KEYWORD EXTRACTION USING CENTROID GRAPH AND WITH A STUDY OF TEXTRANK AND HITS ALGORITHM

Anns Sebastian [#1], Dr Pushpalatha K.P[*2],
M.Tech Student, School Of Computer Science , Mahatma Gandhi University, Kottayam, Kerala, India
Associate Professor, School Of Computer Science, Mahatma Gandhi University, Kottayam, Kerala, India

Abstract— **Automatic keyword extraction identifies the terms that best describe the subject of the document or else it describe salient features of the article. This paper introduce a keyword extraction algorithm using centroid graph from single document. Algorithm proposed a graph based approach to the centroid values. In addition to extraction algorithm, the paper make a comparative study of HITS and TextRank algorithm. The keywords are recursively evaluated according to the cohesion to the document context. Evaluation of algorithms based on the precision and recall of extracted keywords.**

Keywords — Automatic keyword extraction, MEAD, HITS, TextRank

I INTRODUCTION

Keywords are useful for the people to know about the document context before it is read. Automatic keyword extraction is the process of selecting the salience features of words from document. It is used to identify a small set of keywords from the given document context which define the meaning of the document. It should be done systematically and with either minimal or no human intervention, depending on the model. The goal of automatic extraction is to apply the power and speed of computation to the problems of access and discoverability, adding value to information organization and retrieval without the significant costs .Keywords give a clue about the context of article so that users can decide their interest. They will give an outline about the document context to the readers.

Extraction of a sequence of one or more words provide a compact representation of article's context is a vital role in Text Mining, Information Retrieval, Natural Language Processing. Human made keyword extraction is very time consuming and costly. Also digital information in day to day life is increases exponentially. The existing approaches of keyword extraction are divided into four categories, simple statistics, linguistic, machine learning, and hybrid approaches. Simple statistics methods are simple have limited requirements and don't need the training data. They tend to focus on non-linguistic features of the text such as term frequency, inverse document frequency, and position of a keyword. The statistics information of the words can be used to identify the keywords in the document. Other statistics methods include word frequency, TF*IDF, word co-occurrences. Linguistics Approaches use the linguistic features of the words, sentences and document. Methods which pay attention to linguistic features such as part-of-speech, syntactic structure and semantic qualities tend to add value, functioning sometimes as filters for bad keywords. Machine Learning Approaches consist of a set of training documents, each of which has a range of human-chosen keywords as well. Then the gained knowledge is applied to find keywords from new documents. The Keyphrase Extraction Algorithm use machine learning method. Hybrid approaches is the combination of all reaming approaches and use some heuristic knowledge in the task of keyword extraction such as the position, length, layout feature of the words, html tags around of the words. In this paper proposes MEAD extraction of keywords using its centroid value in graph based model. Also make a comparative study of mostly widely used rank based algorithms such as HITS and TextRank algorithm for the extraction.

This paper organized as follows, Section II describe the MEAD extraction of keywords using its centroid value in graph based model. Section III describe the TextRank algorithm. Section III define the HITS algorithm and its working. Section V make comparison of HITS and TextRank algorithm. Section VI deals with the result and discussion pretend with tabulation and graph. Section VII presents the conclusion

II KEYWORD EXTRACTION USING CENTROID GRAPH (KEUCG)

MEAD based on sentence extraction. For each sentence in a cluster of related words, MEAD computes three features and uses a linear combination of the three to determine what sentences are most salient. By using this method we extract words that describe salience of document context. KEUCG is a graph based approach to the centroid value in order to increase its efficiency. MEAD extraction algorithm used three features to compute the sentence score of the each sentence. The three features are centroid score, positional value, first sentence overlap. Sentence score are computed based on the linear combination of all three features. Hierarchal selection of sentence based on its sentence score. The keyword extraction we only adopt the centroid value of each word. MEAD decide which word is to be extracted based on its value. The input to the MEAD is a document and output is set of keyword extracted from the document. Following are the steps of keyword extraction.

A. Pre-Processing

Pre-Processing is the primary step of Natural Language Processing. It is cleaning of data and convert into list of words. The list of words are converted into lower case tokens. The token is a string of characters collected together in a document. In simple words, tokens are the words in the sentences that are used in lexical analysis. Tokenization is the process of splitting the sentence into words, phrases, symbols or other components. Stop words are the commonly used

words and are removed by comparing the words with dictionary of stop words. Stop-word elimination ultimately enhance performance of feature extraction algorithm.

B. Frequency Calculation

The Frequency of each token is computed. The frequency of word means how many times the word is repeated in the document .The average frequency of word is how many times that particular word is repeated divided by total number of words. The frequency and its average of each words is computed whereas df is the document frequency and tf is the average frequency.

C. Centroid Value

Centroid value of each word is computed using MEAD extraction algorithm. A centroid is a set of words that are statistically important to the document context. Centroid could be used both to classify the relevant document and identify the salience feature of the document. The centroid value of each word w in the sentence i is computed using(1).

$$Cw;i=tf*\log 10(n/df) \quad (1)$$

tf = average frequency
df=document frequency
n= article size

D. Centroid Graph

In this algorithm tokens are the nodes of the graph. The first step is to retrieve most relevant words from document based on its centroid value. These are the base set of the graph. The algorithm performs a series iteration on the base set to update the relation of base set to other set of nodes. Rank value of each base set node depend on its centroid value and number of degree of outgoing edges. Base set are arrange based on its rank value and eliminate less rank nodes. The base set provide extracted keywords contain salient features of the document. In the Figure1 S,V,T,Y belongs to base set of nodes

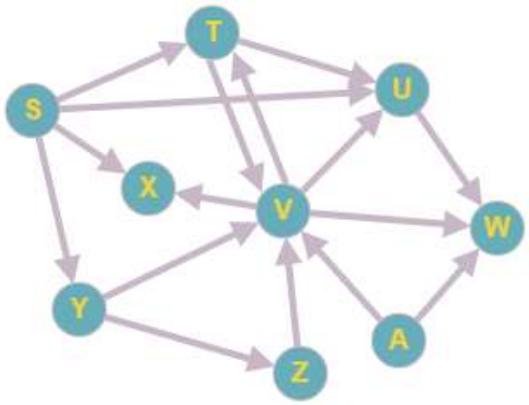

Figure 1 Centroid Graph

III TEXTRANK ALGORITHM

In a graph based ranking algorithm a graph represents article and interconnected words or other text entities with meaningful relationship. TextRank algorithm is adopted from the PageRank algorithm. PageRank algorithm is used by Google search to rank their website based on its search engine results. PageRank is developed by Larry Page one of the founders of Google. Rank of the page determine how much that page is important is depends on the counting number and quality of links to that page. PageRank is a link analysis algorithm and it assigns a numerical weighting to each element of a hyperlinked set of documents, such as the World Wide Web, with the purpose of "measuring" its relative importance within the set. TextRank is inherited from the PageRank algorithm. PageRank is related to webpage whereas TextRank is related to the text document. It used to analysis text document in the same manner of web pages.

In the TextRank we consider the sentence equivalent to the web pages.The PageRank the rank is depends on the number of the links to that page whereas in the TextRank the link is the probability of going from sentence A to sentence B is equal to the similarity of the 2 sentences. The algorithm is language agnostic and unsupervised. In the TextRank , we have to build a graph that represents the text, and vertices represents the words and edge represents interconnection of words. The Modified version of the PageRank is following,

$$PR(A)=(1-d)+d(PR(B)/L(B)+PR(C)/L(C)+\ldots.) \quad (2)$$

Where
PR(A)=PageRank of A
B,C,….=Links to the page A
PR(B)=Page rank of B
L(B)=number of links to the page B
(1-d)= To make up for some pages that do not have any out-links to avoid losing some page ranks
PR(B)/L(B)= PageRank of B distributing to all pages that B links to.
d = damping factor which can be set between 0 and 1.

Damping factor: The PageRank theory holds that any imaginary surfer who is randomly clicking on links will eventually stop clicking. The probability, at any step, that the person will continue is called a damping factor d [4]. The damping factor can be set to any value such that $0<d<1$,nominally it is set around 0.85. The damping factor is subtracted from 1 and this term is then added to the product of the damping factor and the sum of the incoming PageRank scores. In the TextRank we can neglect the damping factor because there no occurrence of random clicking on articles while extracting the keywords. So in the TextRank we use simple equation PageRank.

$$PR(U)=\Sigma\, PR(V)/L(U) \quad (3)$$

A. Implementation Of TextRank Algorithm

In the TextRank algorithm the input is graph.we have to build a graph that represents the text, and interconnects words or other text entities with meaningful relations The following pseudo code explain the method for implementing TextRank algorithm.

Step1: Build Graph g for the articles, words represent vertex and edge represents interconnection words.

Step2: Initialize the rank value of each word by 1/n where n is the total number of words in the article to be ranked.

Step3: Repeat for each node i such that $0 \leqslant i < n$ for k times. Let Rank be an List of n element which represent rank of each word in the article

$$Rank \leftarrow \Sigma\, Rank[i]/Degree[i] \quad (3)$$

Step4:Update the rank value of each word in the article for k iteration

Rank[j] ←Rank

Whereas
Rank[j] = TextRank of each word in the article
Rank[i] =Rank of each word that is pointing to word j

Degree[i]=Outgoing degree of word j

Repeat from step 3 until the rank value converges that is the values of two consecutive iterations match.

IV HITS ALGORITHM

Hyperlink-Induced Topic Search (HITS; also known as hubs and authorities) is a link analysis algorithm that rates Web pages, developed by Jon Kleinberg. The idea behind Hubs and Authorities stemmed from a particular insight into the creation of web pages when the Internet was originally forming that is, certain web pages, known as hubs, served as large directories that were not actually authoritative in the information that they held, but were used as compilations of a broad catalog of information that led users direct to other authoritative pages. In other words, a good hub represented a page that pointed to many other pages, and a good authority represented a page that was linked by many different hubs. Here we inherited the properties of HITS algorithm to extracted the keywords from the document. Instead of webpage we use the words of the document context. We have to build a graph that represents the document and vertex represents the words of document, edges represents the correlation of words.

The HITS algorithm assign two scores for each word, its authority score which estimate the value of the words and its hub score which determines the value of its links to other word.

A. Authority And Hub Update

Authority and hub values are defined in terms of one another in a mutual recursion. An authority value is computed as the sum of the scaled hub values that point to that word. A hub value is the sum of the scaled authority values of the word it points to. The algorithm performs a series of iterations, each consisting of two basic steps:

•Authority Update: Update each node's Authority score to be equal to the sum of the Hub Scores of each node that points to it. That is, a node is given a high authority score by being linked from pages that are recognized as Hubs for information.

•Hub Update: Update each node's Hub Score to be equal to the sum of the Authority Scores of each node that it points to. That is, a node is given a high hub score by linking to nodes that are considered to be authorities on the subject.

The Hub score and Authority score for a node is calculated with the following algorithm:

- Start with each node having a hub score and authority score of 1.
- Run the Authority Update Rule
- Run the Hub Update Rule
- Normalize the values by dividing each Hub score by square root of the sum of the squares of all Hub scores, and dividing each Authority score by square root of the sum of the squares of all Authority scores.
- The squares of all Authority scores.
- Repeat from the second step as necessary

❖ Authority Update Rule

Update auth(p) to the summations, where n is the total number of words connected to w and i is a word connected to w. That is, the Authority score of a page is the sum of all the Hub scores of pages that point to it.

$$auth(p)=\sum_{i=1}^{n} hub(i) \quad (4)$$

❖ Hub Update Rule

Update hub(p) to the summations. where n is the total number of words w connects to and i is a word which p connects to. Thus a page's Hub score is the sum of the Authority scores of all its linking words.

$$hub(p)=\sum_{i=1}^{n} auth(i) \quad (5)$$

The final hub-authority scores of nodes are determined after infinite repetitions of the algorithm.

B. Implementation Of HITS Algorithm

Authority and hub values are defined in terms of one another in a mutual recursion. An authority value is computed as the sum of the scaled hub values that point to that word. A hub value is the sum of the scaled authority values of the word it points to. The following pseudo code explain the method for implementing HITS algorithm

```
Step1: G := set of words in the document
Step2: for each word w in G do
   Step3:    w.auth=1 // w.auth is the authority score of
             the word w.
Step4:    w.hub=1 // w.hub is the hub score of the
             word w
Step5: for i=1 to k do // run the algorithm k times
Step6:   norm=0
Step7:   for each word w in G do // update all
                    authority value first
Step8:          w.auth=0
Step9:  for each word v in w.incoming Neighbors do
          //w.incomingNeighbors is the set of pages
            that link to w
Step10:   w.auth += v.hub
Step11:    norm += square(w.auth)
           //calculate the sum of the squared auth
              values to normalise
Step12:      norm = sqrt(norm)
Step13:      for each word w in G do
                // update the auth scores
Step 14:        p.auth = p.auth / norm
                // normalise the auth values
Step15:          norm=0
Step16:  for each word w in G do//Update hub score
Step17:    w.hub=0
Step18:  for each word r in w.outgoingNeighbors do
          // w.outgoingNeighbors is the set of words
             that w links to
Step19:       w.hub += r.auth
Step20:       norm += square(p.hub)
             // calculate the sum of the squared hub
                values to normalise
Step21:      norm = sqrt(norm)
Step22:    for each page p in G do
           // then update all hub values
Step23:       p.hub = p.hub / norm
              // normalise the hub value
Step24:  Stop
```

In each iteration diverging values of authority and hub are obtained. So, it is necessary to normalize the values after each iteration. Normalization is done by dividing each Hub

score by the square root of sum of the squares of all the Hub scores, and dividing each Authority score by the square root of sum of the squares of all the Authority scores. The score of each node, that is each word is the average of hub and authority score

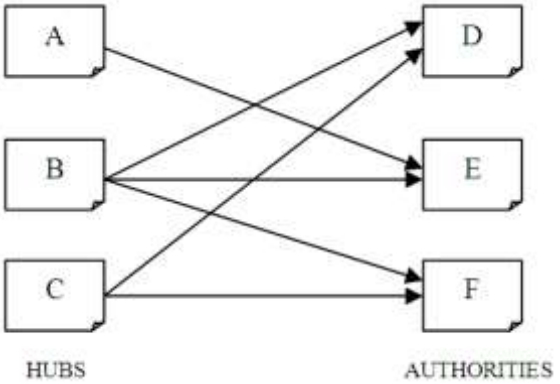

Figure 2 Hubs and Authorities

V. COMPARISON TEXTRANK AND HITS

In order to compare the TextRank algorithm and HITS algorithm, we build graph for the documents contain nodes represents the words. The same graph given as the input for the both TextRank and HITS algorithm for k iterations. Each node represents words in the article. Result of ranking according to HITS (average of Authority score and Hub score) and TextRank algorithm of Graph g show below table. The following Graph ,Text is considered as the input to both algorithm.

"A virus named after Kampung Sungai Nipah, a village in Malaysia, where it was first discovered in 1998-99 . The virus, that eventually killed 105 people in Malaysia, was first suspected to be Japanese encephalitis (JE) which, like the Nipah virus, induces brain inflammation .The virus, which was traced back to the pigs, led to a large-scale culling of the animals in this region . Pig feed was contaminated with bat excretions . Nipah is believed to be transmitted from bat excretions to people." This text is considered for the graph.

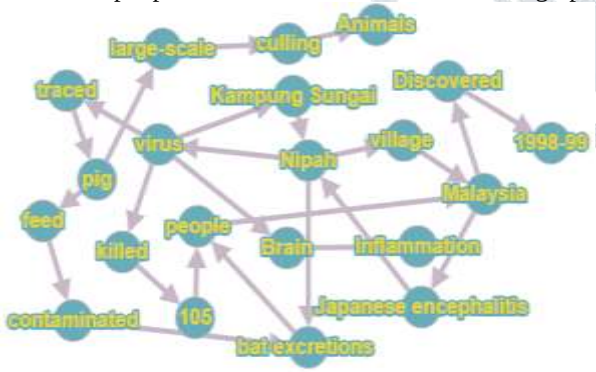

Figure 3 Graphical Represntation Of Text

A. *TextRank Algorithm Results*

TABLE I

TextRank Of Different Words In The Graph At Iteration 1, 5 And 10

	Iteration 1	Iteration 5	Iteration 10
Nipah	0.06	0.709	1.023
Virus	0.035	0.709	1.023
Malaysia	0.04	0.6361	0.823
Pig	0.04	0.622	0.816
Bat	0.04	0.395	0.623
Kampung Sungai	0.02	0.275	0.458
People	0.02	0.159	0.256
Japanese encephalitis	0.02	0.283	0.369
Animals	0.02	0.125	0.235
Culling	0.02	0.024	0.142

B. *HITS Algorithm Results*

TABLE II

Authority Score Of Different Words In Graph In Iteration 1,5 And 10.

	Iteration 1	Iteration 5	Iteration 10
Nipah	0.4120	0.456	0.501
Virus	0.27472	0.3872	0.4231
Malaysia	0.2747	0.2769	0.356
Pig	0.1373	0.1386	0.2356
Bat	0.2747	0.2769	0.2769
Kampung Sungai	0.1373	0.2345	0.3245
People	0.274	0.2747	0.274
Japanese encephalitis	0.1373	0.235	0.2456
Animals	0.1373	0.1373	0.201
Culling	0.1373	0.1323	0.1323

TABLE III

Hub Score Of Different Words In Graph In Iteration 1,5 And 10.

	Iteration 1	Iteration 5	Iteration 10
Nipah	0.4120	0.482	0.512
Virus	0.37472	0.3756	0.4322
Malaysia	0.2747	0.3231	0.4121
Pig	0.2747	0.2747	0.2896
Bat	0.2747	0.2747	0.2896
Kampung Sungai	0.1373	0.1423	0.2364
People	0.2747	0.2747	0.301
Japanese encephalitis	0.1373	0.2363	0.2463
Animals	0.1373	0.1373	0.203
Culling	0.1373	0.1383	0.1383

TABLE IV

Score Of Each Word In Graph Iteration1,5,10

	Iteration 1	Iteration 5	Iteration 10
Nipah	0.4120	0.469	0.4276
Virus	0.32472	0.3814	0.4276
Malaysia	0.2747	0.3	0.38405
Pig	0.206	0.2665	0.2626
Bat	0.2747	0.2758	0.2832
Kampung Sungai	0.07865	0.1884	0.2804
People	0.2747	0.2747	0.2875
Japanese	0.1373	0.2.356	0.2459

encephalitis			
Animals	0.1373	0.1373	0.202
Culling	0.1373	0.1353	0.1353

HITS is a general algorithm is used for calculating the authority and hub score in order to rank the data. The basic aim of the algorithm is induce the graph by finding set of words that is relevant to the document. In TextRank generated entire document graph rather than a small subset. The rank calculation is based on the correlation of words in the graph and depends on the degree outgoing , incoming nodes in the graph.

VI. RESULT AND DISCUSSION

In this section the proposed model KEUCG is compared with the performance of HITS and TexrRank algorithm. The evaluation method is using precision and recall. Experiment results are conducted and analysed using a set of newspapers. Here we show the results of 5 newspapers under health category, where each has its predefined keywords. Evaluation metric is considered are the Precision, Recall and F-Measure which are the standard metric for retrieval effectiveness in information retrieval. Calculated follows,

$$Precision = TP/(TP+FP) \quad (6)$$

$$Recall = TP/(TP+FN) \quad (7)$$

$$F = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (8)$$

Where ;

TP =keyword extracted keywords by the algorithm and already found in document's predefined keyword.

FP = keyword extracted keywords by the algorithm and doesn't found in document's predefined keyword.

FN= document's predefined keyword that are not extracted by the algorithm

TABLE V

Precision, Recall and F-Measure Results Of KEUCG Algorithm

NEWS PAERS	PRECISION	RECALL	F
HINDUSTAN TIMES	0.8	0.7272	0.761
INDIA TODAY	0.72	0.86	0.792
INDIAN EXPRESS	0.869	0.75	0.805
THE HINDU	0.80	0.72	0.7578
TIMES OF INDIA	0.9	0.81	0.8476

P=Precision
R=Recall
F=F-Measure

TABLE VI

Precision, Recall and F-Measure Results Of HITS Algorithm

NEWS PAERS	PRECISION	RECALL	F
HINDUSTAN TIMES	0.6	0.5	0.545
INDIA TODAY	0.7	0.58	0.634
INDIAN EXPRESS	0.7	0.583	0.6361
THE HINDU	0.6	0.5	0.54
TIMES OF INDIA	0.6	0.46	0.520

P=Precision
R=Recall
F=F-Measure

TABLE VII

Precision, Recall and F-Measure Results Of TextRank Algorithm

NEWS PAERS	PRECISION	RECALL	F
HINDUSTAN TIMES	0.5	0.71	0.586
INDIA TODAY	0.6	0.5	0.5454
INDIAN EXPRESS	0.7	0.68	0.689
THE HINDU	0.7	0.58	0.6343
TIMES OF INDIA	0.7	0.623	0.65925

P=Precision
R=Recall
F=F-Measure

Table V,VI, VII shows the precision, recall and F results for term keyword extraction based on Keyword Extraction Using Centroid Graph, HITS algorithm and TextRank algorithm. KEUCG algorithm results in enhanced precision , Recall and F evaluation over other methods.

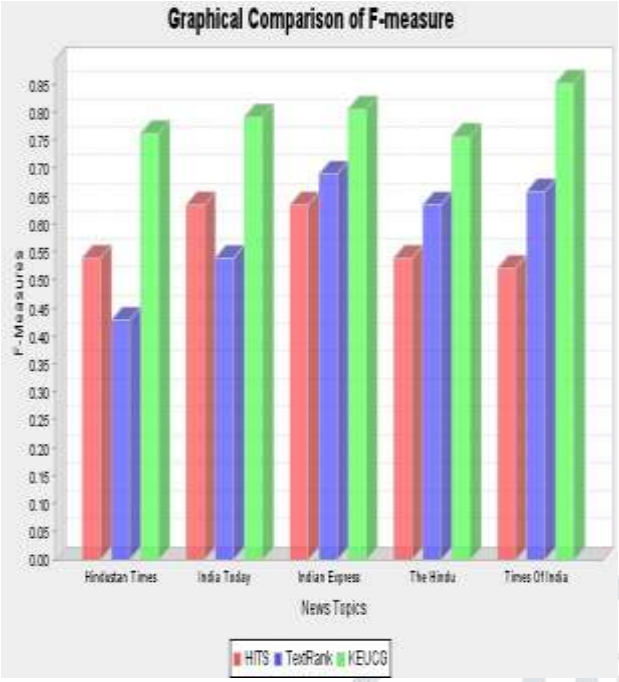

Figure 4 Graphical Comparison Of F-Measure

Fig 4. shows a graphical comparison of topic wise F-measures of three different approaches and it has been observed that KEUCG is better among the three as it has highest F-measures in most of the topics as compared to the other two approaches. As can be see that KEUCG approach extracted keywords is similar to the predefined keywords in the document. It shows better efficiency when we compare to the other two approaches. The efficiency is calculated by number of keywords extracted by the algorithm and already found in document's predefined keywords divided by total number of keywords in document's predefined keywords. The KEUCG have the efficiency of 80% whereas other two approaches have the efficiency of 70% or 60%.The KEUCG shows better results in keyword extraction of single documents.

VII. CONCLUSION

The work has suggested a method for automatic keyword extraction from single document and it compare with other two methods by reviewing the results and evaluation's score. It has been observed that the proposed method shows better results in extracting keywords from single document compare to other two approaches. On the basis of this study we can conclude that both TextRank and HITS algorithm are different link analysis algorithm. TextRank is inherited from PageRank. TextRank and PageRank are almost similar ,the link in PageRank is converted into the similarity of two words in the TextRank. HITS algorithm rank the pages according to authority and hubness of a word. After going through exhaustive analysis of TextRank and HITS algorithms it shows better results in the web page analysis. PageRank is used in the Google search engine and HITS algorithm used in IBM search engine clever. KEUCG method is a combination centroid based MEAD algorithm and graph based keyword extraction. It can be conclude that keyword extracted from single document using KEUCG is close to the predefined keyword of the document.

REFERENCES

[1] T. B. Mirani and S. Sasi, "Two-level text summarization from online news sources with sentiment analysis," 2017 International Conference on Networks & Advances in Computational Technologies (NetACT), Thiruvanthapuram, 2017, pp. 19-24.

[2] Jenq-Haur Wang, Jeng-Yuan Yang "Statistical Single-Document Summarization for Chinese News Articles"2012 26th International Conference on Advanced Information Networking and Applications Workshops.

[3] Kumodini V. Tate, Bhushan R. Nandwalkar "Document Recommendation Using Keyword Extraction for Meeting Analysis" International Journal of Computer Science and Information Technologies Vol. 7 (4) , 2016, 1763-1766

[4] Pooja Devi, Ashlesha Gupta "Comparative Study of HITS and PageRank Link based Ranking Algorithms" International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 2, February 2014

[5] Kumodini V. Tate, Bhushan R. Nandwalkar "Document Recommendation Using Keyword Extraction for Meeting Analysis" International Journal of Computer Science and Information Technologies, Vol. 7 (4) , 2016, 1763-1766.

[6] M. W. Berry, J. Kogan, Text Mining: Applications and Theory, Wiley, UK, 2010.

[7] Slobodan Beliga" Keyword extraction: a review of methods and approaches"

[8] Wu, A. M. Agogino, "Automating Keyphrase Extraction with Multi-Objective Genetic Algorithms, in Proc. of the 37th HICSS, pp. 104-111, , 2003.

[9] Y. Zhang, E. Milios, N. Zincir-Heywood, "A Comparison of Keyword- and Keyterm-based Methods for Automatic Web Site Summarization" in Tech. Report: Papers for the on Adaptive Text Extraction and Mining, pp. 15-20, San Jose, 2014.

[10] I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, C. G. Nevill-Manning, "Kea: Pra-ctical Automatic Keyphrase Extraction" in Proc. of the 4th ACM Conf. of the Digital Libraries, Berkeley, CA, USA, 1999.

[11] P. D. Turney, "Learning to Extract Keyphrases from Text" in Tech. Report, National Research Council of Canada, Institute for Information Technology, 1999.

[12] A. Hulth, "Improved Automatic Keyword Extraction Given More Linguistic Knowledge" in Proc. of EMNLP 2003, pp. 216-223, Stroudsburg, USA, 2003.