each document in last column of matrix using the proposed formulae.

Step 6: Sorting M based on last column gives ordered list of documents against a query.

Docs	T_1	T_2	T_3	…	T_m	S_1	S_2	S_3	…	S_k	Score
D_1		2	3		3		1	5			14
D_2	3	2	1		2	1		2		6	17
D_3		6				4		1		2	13
:											X
D_N	5		4				3				12
DF	2	3	3	x	2	2	2	3	x	2	

Table 3.1 Matrix Representing Sample Run of proposed algorithm with dummy values

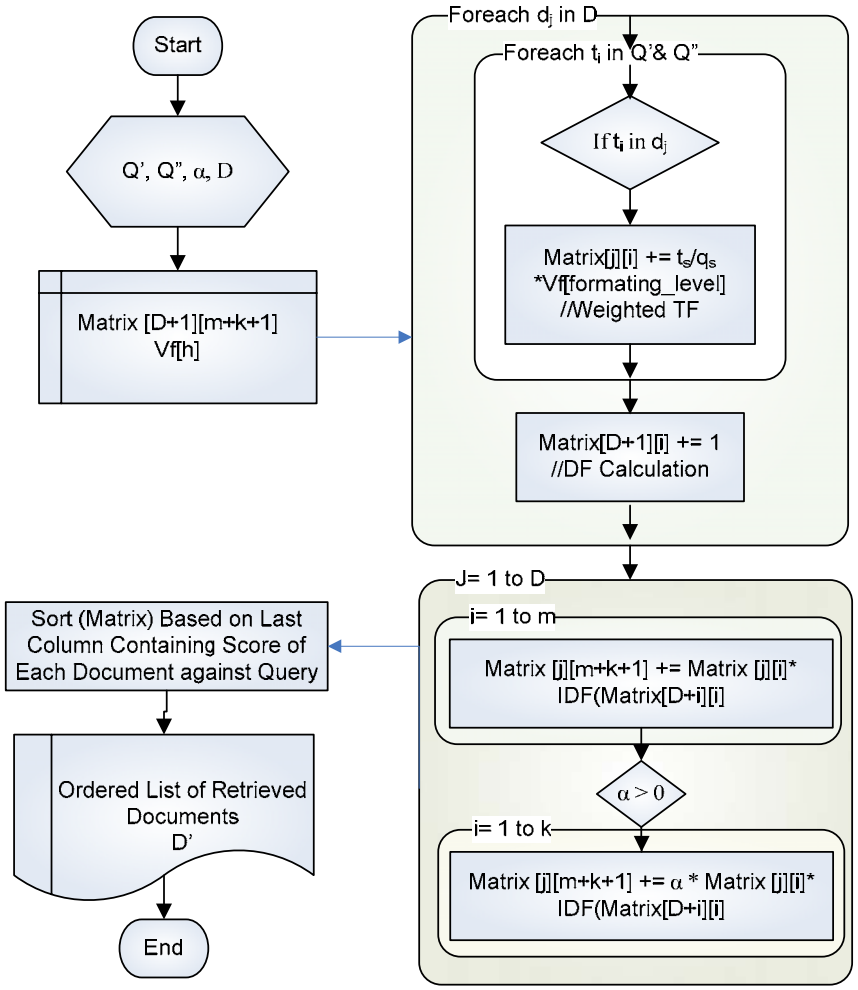

Fig. 3.3 Flow chart WTF IDF Algorithm

3. Proposed Approach

Chapter 4

4 Results and Discussion

Measuring the performance of IR system is non-trivial. Generally an algorithm is evaluated using a test corpora and then calculating Precision, Recall and Mean Average Precision (MAP) (20), a relatively less used measure are F- Measure, E- Measure and Fallout are also used as matrices for measuring the effectiveness of retrieval algorithm.

4.1 Dataset

A gold standard Cystic Fibrosis Corpus (CFC) (1) is used for evaluating the algorithm and comparing the performance of existing and proposed algorithms. The Cystic Fibrosis Database (CF) consists of seven files: cf74 to cf79 containing 1,239 documents published from 1974 to 1979 discussing Cystic Fibrosis aspects. CF query a set of 100 queries with the respective relevant documents as answers by 4 different domain experts at scale of 0, 1, 2 from irrelevant, slightly relevant and relevant.

4.2 Performance Evaluation

The performance of proposed approach WTFIDF is compared with TFIDF, BM25 and pivoted length normalized, all variable such as stemming using porter stemmer, stopwords and pre-processing were identical.

We have evaluated proposed and given algorithms against above listed evaluation matrices. Table 4.1 presents a sample run of randomly selected cfquery no 00033 using proposed algorithm on the CFC.

@ R	Precision	Recall	F Measure	E-Measure β=1	E-Measure β>1 (1.6)	E-Measure β<1 (0.7)	Fallout
5	1	0.087719	0.16129	0.16129	0.114594	0.266458	0
10	1	0.175439	0.298508	0.298508	0.222629	0.445609	0
20	0.85	0.298246	0.441559	0.441559	0.357982	0.570582	0.002538
40	0.625	0.438597	0.515464	0.515464	0.475016	0.561798	0.01269

Table 4.1 cfquery no 00033 using proposed algorithm on the CFC showing IR evaluation metrics

4.2.1 Average Precision and MAP

Average Precision and MAP of all 100 queries using proposed and given algorithms is presented in Table 4.2 and Table 4.3 and graphically in Fig. 4.1 and Fig. 4.2. Table 4.3 presents MAP of 25, 50, 75 and 100 queries showing WTFIDF outperformed exiting approaches at each level; same is represented in Fig. 4.3 and Fig. 4.4.

Query No	TF IDF	Pivoted LN	Okapi BM25	WTF IDF
1	0.0656	0.0956	0.2147	0.2701
2	0.1775	0.3807	0.0280	0.1245
3	0.1180	0.0846	0.1290	0.1451
4	0.1827	0.0374	0.0578	0.2644
5	0.1603	0.2018	0.3376	0.5010
6	0.1353	0.1579	0.3099	0.1752
7	0.0527	0.0686	0.0711	0.1819
8	0.0463	0.0338	0.0928	0.0348
9	0.0751	0.1298	0.2686	0.1760
10	0.0671	0.0961	0.0872	0.2011
11	0.3910	0.4729	0.5232	0.6392
12	0.0057	0.0268	0.1508	0.1800
13	0.1169	0.1538	0.0931	0.1481
14	0.1065	0.1783	0.2139	0.1782
15	0.1365	0.1403	0.2923	0.5927
16	0.2083	0.2530	0.3407	0.2187
17	0.0874	0.0814	0.1732	0.1346
18	0.4546	0.3012	0.2832	0.4616
19	0.0255	0.0587	0.1500	0.0760
20	0.4602	0.1348	0.4688	0.6469
21	0.1224	0.0859	0.2556	0.1782
22	0.1328	0.1077	0.2880	0.2309
23	0.1518	0.0503	0.0761	0.0874
24	0.0376	0.0355	0.0316	0.0636
25	0.1263	0.2169	0.2220	0.2760
26	0.1231	0.0635	0.2765	0.5132
27	0.0997	0.0280	0.0436	0.1985
28	0.0784	0.1223	0.1482	0.2097
29	0.0409	0.0636	0.0869	0.2199

30	0.1174	0.0719	0.1656	0.4819
31	0.2962	0.0725	0.3824	0.4248
32	0.0931	0.0348	0.0909	0.0846
33	0.3890	0.5844	0.2048	0.5563
34	0.2857	0.3165	0.3964	0.3694
35	0.0552	0.0524	0.0741	0.1910
36	0.0875	0.0478	0.0933	0.0718
37	0.2631	0.4003	0.5833	0.5557
38	0.0511	0.1933	0.2499	0.5211
39	0.3181	0.4255	0.5003	0.5720
40	0.1385	0.1738	0.2390	0.1910
41	0.2278	0.0416	0.1201	0.1833
42	0.2153	0.1969	0.1723	0.1959
43	0.1964	0.2671	0.3016	0.3672
44	0.2552	0.2352	0.4074	0.4931
45	0.2194	0.0340	0.1487	0.3156
46	0.0713	0.0461	0.2103	0.2272
47	0.0526	0.0784	0.1244	0.0658
48	0.2333	0.1268	0.3586	0.2963
49	0.1937	0.1615	0.2864	0.3170
50	0.1473	0.1487	0.1417	0.2669
51	0.1685	0.4483	0.4061	0.4061
52	0.5008	0.4483	0.0466	0.6469
53	0.2189	0.1556	0.1905	0.2284
54	0.2420	0.2005	0.3176	0.2572
55	0.0131	0.0413	0.0216	0.0412
56	0.1027	0.0625	0.0666	0.1233
57	0.2002	0.4474	0.4495	0.3169
58	0.0933	0.4038	0.2935	0.2293
59	0.1145	0.2035	0.1579	0.1788
60	0.0266	0.0399	0.0504	0.1874
61	0.1755	0.1621	0.2690	0.2579
62	0.2182	0.2415	0.2731	0.2263
63	0.0612	0.0766	0.0840	0.1454
64	0.1853	0.1414	0.1835	0.1244
65	0.1871	0.1679	0.2351	0.3335
66	0.0996	0.1295	0.1778	0.1366
67	0.0629	0.0966	0.3708	0.1649
68	0.0800	0.0723	0.1617	0.0894
69	0.1535	0.0730	0.0593	0.2361
70	0.1150	0.0954	0.0815	0.3636
71	0.2816	0.3206	0.0873	0.4038
72	0.2134	0.1677	0.1516	0.3291
73	0.3288	0.0449	0.1466	0.4812
74	0.1968	0.1771	0.2297	0.3649
75	0.2777	0.0943	0.3268	0.3337
76	0.1820	0.0301	0.1869	0.0733
77	0.1704	0.0894	0.1500	0.1103
78	0.0690	0.0934	0.2421	0.2421
79	0.0369	0.0921	0.2309	0.1910
80	0.1487	0.2740	0.3645	0.2677
81	0.1788	0.0701	0.2661	0.2244
82	0.0742	0.1234	0.2498	0.1320
83	0.0960	0.1356	0.2340	0.1098
84	0.1632	0.1578	0.2025	0.1793
85	0.0217	0.0842	0.0822	0.0770
86	0.0931	0.1737	0.2668	0.2266
87	0.0400	0.0765	0.0876	0.0909
88	0.0818	0.0510	0.0614	0.1531
89	0.2437	0.2417	0.3008	0.2420

90	0.5462	0.5149	0.4169	0.4829
91	0.2496	0.2808	0.3473	0.2561
92	0.0961	0.1210	0.3015	0.3011
93	0.0434	0.0194	0.0289	0.1315
94	0.2544	0.2189	0.4441	0.4053
95	0.2367	0.0606	0.0269	0.4041
96	0.3319	0.0274	0.0918	0.2531
97	0.1624	0.1424	0.2039	0.1966
98	0.1311	0.0505	0.1210	0.3611
99	0.6022	0.2127	0.0268	0.5172
100	0.2101	0.0250	0.0434	0.2979
MAP	**0.1667**	**0.1545**	**0.2078**	**0.2661**

Table 4.2 Comparison of Average Precision and MAP of 100 cfqueries

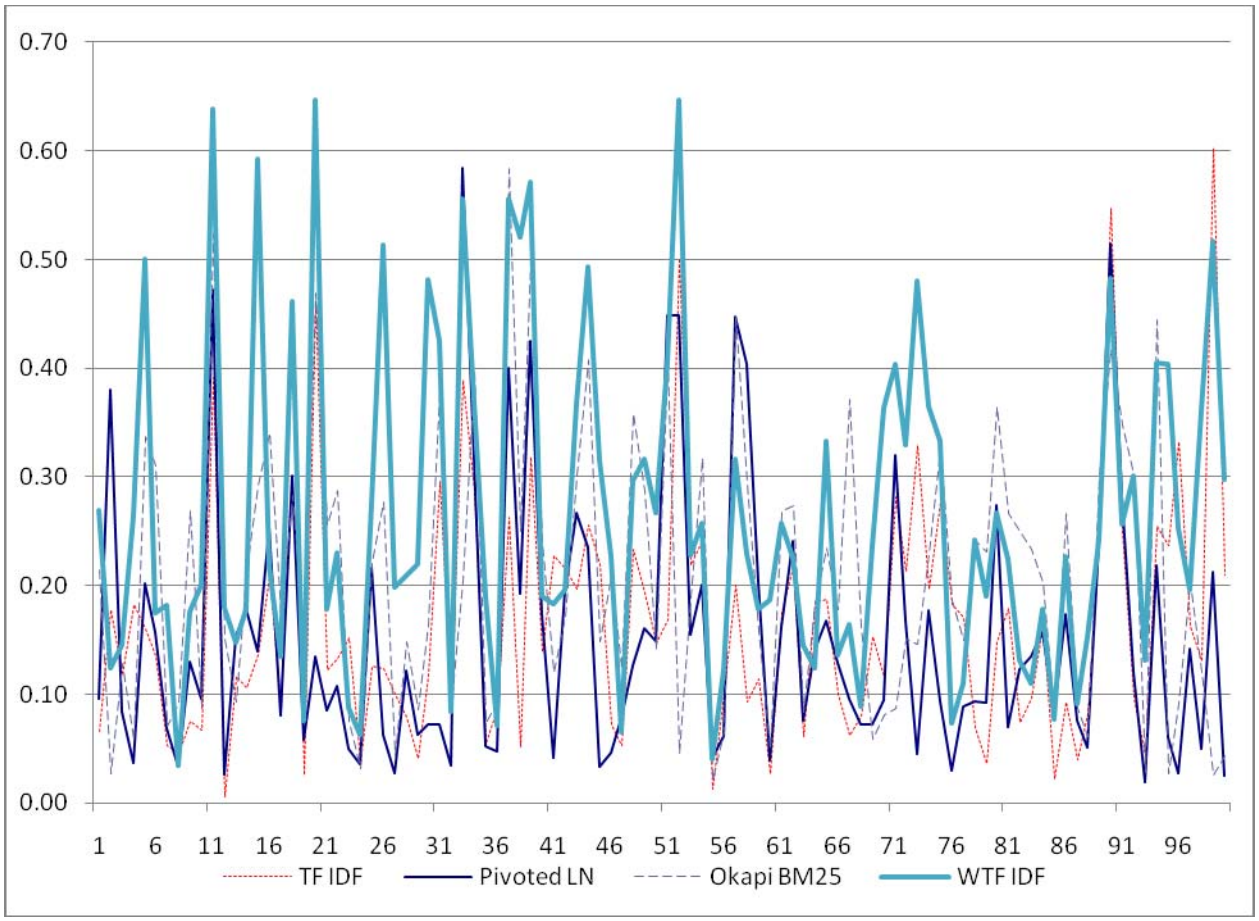

Fig. 4.1 Comparison of Average Precision of 100 cfqueries

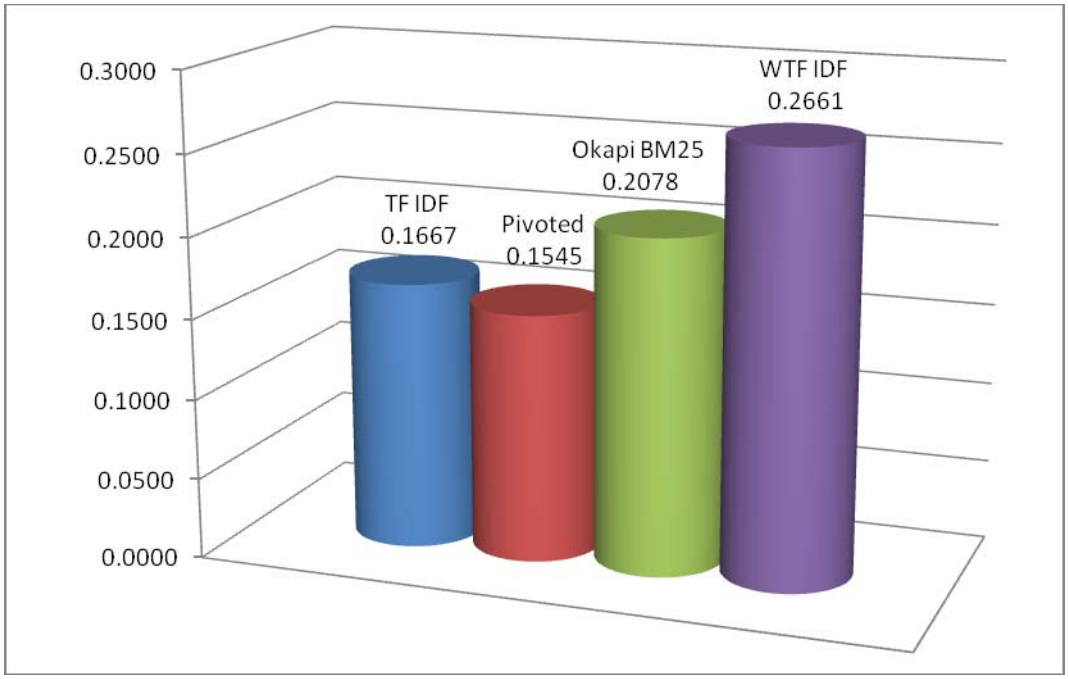

Fig. 4.2 Comparison of MAP of 100 cfqueries

Number of Queries	TF IDF	Pivoted LN	Okapi BM25	WTF IDF
MAP @25	0.1449	0.1403	0.2091	0.2577
MAP @50	0.1581	0.1572	0.2230	0.2840
MAP @75	0.1631	0.1594	0.2104	0.2731
MAP @100	**0.1667**	**0.1545**	**0.2078**	**0.2661**

Table 4.3 Comparison of MAP @ 25, 50, 75 & 100 cfqueries

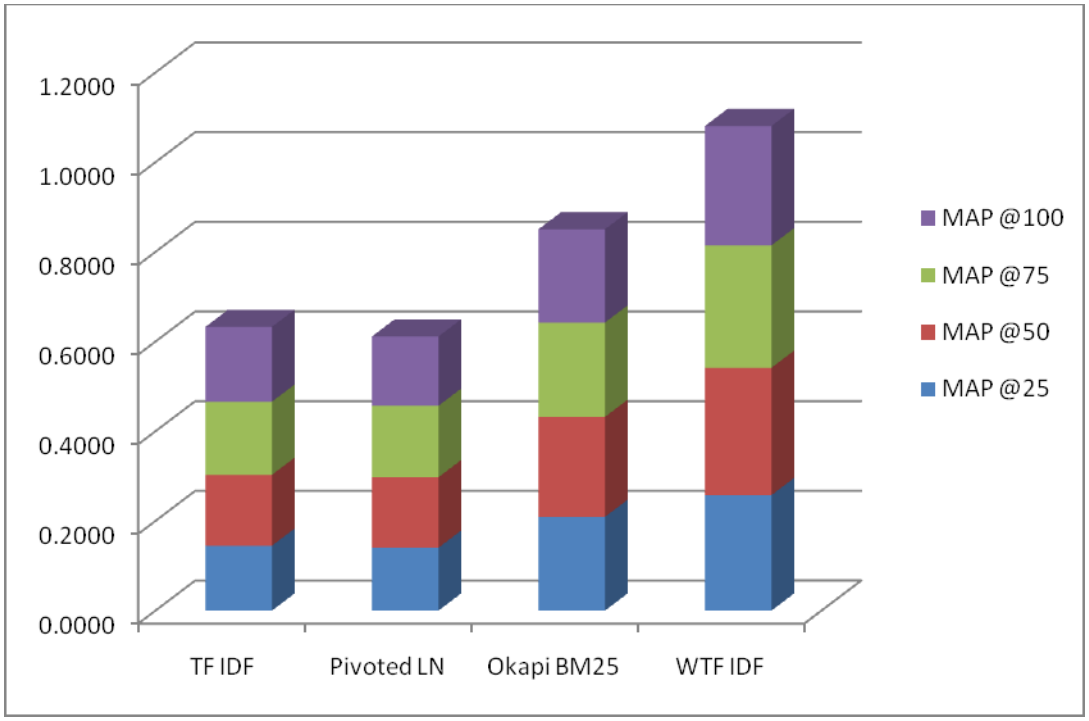

Fig. 4.3 Comparison of MAP of 100 cfqueries

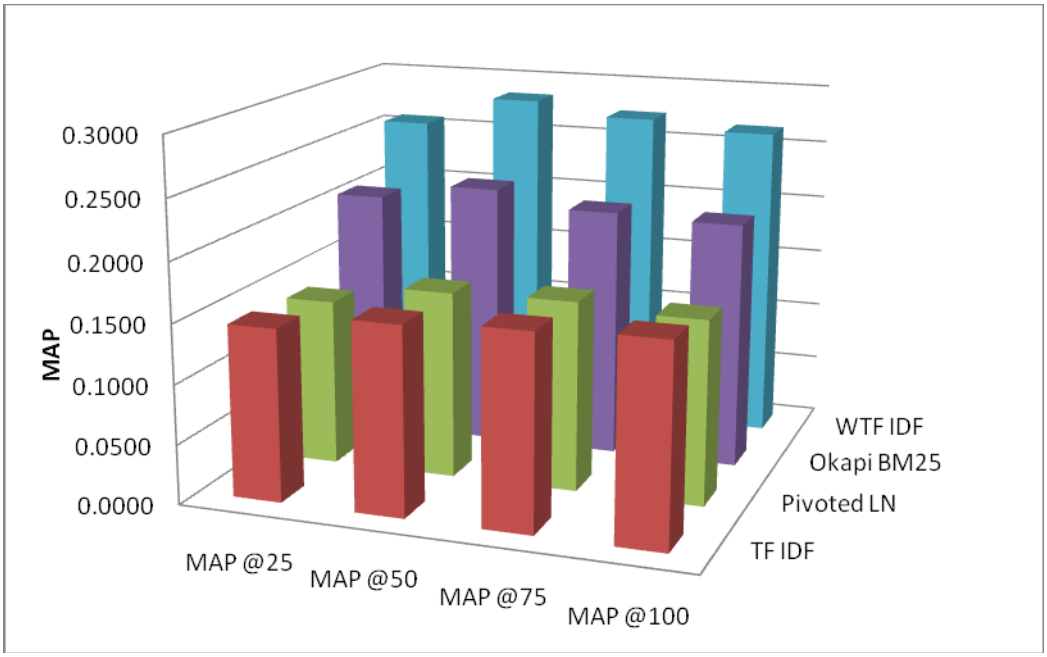

Fig. 4.4 Comparison of MAP of 100 cfqueries

4.2.2 Precision

Table 4.4 presents the better results for proposed approach WTF IDF in comparison of Precision at all documents retrieved and at 1st 10 documents retrieved and pictorially represented in Fig 4.5 and Fig 4.6.

	Precision				Precision @ 10			
Query No	TF IDF	Pivoted LN	Okapi BM25	WTF IDF	TF IDF	Pivoted LN	Okapi BM25	Weighted TF IDF
1	0.0276	0.0308	0.0308	0.0284	0.4000	0.1000	0.5000	0.4000
2	0.0056	0.0091	0.0091	0.0103	0.1000	0.1000	0.5000	0.1000
3	0.0375	0.0390	0.0390	0.0348	0.2000	0.1000	0.3000	0.3000
4	0.0073	0.0159	0.0159	0.0097	0.3000	0.1000	0.3000	0.2000
5	0.1057	0.1556	0.1556	0.2411	0.5000	0.3000	0.6000	0.8000
6	0.0194	0.0231	0.0231	0.0200	0.3000	0.3000	0.4000	0.2000
7	0.0226	0.0236	0.0236	0.0227	0.2000	0.2000	0.2000	0.4000
8	0.0238	0.0240	0.0240	0.0186	0.2000	0.2000	0.2000	0.4000
9	0.0081	0.0112	0.0112	0.0121	0.1000	0.2000	0.2000	0.2000
10	0.0202	0.0246	0.0246	0.0203	0.1000	0.1000	0.3000	0.4000
11	0.0178	0.0390	0.0390	0.0286	0.5000	0.5000	0.5000	0.8000
12	0.0057	0.0058	0.0058	0.0057	0.5000	0.5000	0.1000	0.2000
13	0.0194	0.0214	0.0214	0.0197	0.2000	0.2000	0.2000	0.3000
14	0.0444	0.0475	0.0475	0.0448	0.3000	0.5000	0.5000	0.4000
15	0.0839	0.1260	0.1260	0.3094	0.1000	0.3000	0.6000	0.6000
16	0.1561	0.1724	0.1724	0.1505	0.4000	0.5000	0.8000	0.7000
17	0.0444	0.0462	0.0462	0.0448	0.1000	0.1000	0.4000	0.2000
18	0.0169	0.0218	0.0218	0.0191	0.7000	0.5000	0.5000	0.5000
19	0.0178	0.0209	0.0209	0.0224	0.7000	0.5000	0.2000	0.5000
20	0.0371	0.0416	0.0416	0.0371	0.9000	0.2000	0.9000	1.0000
21	0.0202	0.0209	0.0209	0.0203	0.3000	0.2000	0.4000	0.4000
22	0.0565	0.0833	0.0833	0.0570	0.4000	0.2000	0.7000	0.6000

23	0.0282	0.0292	0.0292	0.0279	0.4000	0.2000	0.2000	0.2000
24	0.0250	0.0270	0.0270	0.0282	0.1000	0.2000	0.2000	0.1000
25	0.0412	0.0701	0.0701	0.0420	0.4000	0.3000	0.3000	0.5000
26	0.0266	0.0497	0.0497	0.1250	0.3000	0.3000	0.3000	0.7000
27	0.0089	0.0125	0.0125	0.0089	0.3000	0.3000	0.1000	0.2000
28	0.0412	0.0524	0.0524	0.0630	0.2000	0.4000	0.4000	0.3000
29	0.0355	0.0526	0.0526	0.0621	0.2000	0.1000	0.2000	0.4000
30	0.0161	0.0186	0.0186	0.0136	0.2000	0.1000	0.2000	0.5000
31	0.0468	0.0505	0.0505	0.0567	0.6000	0.1000	0.8000	0.9000
32	0.0242	0.0280	0.0280	0.0248	0.2000	0.1000	0.1000	0.2000
33	0.0464	0.0664	0.0664	0.0468	0.8000	1.0000	0.5000	1.0000
34	0.0338	0.0423	0.0423	0.0316	0.5000	0.4000	0.5000	0.6000
35	0.0108	0.0107	0.0107	0.0113	0.5000	0.1000	0.2000	0.2000
36	0.0073	0.0090	0.0090	0.0078	0.2000	0.1000	0.1000	0.2000
37	0.0823	0.0931	0.0931	0.0795	0.6000	0.7000	1.0000	1.0000
38	0.0480	0.0749	0.0749	0.1229	0.6000	0.4000	0.4000	0.8000
39	0.1540	0.2699	0.2699	0.5307	0.5000	0.8000	0.7000	0.7000
40	0.1229	0.1258	0.1258	0.1174	0.1000	0.8000	0.6000	0.2000
41	0.0121	0.0143	0.0143	0.0121	0.4000	0.8000	0.1000	0.3000
42	0.0379	0.0414	0.0414	0.0383	0.8000	0.7000	0.4000	0.6000
43	0.0791	0.1346	0.1346	0.1909	0.6000	0.5000	0.3000	0.3000
44	0.1135	0.1309	0.1309	0.1399	0.3000	0.2000	1.0000	0.7000
45	0.0162	0.0200	0.0200	0.0230	0.5000	0.2000	0.3000	0.4000
46	0.0158	0.0140	0.0140	0.0144	0.1000	0.2000	0.3000	0.2000
47	0.0484	0.0524	0.0524	0.0481	0.1000	0.2000	0.3000	0.1000
48	0.0243	0.0287	0.0287	0.0243	0.3000	0.2000	0.7000	0.6000
49	0.0291	0.0336	0.0336	0.0291	0.6000	0.4000	0.6000	0.7000
50	0.0202	0.0231	0.0231	0.0232	0.4000	0.2000	0.2000	0.4000
51	0.1945	0.1945	0.1945	0.1945	0.1000	0.8000	0.9000	0.9000
52	0.0016	0.0021	0.0021	0.0013	0.1000	0.8000	0.9000	0.1000

53	0.0187	0.0233	0.0233	0.0228	0.4000	0.2000	0.3000	0.3000
54	0.0500	0.0549	0.0549	0.0581	0.8000	0.5000	0.7000	0.6000
55	0.0194	0.0223	0.0223	0.0175	0.8000	0.5000	0.7000	0.6000
56	0.0258	0.0317	0.0317	0.0325	0.2000	0.1000	0.1000	0.3000
57	0.0412	0.0529	0.0529	0.0397	0.6000	0.6000	0.5000	0.5000
58	0.0767	0.0929	0.0929	0.0862	0.1000	0.8000	0.5000	0.5000
59	0.1171	0.1318	0.1318	0.1392	0.1000	0.1000	0.1000	0.5000
60	0.0276	0.0449	0.0449	0.0633	0.1000	0.1000	0.1000	0.3000
61	0.0565	0.0608	0.0608	0.0646	0.7000	0.3000	0.7000	0.7000
62	0.1162	0.1250	0.1250	0.1280	0.5000	0.6000	0.6000	0.4000
63	0.0315	0.0330	0.0330	0.0387	0.1000	0.1000	0.6000	0.3000
64	0.0404	0.0467	0.0467	0.0400	0.5000	0.2000	0.4000	0.3000
65	0.0759	0.0892	0.0892	0.0760	0.2000	0.3000	0.3000	0.8000
66	0.0508	0.0549	0.0549	0.0636	0.1000	0.4000	0.4000	0.3000
67	0.0113	0.0151	0.0151	0.0123	0.1000	0.1000	0.4000	0.3000
68	0.0363	0.0456	0.0456	0.0457	0.3000	0.2000	0.4000	0.2000
69	0.0121	0.0142	0.0142	0.0122	0.3000	0.2000	0.1000	0.3000
70	0.0137	0.0176	0.0176	0.0161	0.2000	0.1000	0.1000	0.6000
71	0.0048	0.0048	0.0048	0.0048	0.2000	0.2000	0.1000	0.2000
72	0.0105	0.0133	0.0133	0.0105	0.4000	0.3000	0.3000	0.5000
73	0.0145	0.0173	0.0173	0.0174	0.6000	0.1000	0.2000	0.5000
74	0.0057	0.0058	0.0058	0.0052	0.2000	0.1000	0.2000	0.2000
75	0.0169	0.0193	0.0193	0.0166	0.5000	0.1000	0.5000	0.6000
76	0.0137	0.0135	0.0135	0.0139	0.2000	0.1000	0.2000	0.1000
77	0.0654	0.0707	0.0707	0.0757	0.4000	0.1000	0.3000	0.1000
78	0.0621	0.0728	0.0728	0.0728	0.1000	0.1000	0.7000	0.7000
79	0.0347	0.0403	0.0403	0.0350	0.1000	0.2000	0.6000	0.5000
80	0.0274	0.0672	0.0672	0.0281	0.4000	0.5000	0.7000	0.3000
81	0.0226	0.0281	0.0281	0.0260	0.3000	0.2000	0.7000	0.5000
82	0.0500	0.0535	0.0535	0.0502	0.3000	0.2000	0.6000	0.3000

83	0.0258	0.0286	0.0286	0.0252	0.3000	0.4000	0.4000	0.2000
84	0.0129	0.0159	0.0159	0.0134	0.3000	0.2000	0.3000	0.3000
85	0.0250	0.0268	0.0268	0.0294	0.3000	0.2000	0.1000	0.2000
86	0.0379	0.0418	0.0418	0.0379	0.5000	0.1000	0.5000	0.5000
87	0.0307	0.0340	0.0340	0.0309	0.5000	0.1000	0.1000	0.3000
88	0.0113	0.0125	0.0125	0.0114	0.1000	0.1000	0.1000	0.2000
89	0.0137	0.0170	0.0170	0.0137	0.5000	0.4000	0.3000	0.4000
90	0.0226	0.0264	0.0264	0.0241	1.0000	0.8000	0.6000	0.6000
91	0.2455	0.2607	0.2607	0.2485	0.3000	0.5000	0.7000	0.2000
92	0.0872	0.1026	0.1026	0.0911	0.2000	0.5000	0.9000	0.8000
93	0.0157	0.0167	0.0167	0.0146	0.1000	0.5000	0.1000	0.3000
94	0.0375	0.0810	0.0810	0.0700	0.5000	0.5000	0.6000	0.5000
95	0.0073	0.0069	0.0069	0.0060	0.2000	0.1000	0.6000	0.2000
96	0.0097	0.0098	0.0098	0.0092	0.4000	0.1000	0.2000	0.2000
97	0.0089	0.0104	0.0104	0.0089	0.4000	0.1000	0.3000	0.3000
98	0.0122	0.0184	0.0184	0.0162	0.3000	0.1000	0.2000	0.4000
99	0.0040	0.0052	0.0052	0.0051	0.3000	0.1000	0.2000	0.2000
100	0.0089	0.0102	0.0102	0.0088	0.2000	0.1000	0.1000	0.2000
Average	**0.0396**	**0.0486**	**0.0486**	**0.0529**	**0.3470**	**0.3000**	**0.4050**	**0.4190**

Table 4.4 Comparison of Precision and Precision at R=10 of 100 cfqueries

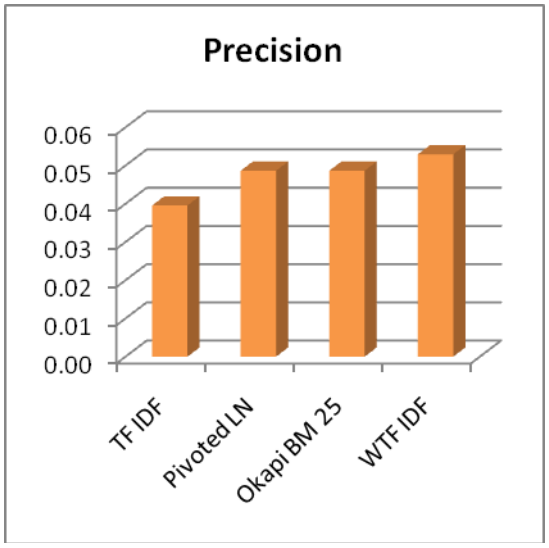

Fig. 4.5 Comparison of Avg. of Precision

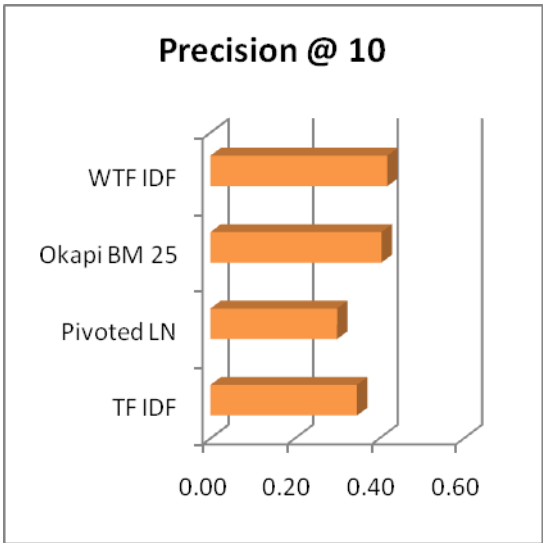

Fig. 4.6 Comparison of Avg. of Precision @10

4.2.3 Recall

Table 4.5 presents the better results for proposed approach WTF IDF in comparison of Recall at all documents retrieved and at 1st 10 documents retrieved and pictorially represented in Fig 4.7 and Fig 4.8.

	Recall				Recall @ 10			
Query No	TF IDF	Pivoted LN	Okapi BM25	Weighted TF IDF	TF IDF	Pivoted LN	Okapi BM25	Weighted TF IDF
1	1.0000	0.9118	0.9118	0.7059	0.3636	0.0294	0.1471	0.1176
2	1.0000	1.0000	1.0000	1.0000	0.1429	0.0294	0.1471	0.1429
3	0.9767	0.9767	0.9767	1.0000	0.0465	0.0233	0.0698	0.0698
4	1.0000	1.0000	1.0000	0.8889	0.3333	0.0233	0.0698	0.2222
5	1.0000	0.6412	0.6412	0.4122	0.0382	0.0229	0.0458	0.0611
6	1.0000	1.0000	1.0000	1.0000	0.1250	0.1250	0.1667	0.0833
7	1.0000	0.9286	0.9286	1.0000	0.0714	0.0714	0.0714	0.1429
8	1.0000	0.9091	0.9091	1.0000	0.0714	0.0714	0.0909	0.1429
9	1.0000	1.0000	1.0000	0.8000	0.1000	0.2000	0.2000	0.2000
10	1.0000	0.9600	0.9600	1.0000	0.0400	0.0400	0.1200	0.1600

11	1.0000	0.9091	0.9091	0.9545	0.2273	0.2273	0.2273	0.3636
12	1.0000	0.8571	0.8571	1.0000	0.2273	0.2273	0.1429	0.2857
13	1.0000	0.9167	0.9167	0.7083	0.0833	0.0833	0.0833	0.1250
14	1.0000	0.9636	0.9636	1.0000	0.0545	0.0909	0.0909	0.0727
15	0.9904	0.5962	0.5962	0.4135	0.0096	0.0288	0.0577	0.0577
16	0.9722	0.8667	0.8667	0.9111	0.0222	0.0278	0.0444	0.0389
17	1.0000	0.9818	0.9818	1.0000	0.0182	0.0182	0.0727	0.0364
18	1.0000	1.0000	1.0000	0.7619	0.3333	0.2381	0.2381	0.2381
19	1.0000	1.0000	1.0000	0.9091	0.3333	0.2381	0.0909	0.2381
20	1.0000	0.9348	0.9348	1.0000	0.1957	0.0435	0.1957	0.2174
21	1.0000	0.8800	0.8800	1.0000	0.1200	0.0800	0.1600	0.1600
22	1.0000	0.6000	0.6000	1.0000	0.0571	0.0286	0.1000	0.0857
23	1.0000	0.7429	0.7429	0.9714	0.1143	0.0286	0.0571	0.0571
24	1.0000	0.8387	0.8387	0.7097	0.0323	0.0286	0.0571	0.0323
25	1.0000	0.9608	0.9608	1.0000	0.0784	0.0588	0.0588	0.0980
26	1.0000	0.7273	0.7273	0.6364	0.0909	0.0588	0.0909	0.2121
27	1.0000	0.8182	0.8182	1.0000	0.2727	0.0588	0.0909	0.1818
28	1.0000	0.8235	0.8235	0.6667	0.0392	0.0784	0.0784	0.0588
29	1.0000	0.9545	0.9545	0.7500	0.0392	0.0227	0.0455	0.0909
30	1.0000	0.9412	0.9412	0.9412	0.1176	0.0588	0.1176	0.2941
31	1.0000	0.8793	0.8793	0.8276	0.1034	0.0588	0.1379	0.1552
32	1.0000	0.9333	0.9333	1.0000	0.0667	0.0588	0.0333	0.0667
33	1.0000	0.8772	0.8772	1.0000	0.1404	0.1754	0.0877	0.1754
34	0.9487	0.9487	0.9487	1.0000	0.1282	0.1026	0.1282	0.1538
35	0.9286	0.8571	0.8571	1.0000	0.1282	0.0714	0.1429	0.1429
36	1.0000	1.0000	1.0000	1.0000	0.2222	0.1111	0.1111	0.1429
37	0.9898	0.9694	0.9694	1.0000	0.0612	0.0714	0.1020	0.1020
38	0.7547	0.7925	0.7925	0.5472	0.0612	0.0755	0.0755	0.1509
39	0.8272	0.8148	0.8148	0.5864	0.0309	0.0494	0.0432	0.0432
40	0.9384	0.9178	0.9178	0.9932	0.0068	0.0494	0.0411	0.0137

41	1.0000	1.0000	1.0000	1.0000	0.2667	0.0494	0.0667	0.2000
42	1.0000	0.9149	0.9149	1.0000	0.1702	0.1489	0.0851	0.1277
43	1.0000	0.8571	0.8571	0.6429	0.0612	0.0510	0.0306	0.0306
44	1.0000	0.9643	0.9643	0.8786	0.0214	0.0143	0.0714	0.0500
45	1.0000	1.0000	1.0000	0.9500	0.2500	0.0143	0.1500	0.2000
46	1.0000	0.7778	0.7778	0.6667	0.0556	0.0143	0.1667	0.1111
47	1.0000	0.9667	0.9667	0.9833	0.0167	0.0333	0.0500	0.0167
48	1.0000	1.0000	1.0000	1.0000	0.1000	0.0667	0.2333	0.2000
49	1.0000	0.9722	0.9722	1.0000	0.1667	0.1111	0.1667	0.1944
50	1.0000	0.9200	0.9200	0.7600	0.1600	0.0800	0.0800	0.1600
51	1.0000	1.0000	1.0000	1.0000	0.0041	0.0332	0.0373	0.0373
52	1.0000	1.0000	1.0000	0.5000	0.5000	0.0332	0.0373	0.5000
53	1.0000	1.0000	1.0000	0.9565	0.1739	0.0870	0.1304	0.1304
54	1.0000	0.9839	0.9839	0.8871	0.1290	0.0806	0.1129	0.0968
55	1.0000	0.9167	0.9167	0.6250	0.1290	0.0806	0.1129	0.0968
56	1.0000	1.0000	1.0000	0.9375	0.0625	0.0313	0.0313	0.0938
57	1.0000	1.0000	1.0000	0.9608	0.1176	0.1176	0.0980	0.0980
58	1.0000	0.9789	0.9789	0.9158	0.0105	0.0842	0.0526	0.0526
59	1.0000	0.8828	0.8828	0.7586	0.0105	0.0069	0.0069	0.0345
60	1.0000	0.8235	0.8235	0.6176	0.0105	0.0069	0.0069	0.0882
61	1.0000	0.8714	0.8714	0.7714	0.1000	0.0429	0.1000	0.1000
62	1.0000	0.9792	0.9792	0.8611	0.0347	0.0417	0.0417	0.0278
63	1.0000	0.8205	0.8205	0.8462	0.0256	0.0256	0.0417	0.0769
64	1.0000	0.9800	0.9800	0.9800	0.1000	0.0400	0.0800	0.0600
65	1.0000	0.9681	0.9681	0.9894	0.0213	0.0319	0.0319	0.0851
66	1.0000	0.9683	0.9683	0.9524	0.0159	0.0635	0.0635	0.0476
67	1.0000	0.9286	0.9286	0.9286	0.0714	0.0714	0.2857	0.2143
68	1.0000	0.9778	0.9778	0.8000	0.0667	0.0444	0.0889	0.0444
69	1.0000	0.9333	0.9333	1.0000	0.2000	0.1333	0.0667	0.2000
70	1.0000	1.0000	1.0000	1.0000	0.1176	0.0588	0.0588	0.3529

71	1.0000	1.0000	1.0000	1.0000	0.3333	0.3333	0.1667	0.3333
72	1.0000	1.0000	1.0000	1.0000	0.3077	0.2308	0.2308	0.3846
73	1.0000	0.9444	0.9444	0.7778	0.3333	0.0556	0.1111	0.2778
74	1.0000	0.8571	0.8571	0.7143	0.2857	0.1429	0.2857	0.2857
75	1.0000	0.9524	0.9524	0.9524	0.2381	0.0476	0.2381	0.2857
76	1.0000	0.8235	0.8235	1.0000	0.1176	0.0476	0.1176	0.0588
77	0.9878	0.8902	0.8902	0.8049	0.0488	0.0122	0.0366	0.0122
78	1.0000	1.0000	1.0000	1.0000	0.0130	0.0122	0.0909	0.0909
79	1.0000	0.9302	0.9302	1.0000	0.0130	0.0465	0.1395	0.1163
80	1.0000	0.7941	0.7941	1.0000	0.1176	0.1471	0.2059	0.0882
81	1.0000	1.0000	1.0000	1.0000	0.1071	0.0714	0.2500	0.1786
82	1.0000	0.9677	0.9677	1.0000	0.0484	0.0323	0.0968	0.0484
83	1.0000	1.0000	1.0000	0.9688	0.0938	0.1250	0.1250	0.0625
84	1.0000	0.9375	0.9375	0.6250	0.1875	0.1250	0.1875	0.1875
85	1.0000	0.9032	0.9032	0.9032	0.1875	0.0645	0.0323	0.0645
86	1.0000	1.0000	1.0000	1.0000	0.1064	0.0213	0.1064	0.1064
87	1.0000	1.0000	1.0000	1.0000	0.1064	0.0263	0.0263	0.0789
88	1.0000	1.0000	1.0000	1.0000	0.0714	0.0714	0.0714	0.1429
89	1.0000	1.0000	1.0000	1.0000	0.2941	0.2353	0.1765	0.2353
90	1.0000	0.9643	0.9643	0.7500	0.3571	0.2857	0.2143	0.2143
91	0.9934	0.7829	0.7829	0.9671	0.0099	0.0164	0.0230	0.0066
92	1.0000	0.8889	0.8889	0.8796	0.0185	0.0164	0.0833	0.0741
93	0.9444	0.9444	0.9444	1.0000	0.0556	0.0164	0.0556	0.1667
94	0.9767	0.8837	0.8837	0.7209	0.1163	0.1163	0.1395	0.1163
95	1.0000	0.7778	0.7778	0.5556	0.2222	0.1111	0.1395	0.2222
96	1.0000	0.9167	0.9167	0.7500	0.3333	0.1111	0.1667	0.1667
97	1.0000	1.0000	1.0000	1.0000	0.3636	0.0909	0.2727	0.2727
98	1.0000	0.8667	0.8667	0.8667	0.2000	0.0667	0.1333	0.2667
99	1.0000	1.0000	1.0000	0.8000	0.6000	0.2000	0.1333	0.4000
100	1.0000	0.9091	0.9091	0.6364	0.1818	0.2000	0.0909	0.1818

Average	0.9923	0.9195	0.9195	0.8794	0.1339	0.0796	0.1087	0.1429

Table 4.5 Comparison of Recall and Recall at 10 of 100 cfqueries

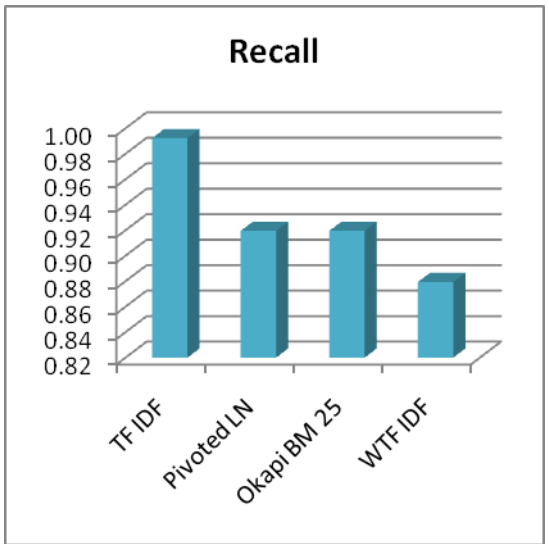

Fig. 4.7 Comparison of Avg. of Recall

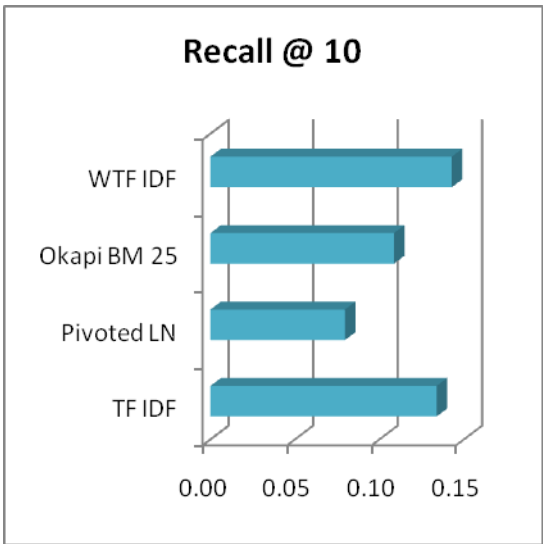

Fig. 4.8 Comparison of Avg. of Recall @10

4.2.4 F Measure

Table 4.6 presents the better results for proposed approach WTF IDF in comparison of F-Measure at all documents retrieved and at 1st 10 documents retrieved and pictorially represented in Fig 4.9 and Fig 4.10.

	F Measure				F Measure @ 10			
Query No	TF IDF	Pivoted LN	Okapi BM25	Weighted TF IDF	TF IDF	Pivoted LN	Okapi BM25	Weighted TF IDF
1	0.0536	0.0595	0.0595	0.0547	0.3810	0.0455	0.2273	0.1818
2	0.0112	0.0180	0.0180	0.0203	0.1176	0.0455	0.2273	0.1176
3	0.0722	0.0751	0.0751	0.0673	0.0755	0.0377	0.1132	0.1132
4	0.0144	0.0313	0.0313	0.0192	0.3158	0.0377	0.1132	0.2105
5	0.1912	0.2504	0.2504	0.3042	0.0709	0.0426	0.0851	0.1135
6	0.0380	0.0451	0.0451	0.0393	0.1765	0.1765	0.2353	0.1176
7	0.0442	0.0460	0.0460	0.0443	0.1053	0.1053	0.1053	0.2105

8	0.0465	0.0468	0.0468	0.0365	0.1053	0.1053	0.1250	0.2105
9	0.0160	0.0221	0.0221	0.0239	0.1000	0.2000	0.2000	0.2000
10	0.0396	0.0479	0.0479	0.0397	0.0571	0.0571	0.1714	0.2286
11	0.0349	0.0748	0.0748	0.0556	0.3125	0.3125	0.3125	0.5000
12	0.0112	0.0115	0.0115	0.0113	0.3125	0.3125	0.1176	0.2353
13	0.0380	0.0419	0.0419	0.0383	0.1176	0.1176	0.1176	0.1765
14	0.0851	0.0906	0.0906	0.0857	0.0923	0.1538	0.1538	0.1231
15	0.1548	0.2081	0.2081	0.3539	0.0175	0.0526	0.1053	0.1053
16	0.2690	0.2876	0.2876	0.2583	0.0421	0.0526	0.0842	0.0737
17	0.0850	0.0882	0.0882	0.0857	0.0308	0.0308	0.1231	0.0615
18	0.0333	0.0427	0.0427	0.0373	0.4516	0.3226	0.3226	0.3226
19	0.0349	0.0409	0.0409	0.0437	0.4516	0.3226	0.1250	0.3226
20	0.0716	0.0796	0.0796	0.0716	0.3214	0.0714	0.3214	0.3571
21	0.0396	0.0407	0.0407	0.0397	0.1714	0.1143	0.2286	0.2286
22	0.1070	0.1463	0.1463	0.1079	0.1000	0.0500	0.1750	0.1500
23	0.0549	0.0562	0.0562	0.0542	0.1778	0.0500	0.0889	0.0889
24	0.0488	0.0524	0.0524	0.0543	0.0488	0.0500	0.0889	0.0488
25	0.0791	0.1307	0.1307	0.0807	0.1311	0.0984	0.0984	0.1639
26	0.0519	0.0930	0.0930	0.2090	0.1395	0.0984	0.1395	0.3256
27	0.0176	0.0246	0.0246	0.0176	0.2857	0.0984	0.0952	0.1905
28	0.0791	0.0986	0.0986	0.1151	0.0656	0.1311	0.1311	0.0984
29	0.0686	0.0998	0.0998	0.1148	0.0656	0.0370	0.0741	0.1481
30	0.0316	0.0365	0.0365	0.0268	0.1481	0.0741	0.1481	0.3704
31	0.0895	0.0956	0.0956	0.1061	0.1765	0.0741	0.2353	0.2647
32	0.0473	0.0544	0.0544	0.0484	0.1000	0.0741	0.0500	0.1000
33	0.0887	0.1235	0.1235	0.0893	0.2388	0.2985	0.1493	0.2985
34	0.0652	0.0811	0.0811	0.0612	0.2041	0.1633	0.2041	0.2449
35	0.0214	0.0212	0.0212	0.0224	0.2041	0.0833	0.1667	0.1667
36	0.0144	0.0179	0.0179	0.0154	0.2105	0.1053	0.1053	0.1667
37	0.1520	0.1699	0.1699	0.1474	0.1111	0.1296	0.1852	0.1852

38	0.0902	0.1368	0.1368	0.2007	0.1111	0.1270	0.1270	0.2540
39	0.2597	0.4055	0.4055	0.5572	0.0581	0.0930	0.0814	0.0814
40	0.2173	0.2213	0.2213	0.2100	0.0128	0.0930	0.0769	0.0256
41	0.0239	0.0283	0.0283	0.0240	0.3200	0.0930	0.0800	0.2400
42	0.0731	0.0793	0.0793	0.0738	0.2807	0.2456	0.1404	0.2105
43	0.1466	0.2327	0.2327	0.2944	0.1111	0.0926	0.0556	0.0556
44	0.2039	0.2306	0.2306	0.2414	0.0400	0.0267	0.1333	0.0933
45	0.0318	0.0393	0.0393	0.0449	0.3333	0.0267	0.2000	0.2667
46	0.0312	0.0275	0.0275	0.0281	0.0714	0.0267	0.2143	0.1429
47	0.0924	0.0994	0.0994	0.0917	0.0286	0.0571	0.0857	0.0286
48	0.0474	0.0558	0.0558	0.0474	0.1500	0.1000	0.3500	0.3000
49	0.0565	0.0649	0.0649	0.0566	0.2609	0.1739	0.2609	0.3043
50	0.0396	0.0451	0.0451	0.0451	0.2286	0.1143	0.1143	0.2286
51	0.3257	0.3257	0.3257	0.3257	0.0080	0.0637	0.0717	0.0717
52	0.0032	0.0042	0.0042	0.0026	0.1667	0.0637	0.0717	0.1667
53	0.0366	0.0455	0.0455	0.0445	0.2424	0.1212	0.1818	0.1818
54	0.0953	0.1040	0.1040	0.1091	0.2222	0.1389	0.1944	0.1667
55	0.0380	0.0435	0.0435	0.0340	0.2222	0.1389	0.1944	0.1667
56	0.0504	0.0615	0.0615	0.0628	0.0952	0.0476	0.0476	0.1429
57	0.0791	0.1005	0.1005	0.0763	0.1967	0.1967	0.1639	0.1639
58	0.1424	0.1697	0.1697	0.1576	0.0190	0.1524	0.0952	0.0952
59	0.2097	0.2294	0.2294	0.2353	0.0190	0.0129	0.0129	0.0645
60	0.0538	0.0851	0.0851	0.1148	0.0190	0.0129	0.0129	0.1364
61	0.1070	0.1137	0.1137	0.1192	0.1750	0.0750	0.1750	0.1750
62	0.2082	0.2217	0.2217	0.2228	0.0649	0.0779	0.0779	0.0519
63	0.0610	0.0635	0.0635	0.0740	0.0408	0.0408	0.0779	0.1224
64	0.0776	0.0892	0.0892	0.0769	0.1667	0.0667	0.1333	0.1000
65	0.1410	0.1634	0.1634	0.1411	0.0385	0.0577	0.0577	0.1538
66	0.0968	0.1039	0.1039	0.1192	0.0274	0.1096	0.1096	0.0822
67	0.0223	0.0297	0.0297	0.0242	0.0833	0.0833	0.3333	0.2500

68	0.0701	0.0872	0.0872	0.0865	0.1091	0.0727	0.1455	0.0727
69	0.0239	0.0280	0.0280	0.0240	0.2400	0.1600	0.0800	0.2400
70	0.0271	0.0346	0.0346	0.0317	0.1481	0.0741	0.0741	0.4444
71	0.0096	0.0096	0.0096	0.0096	0.2500	0.2500	0.1250	0.2500
72	0.0208	0.0263	0.0263	0.0209	0.3478	0.2609	0.2609	0.4348
73	0.0286	0.0341	0.0341	0.0340	0.4286	0.0714	0.1429	0.3571
74	0.0112	0.0116	0.0116	0.0103	0.2353	0.1176	0.2353	0.2353
75	0.0333	0.0378	0.0378	0.0326	0.3226	0.0645	0.3226	0.3871
76	0.0271	0.0266	0.0266	0.0274	0.1481	0.0645	0.1481	0.0741
77	0.1227	0.1309	0.1309	0.1384	0.0870	0.0217	0.0652	0.0217
78	0.1170	0.1358	0.1358	0.1358	0.0230	0.0217	0.1609	0.1609
79	0.0671	0.0772	0.0772	0.0676	0.0230	0.0755	0.2264	0.1887
80	0.0534	0.1239	0.1239	0.0547	0.1818	0.2273	0.3182	0.1364
81	0.0442	0.0546	0.0546	0.0506	0.1579	0.1053	0.3684	0.2632
82	0.0953	0.1014	0.1014	0.0955	0.0833	0.0556	0.1667	0.0833
83	0.0504	0.0557	0.0557	0.0491	0.1429	0.1905	0.1905	0.0952
84	0.0255	0.0313	0.0313	0.0261	0.2308	0.1538	0.2308	0.2308
85	0.0488	0.0521	0.0521	0.0569	0.2308	0.0976	0.0488	0.0976
86	0.0731	0.0803	0.0803	0.0731	0.1754	0.0351	0.1754	0.1754
87	0.0595	0.0658	0.0658	0.0599	0.1754	0.0417	0.0417	0.1250
88	0.0223	0.0247	0.0247	0.0225	0.0833	0.0833	0.0833	0.1667
89	0.0271	0.0334	0.0334	0.0271	0.3704	0.2963	0.2222	0.2963
90	0.0442	0.0514	0.0514	0.0467	0.5263	0.4211	0.3158	0.3158
91	0.3937	0.3911	0.3911	0.3954	0.0191	0.0318	0.0446	0.0127
92	0.1604	0.1839	0.1839	0.1651	0.0339	0.0318	0.1525	0.1356
93	0.0308	0.0329	0.0329	0.0287	0.0714	0.0318	0.0714	0.2143
94	0.0722	0.1484	0.1484	0.1276	0.1887	0.1887	0.2264	0.1887
95	0.0144	0.0138	0.0138	0.0118	0.2105	0.1053	0.2264	0.2105
96	0.0192	0.0193	0.0193	0.0182	0.3636	0.1053	0.1818	0.1818
97	0.0176	0.0206	0.0206	0.0177	0.3810	0.0952	0.2857	0.2857

98	0.0240	0.0361	0.0361	0.0318	0.2400	0.0800	0.1600	0.3200
99	0.0080	0.0103	0.0103	0.0101	0.4000	0.1333	0.1600	0.2667
100	0.0176	0.0203	0.0203	0.0174	0.1905	0.1333	0.0952	0.1905
Average	**0.0732**	**0.0876**	**0.0876**	**0.0886**	**0.1687**	**0.1096**	**0.1543**	**0.1860**

Table 4.6 Comparison of F-Measure and F-Measure at R=10 of 100 cfqueries

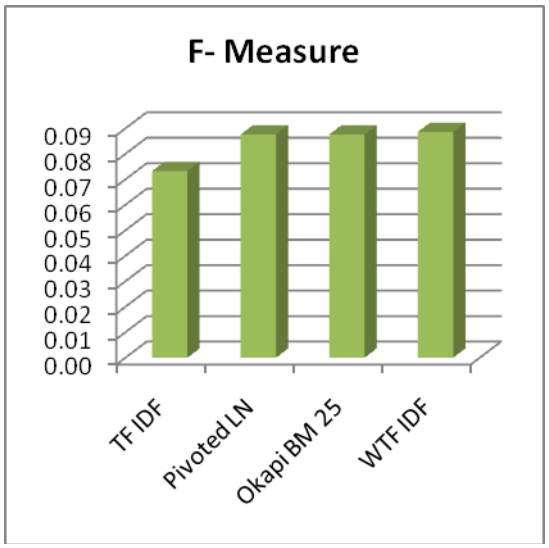

Fig. 4.9 Comparison of Avg. of F-Measure

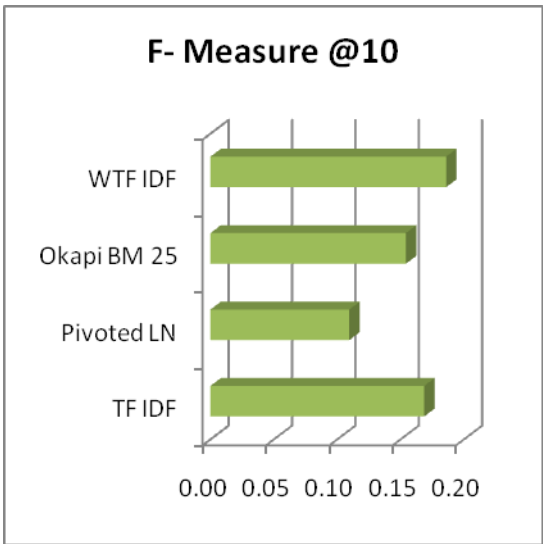

Fig. 4.10 Comparison of Avg. of F-Measure @10

4.2.5 Fallout

Table 4.7 presents the better results for proposed approach WTF IDF in comparison of Fallout at all documents retrieved and at 1st 10 documents retrieved and pictorially represented in Fig 4.11 and Fig 4.12.

	Fallout				Fallout @ 10			
Query No	**TF IDF**	**Pivoted LN**	**Okapi BM25**	**Weighted TF IDF**	**TF IDF**	**Pivoted LN**	**Okapi BM25**	**Weighted TF IDF**
1	0.9959	0.8108	0.8108	0.6805	0.0083	0.0075	0.0041	0.0050
2	1.0000	0.6209	0.6209	0.5471	0.0073	0.0081	0.0081	0.0073
3	0.9022	0.8645	0.8645	0.9967	0.0067	0.0075	0.0059	0.0059
4	1.0000	0.4537	0.4537	0.6642	0.0057	0.0081	0.0081	0.0065

5	1.0000	0.4116	0.4116	0.1534	0.0045	0.0063	0.0036	0.0018
6	1.0000	0.8370	0.8370	0.9663	0.0058	0.0058	0.0049	0.0066
7	1.0000	0.8885	0.8885	0.9967	0.0066	0.0066	0.0066	0.0050
8	0.7420	0.6680	0.6680	0.9540	0.0082	0.0082	0.0066	0.0082
9	1.0000	0.7193	0.7193	0.5297	0.0073	0.0065	0.0065	0.0065
10	1.0000	0.7850	0.7850	0.9959	0.0074	0.0074	0.0058	0.0049
11	1.0000	0.4051	0.4051	0.5850	0.0041	0.0041	0.0041	0.0016
12	0.9992	0.8401	0.8401	0.9935	0.0081	0.0081	0.0073	0.0065
13	1.0000	0.8272	0.8272	0.6963	0.0066	0.0066	0.0066	0.0058
14	0.9992	0.8970	0.8970	0.9907	0.0059	0.0042	0.0042	0.0051
15	0.9903	0.3789	0.3789	0.0846	0.0079	0.0062	0.0035	0.0035
16	0.8933	0.7073	0.7073	0.8744	0.0057	0.0047	0.0019	0.0028
17	1.0000	0.9417	0.9417	0.9916	0.0076	0.0076	0.0051	0.0068
18	1.0000	0.7734	0.7734	0.6749	0.0025	0.0041	0.0041	0.0041
19	0.9992	0.8472	0.8472	0.7173	0.0082	0.0082	0.0066	0.0082
20	1.0000	0.8307	0.8307	1.0000	0.0008	0.0067	0.0008	0.0000
21	1.0000	0.8509	0.8509	0.9959	0.0058	0.0066	0.0049	0.0049
22	1.0000	0.3952	0.3952	0.9906	0.0051	0.0068	0.0026	0.0034
23	1.0000	0.7184	0.7184	0.9842	0.0050	0.0083	0.0066	0.0066
24	1.0000	0.7748	0.7748	0.6275	0.0075	0.0083	0.0083	0.0075
25	1.0000	0.5471	0.5471	0.9781	0.0051	0.0059	0.0059	0.0042
26	1.0000	0.3806	0.3806	0.1219	0.0058	0.0083	0.0058	0.0025
27	1.0000	0.5798	0.5798	0.9992	0.0057	0.0081	0.0073	0.0065
28	1.0000	0.6389	0.6389	0.4259	0.0067	0.0051	0.0051	0.0059
29	1.0000	0.6326	0.6326	0.4167	0.0084	0.0075	0.0067	0.0050
30	0.8519	0.6899	0.6899	0.9493	0.0065	0.0074	0.0065	0.0041
31	0.9992	0.8112	0.8112	0.6765	0.0034	0.0085	0.0017	0.0008
32	0.9992	0.8040	0.8040	0.9752	0.0066	0.0083	0.0074	0.0066
33	0.9907	0.5948	0.5948	0.9831	0.0017	0.0000	0.0042	0.0000
34	0.8825	0.6975	0.6975	0.9967	0.0042	0.0050	0.0042	0.0033

35	0.9698	0.9045	0.9045	0.9976	0.0082	0.0073	0.0065	0.0065
36	1.0000	0.8049	0.8049	0.9341	0.0065	0.0073	0.0073	0.0081
37	0.9474	0.8107	0.8107	0.9939	0.0035	0.0026	0.0000	0.0000
38	0.6695	0.4376	0.4376	0.1745	0.0084	0.0051	0.0051	0.0017
39	0.6834	0.3315	0.3315	0.0780	0.0046	0.0019	0.0028	0.0028
40	0.8948	0.8518	0.8518	0.9973	0.0082	0.0091	0.0037	0.0073
41	1.0000	0.8423	0.8423	0.9967	0.0049	0.0082	0.0074	0.0057
42	1.0000	0.8347	0.8347	0.9891	0.0017	0.0025	0.0050	0.0034
43	1.0000	0.4733	0.4733	0.2340	0.0035	0.0044	0.0061	0.0061
44	0.9945	0.8153	0.8153	0.6879	0.0064	0.0073	0.0000	0.0027
45	0.9992	0.8031	0.8031	0.6628	0.0041	0.0082	0.0057	0.0049
46	0.9156	0.8084	0.8084	0.6740	0.0074	0.0082	0.0057	0.0066
47	1.0000	0.8897	0.8897	0.9907	0.0076	0.0068	0.0059	0.0076
48	0.9967	0.8395	0.8395	0.9967	0.0058	0.0066	0.0025	0.0033
49	1.0000	0.8371	0.8371	0.9967	0.0033	0.0050	0.0033	0.0025
50	1.0000	0.8007	0.8007	0.6582	0.0049	0.0066	0.0066	0.0049
51	1.0000	1.0000	1.0000	1.0000	0.0090	0.0020	0.0010	0.0010
52	1.0000	0.7712	0.7712	0.6249	0.0073	0.0081	0.0081	0.0073
53	0.9951	0.7944	0.7944	0.7763	0.0049	0.0066	0.0058	0.0058
54	1.0000	0.8921	0.8921	0.7570	0.0017	0.0042	0.0025	0.0034
55	1.0000	0.7951	0.7951	0.6938	0.0082	0.0082	0.0082	0.0082
56	1.0000	0.8094	0.8094	0.7407	0.0066	0.0075	0.0075	0.0058
57	1.0000	0.7685	0.7685	0.9975	0.0034	0.0034	0.0042	0.0042
58	1.0000	0.7937	0.7937	0.8059	0.0079	0.0017	0.0044	0.0044
59	0.9991	0.7706	0.7706	0.6216	0.0091	0.0082	0.0082	0.0046
60	0.9934	0.4946	0.4946	0.2581	0.0083	0.0083	0.0083	0.0058
61	1.0000	0.8058	0.8058	0.6689	0.0026	0.0060	0.0026	0.0026
62	1.0000	0.9014	0.9014	0.7717	0.0046	0.0037	0.0037	0.0055
63	1.0000	0.7808	0.7808	0.6833	0.0075	0.0075	0.0083	0.0058
64	0.9992	0.8410	0.8410	0.9882	0.0042	0.0067	0.0050	0.0059

65	1.0000	0.8114	0.8114	0.9878	0.0070	0.0061	0.0061	0.0017
66	1.0000	0.8929	0.8929	0.7517	0.0077	0.0051	0.0051	0.0060
67	1.0000	0.6922	0.6922	0.8539	0.0073	0.0073	0.0049	0.0057
68	0.9992	0.7705	0.7705	0.6290	0.0059	0.0067	0.0050	0.0067
69	1.0000	0.7941	0.7941	0.9959	0.0057	0.0065	0.0074	0.0057
70	1.0000	0.7766	0.7766	0.8502	0.0065	0.0074	0.0074	0.0033
71	1.0000	1.0000	1.0000	1.0000	0.0065	0.0065	0.0073	0.0065
72	1.0000	0.7838	0.7838	0.9959	0.0049	0.0057	0.0057	0.0041
73	1.0000	0.7887	0.7887	0.6478	0.0033	0.0074	0.0066	0.0041
74	0.9992	0.8287	0.8287	0.7808	0.0065	0.0073	0.0065	0.0065
75	1.0000	0.8342	0.8342	0.9745	0.0041	0.0074	0.0041	0.0033
76	1.0000	0.8355	0.8355	0.9877	0.0065	0.0082	0.0065	0.0074
77	1.0000	0.8297	0.8297	0.6966	0.0052	0.0078	0.0061	0.0078
78	1.0000	0.8434	0.8434	0.8434	0.0077	0.0086	0.0026	0.0026
79	1.0000	0.7968	0.7968	0.9916	0.0084	0.0067	0.0033	0.0042
80	1.0000	0.3112	0.3112	0.9759	0.0050	0.0041	0.0025	0.0058
81	0.9992	0.8002	0.8002	0.8671	0.0058	0.0066	0.0025	0.0041
82	1.0000	0.9014	0.9014	0.9975	0.0059	0.0068	0.0034	0.0059
83	1.0000	0.8989	0.8989	0.9942	0.0058	0.0050	0.0050	0.0066
84	1.0000	0.7580	0.7580	0.6043	0.0057	0.0065	0.0057	0.0057
85	1.0000	0.8402	0.8402	0.7657	0.0083	0.0066	0.0075	0.0066
86	1.0000	0.9035	0.9035	1.0000	0.0042	0.0076	0.0042	0.0042
87	1.0000	0.8984	0.8984	0.9933	0.0083	0.0075	0.0075	0.0058
88	1.0000	0.9012	0.9012	0.9951	0.0073	0.0073	0.0073	0.0065
89	1.0000	0.8044	0.8044	1.0000	0.0041	0.0049	0.0057	0.0049
90	0.9992	0.8216	0.8216	0.7027	0.0000	0.0017	0.0033	0.0033
91	0.9925	0.7219	0.7219	0.9508	0.0075	0.0053	0.0032	0.0086
92	1.0000	0.7427	0.7427	0.8382	0.0071	0.0088	0.0009	0.0018
93	0.8747	0.8174	0.8174	0.9967	0.0074	0.0082	0.0074	0.0057
94	0.9013	0.3604	0.3604	0.3445	0.0042	0.0042	0.0033	0.0042

95	1.0000	0.8138	0.8138	0.6789	0.0065	0.0073	0.0081	0.0065
96	1.0000	0.9087	0.9087	0.7889	0.0049	0.0081	0.0065	0.0065
97	1.0000	0.8518	0.8518	0.9967	0.0049	0.0073	0.0057	0.0057
98	0.9951	0.5662	0.5662	0.6454	0.0057	0.0074	0.0065	0.0049
99	0.9992	0.7820	0.7820	0.6353	0.0057	0.0073	0.0081	0.0065
100	1.0000	0.7866	0.7866	0.6393	0.0065	0.0081	0.0073	0.0065
Average	**0.9806**	**0.7540**	**0.7540**	**0.7906**	**0.0059**	**0.0065**	**0.0053**	**0.0050**

Table 4.7 Comparison of Fallout and Fallout at R=10 of 100 cfqueries

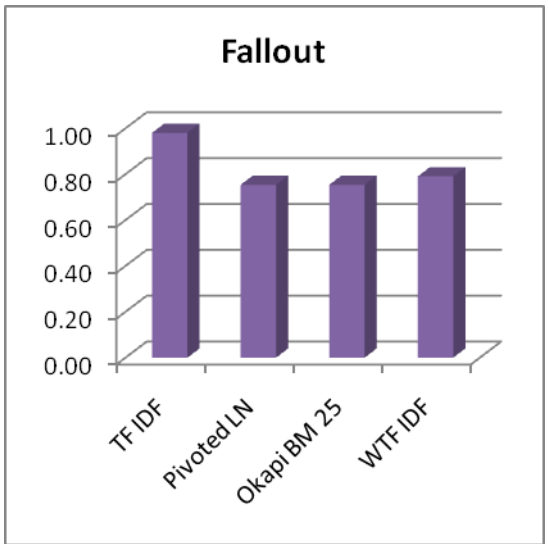

Fig. 4.11 Comparison of Avg. of Fallout

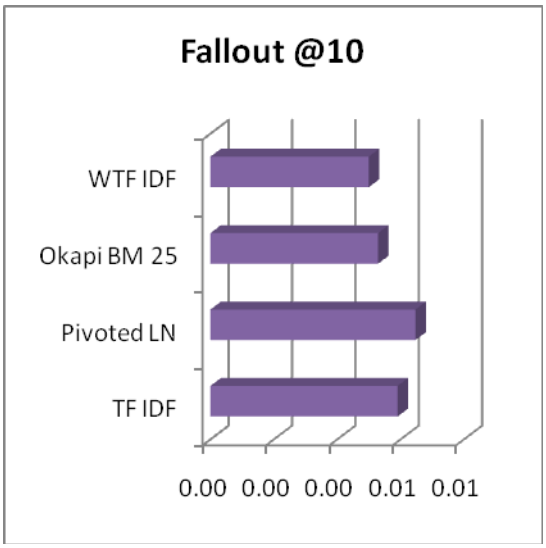

Fig. 4.12 Comparison of Avg. of fallout @10

4.2.6 Summary of Evaluation

Table 4.8 and 4.9 statically proves with a few exception the better performance of proposed approach WTF IDF in comparison with others mentioned above for all documents and for 1st 10 documents retrieved and same is represented in Fig 4.13 and Fig 4.14 showing better performance of proposed approach against others using precision, recall, F-Measure and Fallout as performance evaluation parameters.

Summary				
	TF IDF	Pivoted LN	Okapi BM 25	Weighted TF IDF
Precision	0.0396	0.0486	0.0486	0.0529
Recall	0.9923	0.9195	0.9195	0.8794
F- Measure	0.0732	0.0876	0.0876	0.0886
Fallout	0.9806	0.7540	0.7540	0.7906

Table 4.8 Comparison of Algorithms using all Evaluation Metrics

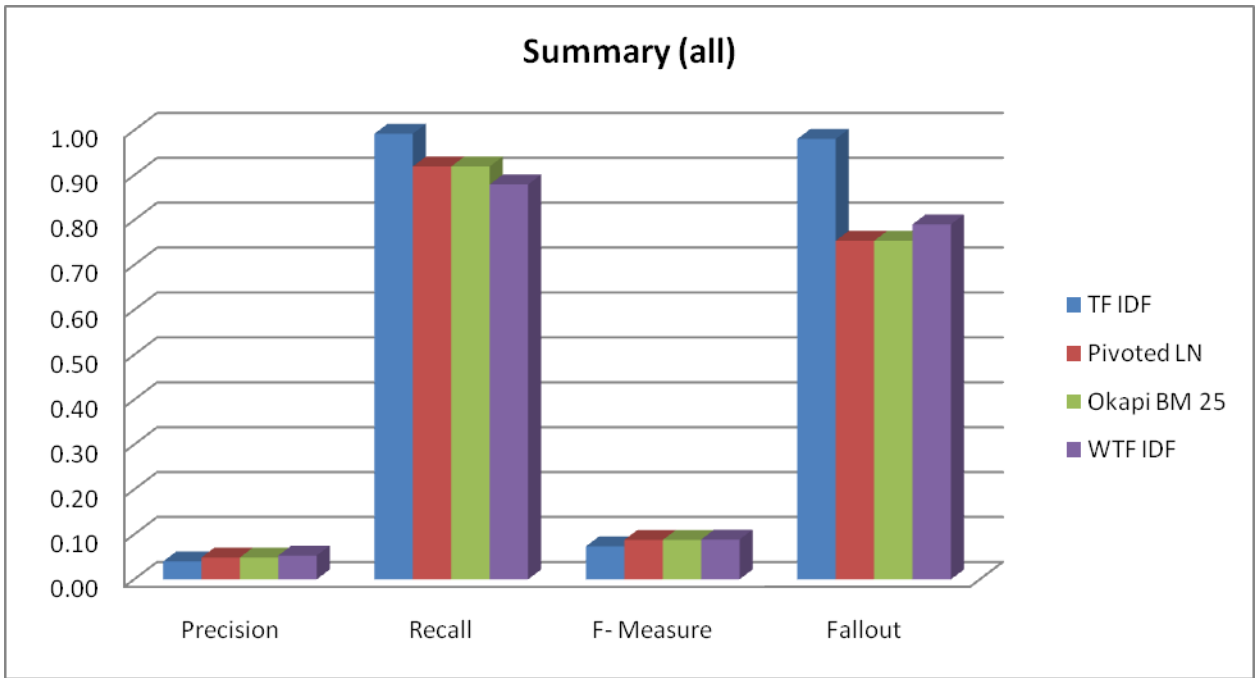

Fig. 4.13 Comparison of all evaluation metrics

Summary @R = 10				
	TF IDF	Pivoted LN	Okapi BM 25	Weighted TF IDF
Precision	0.3470	0.3000	0.4050	0.4190
Recall	0.1339	0.0796	0.1087	0.1429
F- Measure	0.1687	0.1096	0.1543	0.1860
Fallout	0.0059	0.0065	0.0053	0.0050

Table 4.9 Comparison of Algorithms using all Evaluation Metrics @R = 10

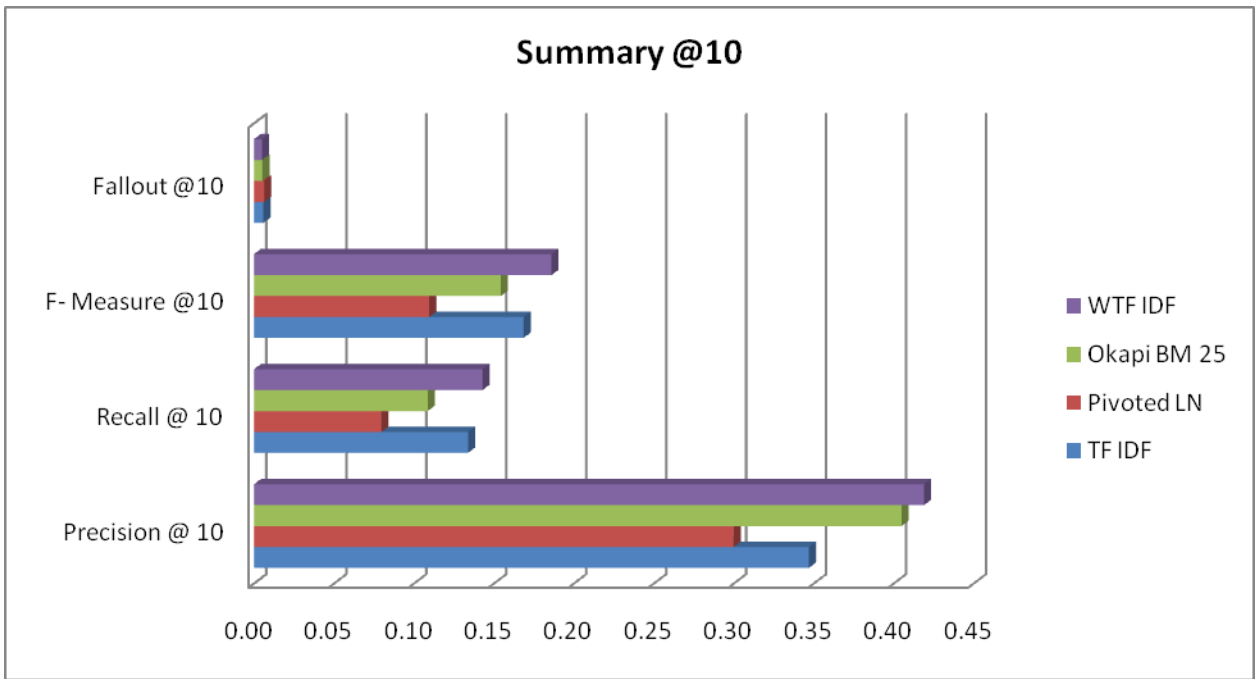

Fig. 4.14 Comparison of all evaluation metrics @10

4.3 Conclusion

In this work we introduced a more exhaustive approach Weighted TF IDF for document retrieval. WTFIDF calculates score of documents against a user query by giving importance to document structure, query word orders and contiguity, and synonym: to incorporate different writing approaches representing the same idea. The WTFID method produces considerable improvements in the major retrieval metrics when compared to the performance of existing relevance scoring formulae such as Okapi BM25, pivoted length normalization and traditional TFIDF. WTFIDF goes beyond the widespread bag of words, term frequency approaches to relevance scoring by incorporating fine-grained information regarding document structure in the relevance judgment process. The profound analysis of returned results shows that WTFIDF returns some novel information due to incorporating synonyms in query words which can't be retrieved by existing formulae.

Appendix A

A. WordNet 1.6 for .NET

WordNet is an online lexical reference system. WordNet was developed at the University of Princteon. Word forms in WordNet are represented in their familiar orthography; word meanings are represented by synonym sets (synset) - lists of synonymous word forms that are interchangeable in some context. Two kinds of relations are recognized: lexical and semantic. Lexical relations hold between word forms; semantic relations hold between word meanings.

To learn more about WordNet, read "Five Papers on WordNet", available via anonymous FTP and in printed form. A book, "WordNet: An Electronic Lexical Database", containing an updated version of "Five Papers on WordNet" and additional papers by WordNet users, will be available from MIT Press in Spring of 1998.

The purpose of these notes is to provide some documentation for the classes and interfaces exported by the .NET version of the library. These notes assume you are using a PC and Windows, but the .Net platform is being ported so that the code may well work in other environments.

Getting Started

Before you begin you should download and install WordNet 1.6 from Princeton on your computer. The standard installation places the dictionary files in C:\WN16\DICT . (This location is hardwired in the library, but can be changed programmatically.) You must also have installed the .NET runtime.

Try out the WordNet browser WNB.exe.

For writing your own applications using the library, the following public classes are provided in the namespace Wnlib (and Wnlib.dll). Their public interfaces are documented in detail in the

rest of this document. The classes you are most likely to use are shown in bold, and the ones you are least likely to use in italic.

Class	Description
AdjMarker	Discriminates attributive and predicative adjectives
AdjSynSet	Discriminates direct and indirect antonyms.
BitSet	A general-purpose class for handling arbitrarily large bitsets.
Exceptions	A class for listing exceptions to rules for plural words etc.
Frame	Used for sample sentences (of verbs).
Lexeme	Discriminates between separate entries in the dictionary for similarly spelled words.
MorphStr	This class works like StrTok to return a list of possible stem (lemma) words for a given work and part of speech.
Opt	A class for helping to specify search options via command line arguments
PartOfSpeech	The 4 parts of speech supported in WordNet 1.6 are Noun, Verb, Adjective, and Adverb. PartOfSpeech.of("noun") , "verb", "adj", and "adv" get this.
PartsOfSpeech	A flag (bitset) class for indicating what parts of speech are available for a word.
Pointer	SynSets contain Pointers to related words such as synonyms.
PointerType	A subsidiary class to SearchSet.
Search	The class used for returning complex and detailed dictionary and thesaurus information. The most useful field is a text buffer giving a readable version of the results of the search, but other data structures are also available (e.g. SynSet)
SearchSet	Defines a set of available searches for a given word and part of speech. Returned by WNDB.is_defined().
SearchType	Enumerates WordNet's searches for synonyms, antonyms etc. Searches may be recursive, so the constructor is of the form new SearchType(false,"OVERVIEW");
StrTok	A general purpose class encapsulating the functionality of string.Split
SynSet	Synonym set class: the main data set returned inside a search.
SynSetFrame	Links information for verbs in SynSets to Frames.
WNDB	The main route in: is_defined() tells you what words are in the dictionary, and what searches are then possible.
WNHelp	A class providing help strings for searches
WNOpt	A class for setting options for printing SynSets and Searches

For example, to look up a word to see if it is the dictionary, call:

```
if ( WNDB.is_defined(word, pos).NonEmpty) …
```

(word is a string, pos is "noun", "verb", "adj", or "adv"). Is_defined returns a SearchSet, so to test whether a particular search is possible, e.g. for "SIMPTR", you can write

```
if (WNDB.is_defined(word,pos)["SIMPTR"] ..
```

The possibilities here are listed under SearchType below.

To check the possible stems for a given word and part of speech,

```
MorphStr st = new MorphStr(word, pos);

String s;

}
```

To search for the main dictionary entry for a word and part of speech,

```
Search se = new Search(word, domorph,pos ,sch,0);
```

Then se.buf is a string (with embedded newlines). The sch parameter can be supplied as a string, such as "OVERVIEW". The final parameter can be nonzero to select a particular sense of the word from the dictionary. A boolean specifies whether to do MorphStr for you along the way.

Strings with embedded newlines are best unpacked with StrTok:

Appendix B

B. Porter Stemmer

Porter stemmer in CSharp, based on the Java port. The original paper is in Porter, 1980, An algorithm for suffix stripping, Program, Vol. 14, no. 3, pp 130-137,

See also http://www.tartarus.org/~martin/PorterStemmer

This revision allows the Porter Stemmer Algorithm to be exported via the .NET Framework. To facilate its use via .NET, the following commands need to be issued to the operating system to register the component so that it can be imported into .Net compatible languages, such as Delphi.NET, Visual Basic.NET, Visual C++.NET, etc.

1. Create a stong name:

    sn -k Keyfile.snk

2. Compile the C# class, which creates an assembly PorterStemmerAlgorithm.dll

    csc /t:library PorterStemmerAlgorithm.cs

3. Register the dll with the Windows Registry and so expose the interface to COM Clients via the type library

    (PorterStemmerAlgorithm.tlb will be created) regasm /tlb PorterStemmerAlgorithm.dll

4. Load the component in the Global Assembly Cache

gacutil -i PorterStemmerAlgorithm.dll

Note: You must have the .Net Studio installed.

Once this process is performed you should be able to import the class via the appropiate mechanism in the language that you are using. i.e in Delphi 7 .NET this is simply a matter of selecting:

Project | Import Type Libary

And then selecting Porter stemmer in CSharp Version 1.4"!

Appendix C

C. Stopwords

A	About	Above	across	after	afterwards
Again	Against	All	almost	alone	along
Already	Also	Although	always	am	among
Amongst	Amoungst	Amount	An	and	another
Any	Anyhow	Anyone	anything	anyway	anywhere
Are	Around	As	At	back	be
Became	Because	Become	becomes	becoming	been
before	Beforehand	Behind	being	below	beside
besides	Between	Beyond	Bill	both	bottom
but	By	Call	Can	cannot	cant
co	Computer	Con	could	couldnt	cry
de	Describe	Detail	Do	done	down
due	During	Each	Eg	eight	either
eleven	Else	Elsewhere	empty	enough	etc
even	Ever	Every	everyone	everything	everywhere
except	Few	Fifteen	Fify	fill	find
fire	First	Five	For	former	formerly
forty	Found	Four	from	front	full
further	Get	Give	Go	had	has
hasnt	Have	He	hence	her	here
hereafter	Hereby	Herein	hereupon	hers	herself
him	Himself	His	how	however	hundred
i	Ie	If	In	inc	indeed
interest	Into	Is	It	its	itself

keep	last	Latter	latterly	least	less
ltd	made	Many	may	me	meanwhile
might	mill	Mine	more	moreover	most
mostly	move	Much	must	my	myself
name	namely	Neither	never	nevertheless	next
nine	no	Nobody	none	noone	nor
not	nothing	Now	nowhere	of	off
often	on	Once	one	only	onto
or	other	Others	otherwise	our	ours
ourselves	out	Over	own	part	per
perhaps	please	Put	rather	re	same
see	seem	Seemed	seeming	seems	serious
several	she	Should	show	side	since
sincere	six	Sixty	So	some	somehow
someone	something	Sometime	sometimes	somewhere	still
Such	system	Take	Ten	than	that
The	their	Them	themselves	then	thence
there	thereafter	Thereby	therefore	therein	thereupon
these	they	Thick	thin	third	this
those	though	Three	through	throughout	thru
Thus	to	Together	Too	top	toward
towards	twelve	Twenty	two	un	under
Until	up	Upon	Us	very	via
Was	we	Well	were	what	whatever
when	whence	Whenever	where	whereafter	whereas
whereby	wherein	Whereupon	wherever	whether	which
while	whither	Who	whoever	whole	whom
whose	why	Will	with	within	without
would	yet	You	your	yours	yourself
yourselves					

5 Bibliography

1. *The Cystic Fibrosis Database: Content and Research Opportunities.* **Shaw, W.M. & Wood, J.B. & Wood, R.E. & Tibbo, H.R.** 1991, LISR 13, pp. 347-366.

2. *An algorithm for suffix stripping Program.* **Porter, M.** 1980, 14(3), pp. 130–137.

3. *Statistical Language Models and Information Retrieval: natural language processing really meets retrieval.* **Jong, Djoerd Hiemstra and Franciska de.** 2001, International 5(8), pp. 288-294.

4. **Baeza-Yates, R.A. and B. Ribeiro-Neto.** *Modern Information Retrieval.* s.l. : Addison- Wesley, London, 1999.

5. *Term-weighting approaches in automatic text retrieval.* **Salton, G. and C. Buckley.** 1988, Information Processing & Management 24 (5), pp. 513– 523.

6. *A linguistically motivated probabilistic model of information retrieval, In C. Nikolaou and C. Stephanidis (Eds.).* **Hiemstra, D.** Berlin Heidelberg : Proceedings of the 2nd European Conference on Research and Advanced Technology for Digital Libraries (ECDL), 1998. Springer-Verlag. pp. 569–584.

7. *A hidden Markov model information retrieval system. In M. Hearst, F. Gey, and R. Tong (Eds.).* **Miller, D.R.H., T. Leek, and R.M. Schwartz.** New York : Proceedings of the 22nd ACM Conference on Research and Development in Information Retrieval (SIGIR'99), 1999. ACM Press. pp. 214–221.

8. *A language modeling approach to information retrieval.* **Ponte, J.M. and W.B. Croft.** New York. : ACM Press, 1998. In W.B. Croft and A. Moffat and C.J. van Rijsbergen and R. Wilkinson and J. Zobel (Eds.) Proceedings of the 21st ACM Conference on Research and Development in Information Retrieval (SIGIR'98). pp. 275–281.

9. *A probabilistic justification for using tf idf term weighting in information retrieval.* **Hiemstra, D.** Berlin Heidelberg. : Springer-Verlag, 2000, International Journal on Digital Libraries 3 (2), pp. 131–139.

10. *Text categorization: A survey.* **Aas, L. and L. Eikvil.** 1999, Norwegian Computing Center, p. 941.

11. *Okapi at TREC-7: automatic ad hoc, filtering, VLC and interactive track. .* **S. E. Robertson, S. Walker, and M. Beaulieu.** 1999, In NIST Special Publication 500-242: Proceedings of the Seventh Text Retrieval Conference (TREC-7), pp. 253–264.

12. *Microsoft Cambridge at TREC-14: Enterprise Track.* **Nick Craswell, Hugo Zaragoza, Stephen Robertson.** November 2005, In Proceedings of the Fourteenth Text REtrieval Conference (TREC 2005). Gaithersburg, USA. Describes application and tuning of Okapi BM25F.

13. *Pivoted document length normalization.* **A. Singhal, C. Buckley, and M. Mitra.** 1996, In SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 21–29.

14. *Enhancing Relevance Scoring with Chronological Term Rank.* **Adam D. Troy, Guo-Qiang Zhang.** 2007, SIGIR'07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 599-606.

15. *Relevance weighting using distance between term occurrences.* **Thistlewaite., D. Hawking and P.** 1996, Technical Report TR-CS-96-08, The Australian National University.

16. *Indri: A language model-based search engine for complex queries.* **T. Strohman, D. Metzler, H. Turtle, and W. B. Croft.** 2005, Technical Report IR-416, University of Massachusetts Amherst.

17. *An information retrieval model using the fuzzy proximity degree of term occurences.* **Mercier., M. Beigbeder and A.** 2005, In SAC '05: Proceedings of the 2005 ACM symposium on Applied computing, pp. 1018–1022.

18. *Term proximity scoring for ad-hoc retrieval on very large text collections.* **S. B¨uttcher, C. L. A. Clarke, and B. Lushman.** 2006, In SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 621–622.

19. *NIST Special Publication . National Institute of Standards and Technology.* **Buckland, E. M. Voorhees and L. P.** November 2005, Proceedings of the Fourteenth Text REtrievalConference (TREC 2005), pp. 500-266.

20. *Evaluating evaluation measure stability.* **Voorhees., Chris Buckley and Ellen M.** 2000, In Proceedings of the 23rd Annual ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 33-40.