Context Score based Term Weighting Model for Text Summarization

Pratik Kamble
PICT, Pune-411043
Maharashtra, India

S. C. Dharamadhikari
Associate Professor
PICT, Pune-411043,
Maharashtra, India

ABSTRACT
Everybody is looking for relevant information briefly, which will cover information with small content. Summarization is the best for this. Current text summarization techniques do not consider the context i.e. background situation in that document. In this paper we are going to present the SentenceRank algorithm which will calculate the weight of the sentence based on the context score. We are going to make effective use of E-VSM : Enhance - Vector Space Model for bigram frequency count in whole corpus, where for each bigram we are going calculate the context score based on Bernoulli's model of randomness [1] [2]. Calculated bigrams context score is used in sentenceRank algorithm to calculate the context sensitive indexing weight of each sentence in a document. To reduce the redundancy in the sentences of summary, Cosine similarity measure is used to remove redundant sentence.

General Terms
Text Summarization, Bigram, Cosine Similarity.

Keywords
Context Score, E-VSM, SentenceRank

1. INTRODUCTION

World wide web is growing at intensive speed which is root cause to fire the outburst of information content. This information content is no longer possible and practical for a human to understand. To interpret this information there is need of automatic system and method for text summarization. Summarization is process of creating excerpts of text documents, technical papers and magazine article which serves the purpose of conventional abstract with no information loss fully by automatic means [3]. Summarization process basically falls under "abstractive" and "extractive". Abstractive based summarization model maximum time gives the summary by squeezing the text data and redeveloping allows summarizer to convey all the information without increasing the summary length. However these models requires complex linguistic processing. When summarization depends on need of user, it is query based or topic based summarization. Summarization can be of opinion type. When it is applied to multi-document it termed as multi-document summarization [1].

Term weighting is an indivisible part of modern text retrieval system. Term is common synonym for word, phrase or any other indexing unit taken hold to identify the contents of a text. different terms have different impact factor in text, an importance in text, this importance indicator- The term weight, is associated with every term. We should use this term weight efficiently to improve the text retrieval system. Till now people are focusing on the feature extraction, but development of term weight is necessary hand in hand with feature extraction to improve retrieval [4]. Term weighting comes in preprocessing stage of retrieval system. When doing summarization preprocessing is there. But nobody focuses on term weighting. Authors in summarization mainly gives importance to ranking algorithm. But evolution in term weighting for summarization can't be ignored.

There should be a term weighting method which will calculate context of a document, which will help to find topicality of the document, itself in the preprocessing stage. Basically context or topicality is nothing but a theme behind that document, a text matter surrounding index terms. This will definitely help to summarize the document at efficient level. Topicality or context of a document plays a major role in a summary to understand document in abstract. So there is need to develop a term weighting scheme which will calculate context and will find topicality. Ahead we discussed little about existing summarization system. This paper mainly focuses on extraction based summarization system. Till today some of the researchers tried to find out relation between tokens, but none have find topicality or context. We are proposing a system which will find the information content along with context, which is used in text summarization [1].

Remainder of this paper has been organized as follows. Section 2 discusses related work for summarization system and term weighting approach. In next section 3 talks about proposed system, followed by discussion and conclusion.

2. RELATED WORK

2.1 Text Summarization

As explained earlier summarization can either be "abstractive" or "extractive." This paper focuses on extractive summarization.

Extracting sentences from document having high weight and forming a summary is usual process text summarization, but according A. Nenkova et al.[5] with word frequency, Content word frequency, composition function and context sensitivity should be considered for effective summarization.

1. Content word frequency : Up to today word frequency is important measure for text summarization but content words are ignored, like noun, verbs and adjectives. frequency of content word will impact text summarization.

2. Composition function : Composition function will calculate the significance of sentences taking content word as function of importance which appear in respective sentence.

Product ($CF \equiv \Pi$) For this choice of *CF*.

$Weight(s_j) = \prod w_i \in s_j\, p(w_i)$

Average $(CF \equiv Avr\ )$ For this choice of *CF*.

$Weight(s_j) = \frac{\sum w_t \in s_j p(w_t)}{|\{w_t | w_t \in s_j\}|}$

Sum $(CF \equiv \sum\ )$ For this choice of *CF*.

$Weight(s_j) = \sum w_i \in s_j p(w_i)$

Where w_i = Input words, $p(w_i)$ = Probability of word distribution = $\frac{n}{N}$, *n* = No. of times word appeared and N = total word input.

s_j = Input sentence.

$Weight(s_j) = CF[p(w_i)]$ for $w_i \in s_j$.

High weight sentence extracted based on composition function until the summary length reached.

3. Context sensitivity : Avoid repetition of sentences in the summary.

N. stokes et al[6] proposed a method to produce better news heading through a news data with the help of lexical cohesion analysis. N. stokes shows when we go through a document the sentences are related to each other. That relation between sentences called as lexical cohesion and lexical coherence. Lexical cohesion is useful in making sentences from the text. And calculation of lexical coherence on the other hand shows is there any meaning in derived sentences. Lexical coherence requires expensive processing. Whereas lexical cohesion is easy to access. It is used to show relationship between sentences and useful in extracting sentences. The system is divided in Three parts

1. Tokenizer.

2. Lexical chain.

3. The sentence extractor.

1st it will divide text in token and will find relation between token. Form the lexical chain of that, then information gleaned from that is use by extractor to create a headline.

L. Li et al.[7] Proposed summarization which takes diversity, coverage and balance into account through structure learning. They focused on the extraction based summarization, enhancing following requirements : Summarization diversity, which reduces the redundancy in between sentences in the summary. Give importance to cover all the information in the respective document when deriving summary. And balance, summary should explain all the details of the document in the balanced way.

Manifold ranking approach and mutual reinforcement principle can be combined together as new approach for multi-document summarization [8][9].

Manifold ranking approach[9] :

1. Calculate information richness of each sentence.

2. Calculate overall ranking score of each sentence.

3. The sentence with high overall ranking score is chosen for summary.

Mutual reinforcement principle shows that extraction of significant sentences and key phrases at the same time. "A sentence should be weighted more if it is contained in the cluster which is more informative to the given query while a cluster should be weighted more if it contains many sentences which are more close to the query[8]."

2.2 Term Weighting Schemes

Information retrieval system has many term weighting schemes. Some of them are discussed below.

2.2.1 Term Frequency (TF) : Using this method [10], every term is suppose to have a value equal to the number of times it appear in a document is as follows:

$W(d, t) = TF(d, t)$ (1)

2.2.2 Term Frequency-Inverse Document Frequency (TF-IDF) : This approach[10] combined TF and IDF to weight the terms and better results were obtained compared with results of TF and IDF separately.TF-IDF expressed as

$W(d, t) = TF(d, t).IDF(t)$ *where,*

$IDF(t) = log(N/n)$ (2)

N = no. of total documents

n = documents in which term appear.

2.2.3 CHI-SQUARE : In [11] Youngjoong Ko, Jinwoo Park and Jungyun Seo explains the use of CHI^2 as the term weighting scheme. CHI^2 measures the lack of independence between a feature f and a category c [12]. Thus CHI^2 factor gives the goodness for a feature within the dataset for a category. Mathematically it is calculated as:

Table 1 General 2×2 contingency table

	Type1	**Type2**	**Total**
Category1	X	Y	X+Y
Category2	Z	W	Z+W
Total	X+Z	Y+W	X+Y+Z+W

$$x^2 = \frac{(XW - YZ)^2 (X + Y + Z + W)}{(X + Y)(Z + W)(X + Z)(Y + W)} \qquad (3)$$

The value of $\chi 2$ statistics values are calculated from training data.

2.2.4 Variance Relevance Factor : In [13], Xiaojun Quan, Wenyin Liu and Bite Qiu define a variant of traditional scheme for questions categorization. They have performed modifications in the TF scheme to normalize the effect of both a positive impact and negative impact to a category with respect to a feature.

$$W(f) = \log\left(\frac{fc + 1}{\overline{fc} + 1}\right) \qquad (4)$$

fc : No of documents belonging to category C_i in which feature f_i exists.

$f\bar{c}$: No of documents not belonging to category C_i in which feature f_i exists.

3. E-VSM : Enhanced-Vector Space Model

E-VSM is nothing but VSM, with important modification. As VSM is based on BOW model. Along with frequency of word in document it stores the positions of the words, this is important modification over standard VSM [14].

Following is E-VSM:

$$D_j = \{\{f_1,(p_1,p_2,\ldots\ldots Pf_1)\},\{f_2,(p_1,p_2,\ldots\ldots P_{f2})\},\ldots \ldots\ldots\{f_t,(p_1,p_2,\ldots\ldots P_{ft})\}\}$$

Where, D_j is vector representation of document j,
f_i is the frequency of the i^{th} term in document j
($p_1 \ldots\ldots p_{fi}$) represents the positions at which i^{th} term appears in document j.

4. PROPOSED SYSTEM

It is clear that when reading a text document, technical paper or magazine article it is not merely made up of sets of unrelated sentences, but these sentences do have relations with each other. They are of rich information content. When we read it, we easily come to conclusion regarding topicality of respective document and what is the context behind it. We can interpret the information content. But for machine to decide its topicality is challenging.

Main theme behind this summarization is to study the relation between terms, terms which co-occur together termed as bigram. As terms are occurring together there is information content in between them.

Many of tried to cover the bigram but none of them tried the approach of E-VSM[14]. We will use the concept of E-VSM to calculate the bigram frequency count through whole corpus followed by Bernoulli's model of randomness[1][2] to calculate the context score of each bigram and that respective score is used in traditional SentenceRank Algorithm to calculate the context sensitive ranking of each sentence in a document. And at last top rank sentences are extracted to form summary.

Thus we are proposing the combined approach of E-VSM for bigram count with Bernoulli's model of randomness for calculation of context score of each bigram used in SentenceRank algorithm for sentence score calculation. Combined used in text summarization.

So it is important to study their co-occurrence globally considering document set and deriving information content between them. And that information content should be utilized to make efficient summarization. Many of researchers have made use of bigram as rich information but none have derived information content in between them. Shown in the basic flow of system :

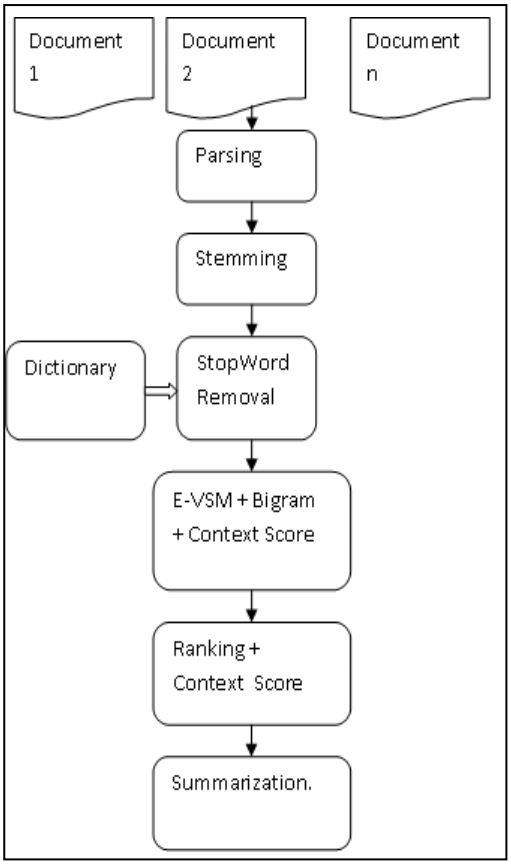

Fig 1: Proposed Flow Of System

4.1 Preprocessing

Preprocessing helps in removing noise and improving results. Preprocessing is done on the text document in which following steps are carried out :

Tokenization : This comes under the stage of parsing. The set of documents is taken. Each and every sentence in each document is tokenized.

Stemming : It is the process reducing inflected or derived words to their root, base or stem form. E.g Cats, catlike, catty as based on root "cat".

StopWords are common words which carry less importance, they too frequently occur in document. We should ignore words are like a, and, the, is, etc.

4.2 Bigram Frequency Count

E-VSM is made useful in remembering position of repeated word in document. Length of position vector is nothing but the appearance frequency of that particular word. These position vectors are searched to calculate how many times the bigram in document. Following is algorithm to calculate the bigram frequency.

In our approach we have little bit modified it, which is helpful studying the whole corpus and searching the corpus from all the corpus, which is important to know the presence of single bigram in whole corpus and hence how much is important the single bigram in respective corpus.

Algorithm 1 : Bigram Frequency Count Algorithm [14]

Input : Two elements of E-VSM i.e. a bigram
{ {f_1, $P_1(p_{11}, p_{12}, p_{13},\ldots.)$} , {$f_2$, $P_2(p_{21}, p_{22}, p_{23},\ldots.)$} }

Output: Frequency count of input bigram

initialize variable *bigramFreqCount* to zero
while P_1 or P_2 does not come to end **do**
  if element in P_1 is smaller than element in P_2 **then**
    if element in P_2 is one greater than element in P_1 **then**
      increment *bigramFreqCount*;
      go to next element in both P_1 and P_2;
    else
      go to next element in P_1;
  else
    go to next element in P_2;
end while
return *bigramFreqCount*;

4.3 Context Score Computation

When all bigrams are clear information content between them can be derived. For context score computation Bernoulli's model of randomness[1][2] is used.

There are few notations to be considered, number of documents in document set is *D*:

Documents : D = $\{D_1, D_2, \ldots\ldots, D_N\}$

w : Unique words per documents,

Let *T* = $\{t_1, t_2, \ldots\ldots, t_w\}$,

Document Frequency of term t_j = D_j, where D_j is no. of documents in which term t_j occur.

Probability of term t_i in document corpus = $P_i = \frac{D_i}{D_N}$, where term t_i occurs in documents D_i out of total documents D_N.

Probability of co-occurrence of term t_i and t_j in document corpus = $P_{co} = \frac{D_{ij}}{D_N}$, where term t_i and t_j co-occurs in documents D_{ij} out of total documents D_N.

Where, D_{ij} = denote the number of documents in which term t_i and term t_j co-occur.

According to classical semantic information theory[15] when a bigram occurs in a multiple document in document corpus we can derive the information content, which is presented in following formula (5) [1] :

$$Inf(Dij) = 0.5 log_2\left(2\pi D_{ij}(1 - D_{ij}/D_j)\right) + D_{ij} \log_2 p_{co}/p_i + \left(D_j - D_{ij}\right) \log_2 1 - p_{co}/1 - p_i \quad (5)[1]$$

where $p_i = D_i/D_N$ and $p_{co} = D_{ij}/D_N$.

Above presented formula helps to find the context score of the respective bigram form total corpus. Term *Inf* denotes the context score of bigram which is also interpreted as information content.

4.4 Sentence Rank

Generalized graph based SentenceRank algorithm is used to rank the sentences, i.e to weight the sentences. But here we are doing a little change in traditional graph based approach, we are considering context score which is calculated earlier in sentence rank algorithm.

Given a document, first task is to create the document graph where all terms in document are taken as the graph node and calculated context score is assigned to edge between nodes, the matrix is created with all row and column as document word, and if there is bigram respective bigram's context score is assigned to it. Graph let G = (V,E) V = $\{V_j | 1 \leq j \leq |V|\}$ denotes the set of vertices, E is edge. The context sensitive indexing weight of each word V_j in document D_i,denoted by *indexWt*(V_j) is to be calculated. It can be found by the recursive SentenceRank Algorithm, is as follows:-

Algorithm 2 : Sentence Rank Algorithm [1]

initialize *indexWt*[v_j] ← 1 ∀ *j, error E* ← 1.

while $E \leq \epsilon$

$E \leftarrow 0$

for *j* ← *1 to* |*s*|

memoWt[v_j] ← *indexWt*[v_j]

indexWt[v_j] ←$\mu.\sum_{\forall k \neq j} indexWt[v_k].\bar{E}_{kj} + 1 - \mu / |V|$

$E \leftarrow E + (indexWt[v_j] - memoWt[v_j])^2$

$E \leftarrow \sqrt{E}$

return *indextWt*

ϵ is error threshold and μ is damping factor.

Using this algorithm respective sentence score is calculated for a document.

5. RESULT & DISCUSSION

For experimentation text documents from news domain are considered. Newswire articles are considered. There are total 50 category newswire news articles. In each category 10 text documents are there, making total 500 text documents. The dataset is obtained from following URL link : http://dragon.ischool.drexel.edu/textsum.asp

When we consider the summary against the document, the length of the summary is far less than the length of the document. And for that the summary to be effective, it should convey proper information, without any loss of information. The summary which are creating has more topicality than that of document. In document there are many terms, many words which are non-topical and hence it is time consuming to read the whole document. Irrespective to that our proposed approach producing a summary which is having more topical term, and hence it is conveying proper information without any loss, and along with the summary is easy to read as number of sentences are less.

To measure the effectiveness of summary regarding topicality, we are using an evaluation technique:

For data set, the Bernoulli lexical association between the bigram terms was calculated using (5). For each document, the average lexical association between the document terms was calculated (stop words were not used). Similarly, average lexical association was computed for the calculated summary , produced by proposed approach. Where Average lexical association is calculation of topical terms in text document. Which calculates the topicality of the summary and notifies its effectiveness. For example, if a document/summary has M words (excluding the stop words), the average lexical association (avLex) [1] for the documents/summary was calculated as

$$avLex = \frac{\sum_{i=1}^{M}\sum_{j=1,j=1}^{M} A_{ij}}{M(M-1)} \quad (6)\ [1]$$

Where A_{ij} corresponds to the lexical association between *i*th and *j*th word in the document/ summary, which is calculated using a whole dataset.

Table 2:Comparison of topicality between Document and Proposed Summary

	Average Lexical Association (Topicality)
avLex Document	0.0007973359598517893
avLex Summary	0.00171587435702347

Table 3: Comparison of topicality between Proposed Summary and Traditional Tf-Idf Summary

	Average Lexical Association (Topicality)
Bigram Based Context Score(Summary)	0.00171587435702347
Traditional Tf-Idf(Summary)	0.0008865248226950354

From both the table we can see that the average lexical association of the calculated summary with respect to the respective document and the traditional summary of that document is more. Average lexical association of summary by proposed approach is remarkably better than document and traditional summary, this means proposed approach summary conveys more information.

Next we will discuss the effect of summary length on average lexical association :

Table 4 Relation between summary length and average lexical association

Summary Length (Sentences)	Average Lexical Association
24	0.656779
19	0.80153
16	0.831456
8	1.715874
6	1.55378
4	1.907985
3	1.683502

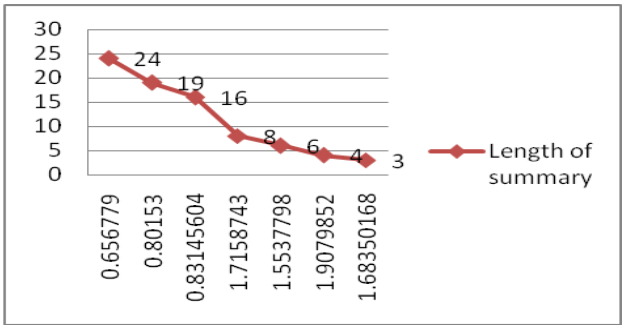

Figure 2 : Length of summary with respect to average lexical association

The above graph gives the relation between length of summary and the average lexical association. How the length of the summary causes average lexical association to vary. Variation is gradual as shown in the figure. From the figure we can see that the average lexical association increases as the summary length decreases. This happens because of the unnecessary word count decreases.

Table 5: Relation between sentence score threshold and average lexical association

Summary Length (Sentences)	Average Lexical Association
24	0.656779
19	0.80153
16	0.831456
8	1.715874
6	1.55378
4	1.907985
3	1.683502

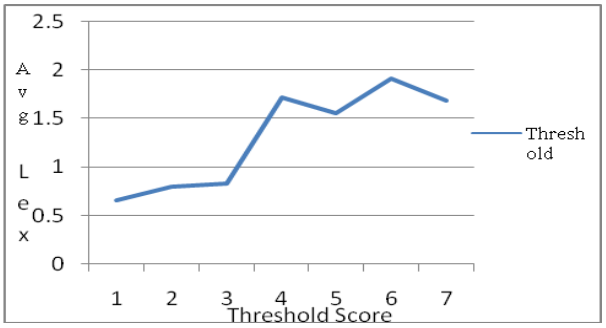

Figure 3 : Sentence score threshold with respect to average lexical association

The above graph gives the relation between sentence score threshold and the average lexical association. How the sentence score causes average lexical association to vary. Variation is gradual as shown in the figure. From the figure we can see that the average lexical association increases as the sentence score increases. This happens because, when sentence score increases the max sentence score sentence are get selected, due to which summary length decreases and

unnecessary word count get deceased which helps to improve average lexical association.

Table 6 : Relation between μ (damping Factor) and average lexical association

μ	Avg Lex Sum
0.65	0.945908987
0.75	0.945908987
0.85	1.292403461
0.95	1.469600917
1.05	1.593552474
1.15	1.593552474
1.25	1.593552474

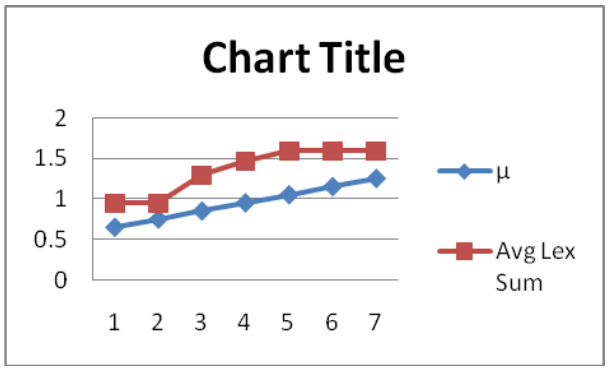

Figure 4 : μ (Damping factor) with respect to Avg lex Summary

From the above graph we can see that, as we increases the μ (Damping Factor) from the range 0.65 the average lexical association constantly increases and after 1.05 it remains constant. so we have kept μ = 1.05 for our summarizer. We get the gradual graph from this relation.

6. CONCLUSION

In this paper, we have proposed combined approach of E-VSM with Bernoulli's model of randomness and SentenceRank algorithm for text summarization. We have evaluated our summary with respect to the average lexical association. We have covered many aspects of the average lexical association with respect to length of summary and sentence score threshold value and the variation of the average lexical association is shown. We have concluded here with that the, context is very important in a summarization retrieval. For future scope we can extend this method for information extraction.

7. ACKNOWLEDGMENTS

We thanks Prof. M. Emmanuel, Head of Dept. of Information Technology, PICT, Pune for constant help and support.

8. REFERENCES

[1] P. Goyal, L. Behera and T. M. McGinnity "A Context-Based Word Indexing Model for Document Summarization," IEEE *Trans.* Knowl. Data Eng., VOL. 25, NO. 8 pp 1693-1705, AUGUST 2013.

[2] G. Amati and C.J. Van Rijsbergen, "Probabilistic Models of Information Retrieval Based on Measuring the Divergence from Randomness," ACM *Trans.* Information Systems, vol. 20, pp. 357-389, http://doi.acm.org/10.1145/582415.582416, Oct. 2002.

[3] H.P. Luhn, "The Automatic Creation of Literature Abstracts," IBM *J.* Research and Development, vol. 2, pp. 159-165, http://dx.doi.org/ 10.1147/rd.22.0159, Apr. 1958.

[4] N. Polettini, " The Vector Space Model in Information Retrieval- Term Weighting Problem" Dept. Inf. comm. Tech., Univ. Trento, Italy, 2004.

[5] A. Nenkova et al.,"A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization," *Proc.* 29th annual international ACM SIGIR conference on Research and development in information retrieval, pp.573 - 580, 2006

[6] N. Stokes et al. "Broadcast news gisting using lexical cohesion analysis," *Conf.* Information Retrieval, Volume 2997, pp 209-222, 2004.

[7] L. Li, "Enhancing diversity, coverage and balance for summarization through structure learning," *Proc.* 18th international conference on World wide web, pp.71-80, April 20-24, 2009, Spain.

[8] X. Cai and W. Li "Mutually Reinforced Manifold-Ranking Based Relevance Propagation Model for Query-Focused Multi-Document Summarization". IEEE Transactions On Audio, Speech, And Language Processing, Vol. 20, No. 5, July 2012.

[9] X. J.Wan, J. W. Yang, and J. G. Xiao, "Manifold-ranking based topic focused multi-document summarization," *Proc.* 18th IJCAI Conf., 2007, pp. 2903–2908.

[10] V. Murthy.G et al.," A comparative study on term weighting methods for automated telugu text categorization with effective classifiers," *J.* Data mining and knowledge management process, vol. 3, pp. 95-105, 2013.

[11] Y. Ko & J. Park & J. Seo , "Improving text categorization using the importance of sentences," *J.* Information Processing and Management, v.40 n.1, p.65-79, January 2004 [doi>10.1016/S0306-4573(02)00056-0]

[12] Y. yan, J. O. Pederson "Comparitive Study on feature selection in Text categorization," *Proc.* Fourteenth International Conference on Machine Learning (ICML 1997), Nashville, Tennessee, USA, July 8-12, 1997, ISBN 1-55860-486-3

[13] Y. Liu, H.T. Loh & A. Sun , "Imbalanced text classification: A term weighting approach," *J.* Expert Systems with Applications, Volume 36, Issue 1, January 2009, pp 690–701

[14] A. Bhakkad, S C Dharamadhikari & Parag Kulkarni, "Efficient Approach to find Bigram Frequency in Text Document using E-VSM," *J.* Computer Applications , 68(19):9-11, April 2013. doi .10.5120/11686-7356.

[15] J. Hintikka, "On Semantic Information," Physics, Logic, and History, pp. 147-172, Springer, 1970.