International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 10, October- 2014

A Dynamic Window Size Based Document Ranking Algorithm for Information Retrieval

Rajib Bag
Computer science and Engineering
Dr. B.R.Ambedkar Institute of Technology
Port Blair, INDIA

Abstract: **Query and documents relevancy has a great importance in information retrieval or data retrieval. User's general tendency is to put words in a query; those are correlated with each other and relevant for those documents that are necessity for searcher. Proximity has a great relevancy factor in finding relevant documents. Dynamic window size based proximity evaluation is effective and better than the fix sized window. Fixed window size has problems of finding the suitable length of the window to fix the query words.**

Now measuring the distance of query words in a dynamic window is a challenge. Another notion of IR is documents size varies from one to another. We have considered partial query matching and term frequency jointly with the dynamic window. A well balanced document ranking equation has been design to evaluate query and document relevance.

Keywords-Information retrieval; Algorithm; Proximity.

I. INTRODUCTION

Information Retrieval (IR) has become a dominant and important issue in this era of information technology. The exponential growth of using electronic storage has led to a great deal of interest in developing useful and efficient tools and software to assist users in searching diverse information with no single pattern of storing them. And also different users need in searching the necessary information with diverse interest of topics. Therefore, building an efficient search engine according to the user's interest and requirements within an affordable time and cost is a big challenge.

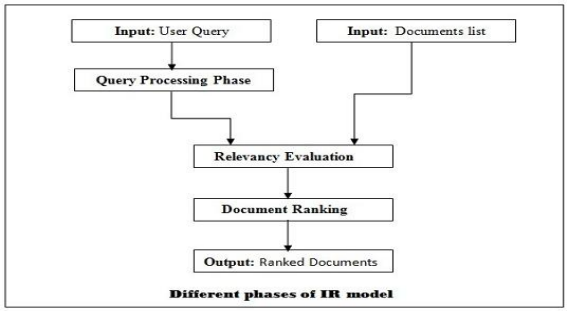

Fig 1: Different types of IR Model

In generic IR Model [Fig 1] model basically consists of 5 parts . First part consists of input section. Two input, one is user query and another one is lists of documents from where informatics documents need to search for user given query.

Query Processing: Now input query comes to the query processing section where given query need to be processed. The words in the query may not be in same as they are in the documents. There tense, parts of speech may differ in the documents. So these things are stemmed in this phase [8]. Another processes is expansion of query by using different methods like WorldNet, synonyms etc [11]. These streaming processes are done before ranking the documents.

Relevancy Evaluation: Functionality of this section is to find documents relevancy with query. There are many methods presents but we have developed a proximity based algorithm to do that specific work. This method is basically a statistical based approach. Document Ranking: If more than one document is found as relevant then how could we present those documents according to the user interest? So, relevant documents are necessary to arrange according to users interest. This section will take care of those things. Output: Ranked documents are output for any IR model.
Conventional ad hoc retrieval models do not take into account the closeness or proximity of terms. Document scores in these models are primarily based on the occurrences or non-occurrences of query-terms considered independently of each other. Intuitively, documents in which query-terms occur closer together should be ranked higher than documents in which the query-terms appear far apart.

The key algorithm of an IR system is similarity computing between queries and documents. Although, there exist, different approaches for finding relevant document for a query. Till now, the most popular algorithm is the inner product of vectors, and the vectors can be built by using weighting technologies, such as binary weight, tf-idf, query expansion, relevant feedback and etc [9]. In other words, most of the existing algorithms are based on vector computing. However, this method usually gets limited

IJERT

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

precision, because sometimes, a vector cannot represent a query properly [1].

In proximity based approach query words appearing closely in the document provide more contributions to the similarity value than the ones appearing separately [4]. The closer the query words in a document, the larger the similarity value between the query and the document. In paper [1] they have intense into two things 1.How the query words appearing closely. And define a window size for that. 2. And they made different windows according to the importance of the query words. Some query words, like named entities and base NP are called Core Words, while the other words are called Surrounding Words. Core Words are much more important than Surrounding Words, and should have special status in the retrieval processing (i.e. having larger weights).But identifying those core words are another difficult problem for IR.

II. Standard Evaluation Functions

A. Relevance

A document is relevant if it is one that the user perceives as containing information of value with respect to their personal information need.

B. Effectiveness

There must have some measuring rudiments that conclude how good our retrieval system is. In IR actual evaluators are user; if users are not satisfied with the result then no search result is productive. For measuring the effectiveness, Users feedbacks, are still important. We will measure our result, using three standard measuring functions. And those are discussed bellow.

C. Precision

It is a decisive factor by which we can measure how many relevant documents are retrieve among total searching documents for one time. In the equation, numerator has an intersection operation taking two terms. The meaning of intersection is how many retrieved documents are relevant. And in denominator there are a number of total documents that are accessed by a single search. In brief what fraction of the returned results is relevant to the information need?

$$Precision = \frac{|\{\text{relevant documents}\} \cap \{\text{retrived documents}\}|}{|\{\text{retrived documents}\}|}$$

For an example number of relevant documents=6; Retrieved documents=10. So precision value=6/10=.6

D. Recall:

This function used for measuring how relevant documents are retrieved in a single search. In the equation, numerator has an intersection operation taking two terms. The meaning of intersection is how many retrieved documents are relevant. And in denominator there is a term relevant documents. Here relevant documents mean, how many such documents present in the total document set. In brief, what fraction of the relevant documents in the collection was returned by the system.

$$Recall = \frac{|\{\text{relevant documents}\} \cap \{\text{retrived documents}\}|}{|\{\text{relevant documents}\}|}$$

For an example Number of relevant documents=4; Retrieved documents=10; Total relevant document (denominator) =10; Recall=4/10=.4

E. F-measure

It is formed by taking two parameter as Precision and Recall .The weighted harmonic mean of precision and recall, the F-measure or balanced F-score is:

$$F = \frac{2.precision\ .recall}{(precision\ + recall)}$$

For an example Precision=.6; Recall=.4;

III. RELATED WORK

A. Different Types Proximity Measure Technique

This section outlines several individual term-term proximity measures, measures which capture proximity of all terms in the query and also outlines some normalization measures[5][6][7][10]. For the term-term proximity measures outlined, it is necessary that the measure is symmetrical. For a specific term-term proximity measure (pm(a, b)) which measures the proximity between term a and b, we wish to find measures where pm(a, b) = pm(b, a). This definition of proximity is intuitive as defined in this work. There are different types of term-term proximity normalized function which are discussed below

Let document D={Zk; Za; Zg; Zb; Zc; Zd; Ze; Zb; Zf; Za; Zg; Za};

1. Span based Proximity distance:

Definition (Span): Span is defined as the length of the shortest document segment that covers all query term occurrences in a document, including repeated occurrences. For example, in the short document D, the span value is 7 for the query (Za; Zb; Zf).

IJERT

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

Definition (Min_Cover): Min Cover is defined as the length of the shortest document segment that covers each query term at least once in a document.
In the above example, if the query is (Za; Zb; D), its Min_Cover would be 1, but if the query is (Za;Zb;Zc) its Min_Cover would be 3 (the length of the segment from the second position to the sixth position).

B. Different Types of Distance aggregation measures:

In this section we shall discuss a pair wise distance between individual term occurrences, and then aggregate the pair wise distances to generate an overall proximity distance value. Specifically, they first pair up all the unique matched query words and measure their closest distances in documents [2][3].
For example, when a query has three different words (Za; Zb; Zc) and a document matches all the three words, we can obtain three different pairs of query term combinations: (Za; Zb); (Zb;Zc), and (Zc;Za). In the example document d, the closest distance for all these three pairs is 1 as they have all occurred next to each other somewhere. In this example Dis(Za;Zb;D) has been used to denote the closest distance between the occurrences of term Za and term Zb in document D. And document D consists of terms {Zk; Za; Zg; Zb; Zc; Zd; Ze; Zb; Zf; Za; Zg; Za};position of term start from 1 and goes up to 12.
Position vector is defined as the list of integer positions in the document D where terms occur. Therefore, the positions of each term reflect the actual ordering in which the terms occur in the document. Let, pos{D} denote the actual positions of a term, in document D and tf {D} be the term-frequency of terms, in document D. Therefore, position(Pos) of term Za = {2, 10, 12},position(Pos)of term Zb = {4, 8}.

1. Definition Minimum pair distance(Min_Dist):
The minimum pair distance is defined as the smallest distance value of all pairs of unique matched query terms. Formally,

$$Min\_Dist = min_{Z_a,Z_b \in Q \cap D,\ q_1 \neq q_2}\{Dis(Z_a, Z_b; D)\}$$

For example, the Min_Dist of the example document D for query Q = {Za;Zb} is 2 i.e (4 - 2).

2. Definition (Average pair distance (Ave-Dist)): The average pair distance is defined as the average distance value of all pairs of unique matched query terms. Formally,

$$Avg\_Dist = \frac{2}{n(n-1)} \sum_{Z_a,Z_b \in Q \cap D, Z_a \neq Z_b} Dis(Z_a; Z_b; D)$$

where n is the number of unique matched query terms in D, and sum will count Dis(za; Zb; D) and Dis(Zb, qa; D) only once.

3. Definition (Maximum pair distance (Max_Dist)): The maximum pair distance is defined as the largest distance value of all pairs of unique matched query terms. Formally,

$$Max\_Dist = max_{Z_a,Z_b \in Q \cap D, Z_a \neq Z_b}\{Dis(Z_a, Z_b; D)\}$$

4. Diff-avg-pos(Za, Zb, D): it is defined as the difference between the average positions of Za and Zb in D. This measure first calculates the average position of each of the terms using the position vectors and then uses the difference as a measure of proximity. In the given example, diff_avg_pos is 2 (i.e. ((2+10+12) =3)((4+8)=2)). It indicates where each term tends to occur (e.g. one term may tend to occur near the beginning of the document, while the other may tend to occur near the end of document). This measure makes use of position information about all occurrences of both query-terms.

5. Avg-min_Dist(a, b, D): It is defined as the average of the shortest distance between each occurrence of the least frequently occurring term and any occurrence of the other term. In the example, b is the least frequently occurring term so Avg_min_Dist = ((42) + (108)) =2 = 2. It can be seen that in two cases a and b occur very close together in D. These terms may constitute a phrase. If this phrase occurs multiple times in a document but far apart in that document, the two previously introduced measures would unfairly penalize the relationship simply because the occurrences of the entire phrase are far apart (i.e. all occurrences are not localized). The first measure (min dist) would not sufficiently reward the number of times the entire phrase occurs. The factor used to average the measure is the frequency of the least frequently occurring term. This is used so that each occurrence of term b (i.e. the least frequently occurring) is matched only once (this also ensures symmetry for this measure). In the example, the occurrence of at position 12 maybe completely unrelated to b (superfluous to the relationship between a and b) and is ignored by the measure.

6. Match-Dist (a, b, D): is defined as the smallest distance achievable when each occurrence of a term is uniquely matched to another occurrence of a term. For the previous distance function, the occurrence of a term may be used twice in the computation of the relationship (if it is an occurrence of the most frequently occurring term). However, if indeed each term is treated as having a pair, each occurrence of the least frequently occurring term should be paired with one distinct occurrence of the second term. The problem can be posed as follows; what is the best way to match the occurrences of pairs of terms so as to minimize the total distance between the pairs? This problem can be solved in exponential time using a dynamic programming algorithm. Fortunately, the frequencies of the query terms in the document are relatively small so that this calculation is feasible on TREC documents.
In the above example, the answer is the same as the previous measure as Match_Dist = ((42) + (108))=2 = 2.

IJERT

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

The frequency of the least frequently occurring term is used in averaging the score.

7. Max-dist (a, b, D): It is the maximum distance between any two occurrences of a and b. In the example max_dist = (128) = 4. This may be a useful measure of distance or may be a useful normalization factor for some of the other proximity measures.

8. Sum (tf (a), tf (b)): It is defined as the sum of the term-frequencies of a and b in D. This measure gives an implicit indication of the proximity of both terms. If this measure is high, the probability of closer occurrences of terms is automatically higher. In the example used, sum is 5 (i.e. 3+2) for document D.

9. Prod (tf (a), tf (b)): It is defined as the product of the term-frequencies of a and b. This measure also gives an implicit indication the proximity of both terms. If this measure is high, the probability of closer occurrences of terms is again automatically higher. In the example, prod is .5. Furthermore, these two measures (sum and prod) can be combined to give an indication of the equality of pair wise occurrences of terms in the entire document. For example, if

$$\sqrt{Prod\big(tf(Z_a), tf(Z_b)\big)}/sum(tf(Z_a), tf(Z_b))$$

=0.5 then both terms occur an equal number of times possibly indicating a closeness between the terms. If it is considerably less than 0.5, one term far more frequent.

10. Dl (D): It is defined as the length of the document and is a factor useful for normalization in IR. It may be very important in the scaling or normalization of some of the proximity measures introduced here. For example, shorter documents are more likely to have closer term proximities. In the example outlined earlier, dl(D) is 12.

D. PROPOSED WORK

In our algorithm we have concentrated on the term frequency as well as dynamic window size. In this algorithm dynamisms presents according to the query words presents in a windows. Window size varies according to the number of key words presents among the words. In our approach, if half of the key words present in a window that will be consider as a Partial Query matching. This partial query matching will take care the distribution of query words in a dynamic window. Documents which do not contain all the query words but some words presence may have equal or greater relevancy. Term frequency will take care, repetition of the same query words. Here half query words distribution has been taken care within the window. So at least 50% of the query words present in a window then it will be in the frame of semi-proximity. It is assume that if keywords proximity is high in the different position of a document then it will get more relevant score. Term frequency is a well known approach for getting relevancy of the document. User generally uses short and co-related words for searching. In proximity based approach, it is to find the words gap between the two key words in the document.

A. Proposed Algorithm:
Input:
1) Array of string K [j] of length n, to store keywords
2) Documents

Output:
1) Term-frequency(TF)
2) Counting Word gap(for Partial and full query)
3) Total query occur.
4) Partial query occur.

Initialize: TF, q, r;

Step1: for i=0 to m-1; // words in a document

Step2: for j=0 to n-1; //key words
Step3: if(S[i] = = K[j]) // This is done by stemming of words.
{
Step4: TF=TF+1;
Step5: While j is not in w[r]
Step6: w1 [q] = w[r] =j and Lock K[j]
r++;
q++;
}
Step7: If size of w[r-1] = n
{query++;
r=0;
store word-gap
Word-gap=0;
}
Step8: if size of w1[q-1]==sealing n/2
{
pquery++;
q=0;
initialize w1[]=0;
store word-gap for partial query;
}

Step 9: else
if(there is a lock in K[j] and j is not in w[ ])
Step 10: word-gap++ ;

B. Description of Algorithm:

Step1: For a particular document, word counting start from the first word of the documents and it will continue up to

IJERT

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

the last word of the document. In this algorithm say m is the number of words in a document.

Step2: In this step query words will be considered one by one. Total words present in a query are n. So index start from 0 to (n-1).

Step3: This is a one of the important step in this algorithm. In English word matching (Considering different form of same words) is an important part for searching the term-frequency. The algorithm we have used is a brute-force for string matching. After string matching, suffix, prefix and some rule of English words formation has been applied. And matching algorithm has complexity O(m*n)
Step4: TF is an abbreviation of term frequency. It is incremented when-ever a match occurred.
Step5: If key words not present in array w[ ].

Step6: stores the index of k[j] to the partial query and full query array for notification that that particular query words have been found.

Step7: Equalling total different query words found versus total number of query words. If total query presents then value will be incremented.

Step8: Condition will check the size of the partial query. If it is matched to the half of the total query then it will consider partial query match, and variable size will be incremented.

Step 9: If query words not matching.

Step 10: It will count words gap.

Complexity of the algorithm is O(m*n).
Example 1: In our algorithm, we have computed the proximity of query words in a document is consisting of

Document={Z1;Z2;Z2;Z4;Z5;Z2;Z3;Z7;Z7;Z6;Z8;Z10}

Query= {Z6; Z2; Z3}

So Word gap=4 and frequency of window=1; Semi-query gap=2 and frequency=1; Extra query term in between half Term-frequency= 5-(1*2)=3;

Now document ranking evaluation which will be described in the next section will be applied for ranking the documents.

C. PROPOSED DOCUMENT RANKING EQUATION:

In our document Ranking Algorithm, we have considered the three things together. First consider how many dynamic windows present and what the total lengths are. This window size depends on words distance among total key words in a query. Concept is frequency of windows grater in a document the relevancy is higher. And size of the words gap among the key words need to be less for query document relevancy.

$$Weight(W_d) = \{(t_1 * \frac{Number\ of\ Query(n)}{\sum_1^n Size\ of\ Window} + t_2 * \frac{Number\ of\ PQ(m)}{\sum_1^m Size\ of\ Window} + t_3 * (C_1 - \frac{1}{TF - (PQ * n)})\}$$

In the above proposed Ranking equation, has major three terms one by one have been discussed.

First term of the equation conveys the number of query present in a document is denoted by n; t1 is weight factor that will enrich the weight of the relevancy, generally put it in between 0 to 1. i.e 0< t1<1. Total number words gap in a document.

Second term of the equation conveys the number of partial query (PQ) present in a document is denoted by m. t2 is weight factor to enrich the weight of the relevancy generally put it in between 0 to 1, i.e 0< t2<1;

Third term of the equation conveys that it is based on term frequency; concept is only considering those numbers of key words that are not present in a partial query (PQ).

Abbreviations used in the above equation are

1. TF is term frequency (Total number of key words presents).

2. PQ = how much key words considered in partial query.
3. m= Number of partial query presents in a documents.

4. C1 is a constant

5. t1,t2, t3 are considered as weighting factor and lies in between 0 and 1. Mathematically we can express as 0< t1<1 ; 0< t2<1 ; 0< t1<1 ..

6. Relation between three weighting factor is t3< t2< t1 .

D. Tools Used for implementation:

Environment: Windows 7
Code: Java(NetBeans IDE 7.0 Beta)
Data Set: FIRE[12](Forum for Information Retrieval Evaluation)

IJERT

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

E. EXPERIMENTAL DATA:

We have experimented on the FIRE data set[12]. In this data set Asian specific day to day news of telegraph, has been stored in different documents. Documents are consisting of different fields of the news. Fields are of different news category like National, International, sports etc. And data set are stored according to the year basis. News of 3 years are stored 2005, 2006, 2007. This data set has been used for many diverse research fields like IR, cross language retrieval with TREC.

1 Corpus:

The group of documents are called corpus over which we perform retrieval as the (document) collection. It is sometimes also referred to as a corpus (a body of texts). Suppose each document is about 1000 words long (23 book pages). If we assume an average of 6 bytes per word including spaces and punctuation, then this is a document collection about 6 GB in size. Typically, there might be about M = 500,000 distinct terms in these documents. In FIRE data set different types of corpus present like English corpus, Hindi corpus, Bengali corpus etc. For our experiment we have used English corpus. Documents with their documents< document-Id > and contents of document starts with <text>and ends</text >. Queries consists of< query ID >, < titles >and< Summary >for each query. Some standard queries are present and according to those queries relevant documents are present in the FIRE-2010 data set.

E. RESULT ANALYSIS

Dynamic window size gives the better result than the traditional TF-IDF. We can see that span based ranking gives better result than the term frequency and pair-wise proximity. That revels on the figure below.

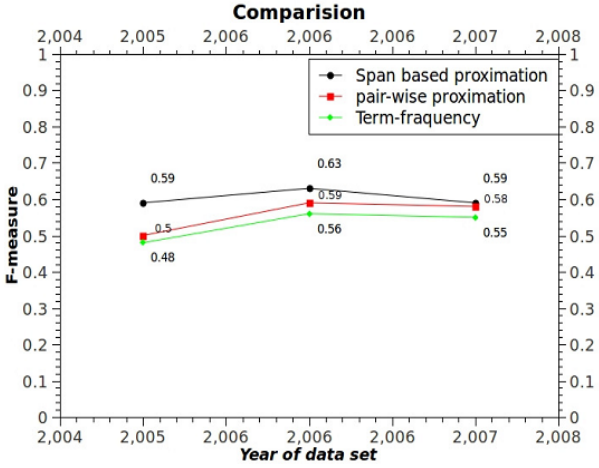

Figure 2. Comparative analysis of three algorithmS

In our experiment Semi-proximity and dynamic window size based evaluation gives the better result. We have tested the algorithm taking the news of each year. Outcome has been shown according to the year in corresponding tables. That reveals the efficiency of algorithm.

Table 1. Results of 2006 data set

Year:2006 Total documents 336	***Recall***	***Precision***	***F-measure***
Dynamic Window	73/100=0.73	73/130=0.56	0.63
Proposed ranking method	83/100=0.83	83/110=0.75	0.73

The above table (Table 1) reveals that year of data set is 2006. And we have experimented 336 numbers of documents. Out of 336 documents relevant documents are 100. We have compared two algorithms. First algorithm is dynamic window or span based proximity approach. Second algorithm is our proposed algorithm. Three columns (Recall, Precision, F-measure) expressed the efficiency of our proposed algorithm. We can see all the three cases our proposed algorithm gives the better result. And from the survey we have seen dynamic window size based or span based retrieval gives better result from term frequency or pair-wise proximity.

Table 2. Results of 2005 data set

Year:2005 Total documents 700	***Recall***	***Precision***	***F-measure***
Dynamic Window	160/250=0.64	73/130=0.55	0.59
Proposed ranking method	200/250=0.80	83/110=0.71	0.73

The above table (Table 2) expresses the experiment has been conducted on 2005 data set. Number of data set is larger than the previous experiments. In this experiment, relevant documents are 250 and out of that 160 are retrieved from span based approach but applying our approach 200 relevant documents is retrieved. Precision value also been improved to .71 from .55. In the precision column 200 relevant documents have been retrieved by extracting 280 documents.

IJERT

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

Table 3.Results of 2007 data set

Year:2007 Total documents 620	***Recall***	***Precision***	***F-measure***
Dynamic Window	154/200=0.77	154/290=0.53	0.59
Proposed ranking method	170/200=0.85	170/220=0.77	0.80

In 2007 data set(in the Table 3), 620 documents have been taken and out of that 200 documents are relevant. Out of 200 relevant documents dynamic window able to retrieve 154 and our proposed algorithm retrieved 170 relevant documents. Retrieving 154 relevant documents dynamic window capture 290 documents, our algorithm takes only 220 documents. F-measure is a harmonic mean of precision and Recall that value also higher than the previously described algorithm.

We present a graphical representation (in figure 3) of F-measure result. Y-axis represents the value of F-measure. Y-axis ranges from 0 to 1. And X-axis presents years of data set (2005, 2006, and 2007). Star marks line shows the result of our proposed method. In three points we got the result (0.75, 0.73, 0.8) that are higher than the previously described algorithm marked as (0.59, 0.63, 0.59). So we can conclude that our proposed method gives the better result than the previous measuring techniques. We have not calculated the fall-out because if F-measure is high then fall-out is automatically low. So it is not necessary to compare. Now one thing necessary to compute how many relevant document are retrieved within each top 10 documents. Result of the different years is shown in the table 4.

Result has been compared taking the parameters Precision (Pr), Recall (Rc) and generate F-measure. Two things need to measure, in this experiment Term frequency and absolute word gaps between two keywords. We can see easily that we consider not exactly the term-frequency rather extra terms present in the document. If we closely look at the algorithm then we can understand that with same searching coast we can get another two parameters. Apply this parameter (proximity of key words) result is sufficiently increased.

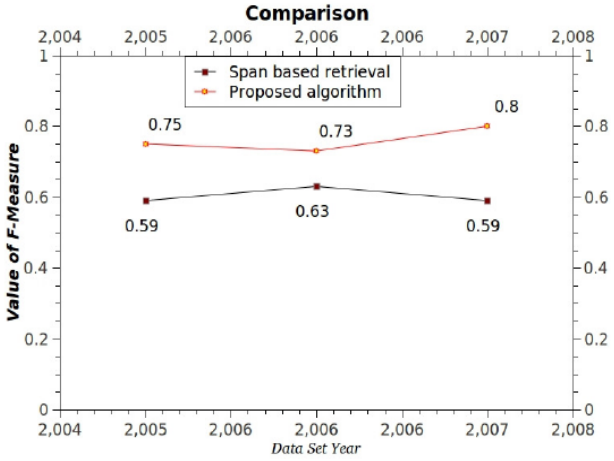

Figure 3. Comparative result with proposed algorithm

Table 4.Results of top retrieved document

Number of retrived documents	***Year 2005***	***Year 2006***	***Year 2007***
Out of 1st 10 documents	8	8	8
Out of 2nd 10 documents	9	8	7
Out of 3rd 10 documents	8	7	7

The above table shows that what are the numbers of related according to the query are found by applying our proposed algorithm. We have considered top 30 documents consisting of 10 documents each, with standard query prescribed in the FIRE data set. Result shows that our ranking algorithm and equation are giving good result.

F. CONCLUSION AND FUTURE WORK

We have concentrated on, how the efficiency of a retrieval system can be improved. Different proximity based approach have been surveyed. After that we have developed an algorithm that is obliging to enhance recall value. We have also design a ranking algorithm to rank the document according to the measured relevancy. In this approach retrieving small size of text document is very helpful. Described scoring is balanced with respect to total query matching; partially query matching as well as term-frequency. Sometimes mixed text size document need to be search, so this balanced equation is effective one.

And this approach not only searching documents according to the query, it is also helpful to search a string for pattern matching. It is applicable for feature search from an image, speech and gives the relevant image or speech for that set of attributes. And it is possible to do the classification according to the ranked document. If we able to generate the ranked class then, it will minimize the searching cost for each time.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

G. ACKNOWLEDGMENT

First of all we would like to thank the Almighty, who has always guided me to work on the right path of the life. Our greatest thanks are to our parents who bestowed ability and strength in our, to complete this work.

This work would not have been possible without the encouragement of Dr. Pabitra Mitra, Associate Professor, IIT-kharagpur. His enthusiasm and optimism made this experience both rewarding and enjoyable.

We are equally grateful to Prof. Manoj Kumar Mishra, Assistant Professor, KIIT-university who always encouraged us to keep going with work and always advised with his invaluable suggestions.

We would like to express our sincere gratitude towards, Principal, HOD and entire faculty and staff members of Dr. B.R.Ambedkar Institute of Technology, PortBlair for their direct-indirect help, cooperation, love and affection for completion this work.

H. REFERENCES

[1] Dik L. LEE, Huei Chuang, Kent Seamons, *Document Ranking and Vector-Space Model*, IEEE, 1997

[2] Qianli Jin, Jun Zhao,Bo Xu,*Window-based Method for Information Retrieval*,2004-2005

[3] Roman Cummins, Colm ORiordan, *Learning in a Pairwise Term-Term Proximity Framework for Information Retrieval*, 2010

[4] Desislava Petkova, W. Bruce Croft,*Proximity-based Document Representation for named Entity Retrieval*,2006

[5] Tao Tao, ChengXiang Zhai,*An Exploration of Proximity Measures in Information Retrieval*,2011

[6] Michel Beigbeder, Annabelle Mercier, *An Information Retrieval Model Using the Fuzzy Proximity Degree of Term Occurences*, 2004

[7] Santosh Kumar Ray and Shilendra Singh, *Rough set Based Social Networking Framework to Retrieve User-Centric Information*, Springer-Verlag, 2009.

[8] Kraaij Wessel and Pohlmann Renee, *Viewing stemming as recall enhancement*, Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval, 40-48,1996.

[9] Frakes W.B. ,*Term conation for information retrieval*, Proceedings of the 7th annual international ACM SIGIR conference on Research and development in information retrieval, 383-389,1984

[10] Shailendra Singh ,*ROUGH FUZZY APPROACH FOR USER-CENTRIC TEX INFORMATION RETRIEVAL*,2005.

[11] Rila Mandala, Tokunaga Takenobu, and Tanaka Hozumi,*The Use of WordNet in Information Retrieval*,2002.

[12] www.isical.ac.in /_re/2010/datadownload.html/english

IJERT

(This work is licensed under a Creative Commons Attribution 4.0 International License.)