Comparison of Multi Document Summarization Techniques

R. Nedunchelian[1], R. Muthucumarasamy[2], E. Saranathan[3]

[1] Saveetha School of Engineering, Saveetha University
[2] Sri Venkateswara College of Engineering, Pennalur.
[3] Sastra University, Thanjavur-613402

Abstract:
This paper focuses on implementation and improvement of the existing MEAD ALGORITHM and Bayesian Classifier for Multi document summarization using Timestamp and the Frequent document concepts and found that summarization using Bayesian Classifier takes lesser time for the same set of inputs.

Keywords:
Timestamp, Frequent document, MEAD, Bayesian classifier

Automatic Summarization [2]

Automatic summarization is the process of taking information source as the frequently used documents, extracting content from it and presenting the most important content to the user in a condensed form and in a manner sensitive to the user's or application's need. It does not start with a predefined set of criteria of interest.

Single V/S Multi Document Summarization [3, 4]

There are two major differences between single and multiple document summarizations. First, most approaches to single document summarization involve extracting sentences from the document.. Second, most single document summarization systems, to a certain extent, make use of the monolithic structure of the document. For example, one simple but quite effective way to write a summary for a single document is to take the first sentence from each paragraph and put them together in their original order. However, for multi-document summarizations, the structure of a single document cannot be readily used in such a straight forward fashion. In this sense, multi document summarization systems usually rely less on the structures of the documents.

MEAD Extraction Algorithm [5]

MEAD is a publicly available tool kit for multi-lingual summarization. The toolkit implements multiple summarization algorithms (at arbitrary compression rates) such as position-based, Centroid, TF * IDF, and query-based methods.The score by MEAD algorithm for the sentence is calculated using equation 1.

$$\textbf{SCORE (Si)} = \sum(\textbf{wc Ci,k + wp Pi,k + wf Fi,k}) \quad \textbf{(1)}$$

Where,
Ci,k- Centroid value for sentence i
Pi,k- Positional value for sentence i
Fi,k- First Sentence Overlap for sentence i
wc , wp , wf are weights, these are the constant values assumed.

Redundancy Based Algorithm

We try to approximate the value by identifying sentence similarity across sentences. Its effect on MEAD is the subtraction of a redundancy penalty (Rs) for each sentence which overlaps with sentences that have higher SCORE values. This can be calculated from equation 2.

$$\textbf{SCORE(Si)} = \sum \textbf{(wc Ci,k + wP Pi,k + wF Fi,k) – wrRs} \quad \textbf{(2)}$$

For each pair of sentences extracted by MEAD, we compute the cross-sentence word overlap according to the following formula:

$$\textbf{Rs = 2 * (\#overlapping words) / (\#words in sentence1 + \#words in sentence2)} \quad \textbf{(3)}$$

Wr = Max (SCORE (Si))
Where SCORE (Si) is computed according to the formula in equation 3 penalty Rs from the previous score

$$\textbf{SCORE(Si)} = (\sum \textbf{(wc Ci,k + wP Pi,k + wF Fi,k)) – wrRs} \quad \textbf{(4)}$$

Naïve Bayesian Classifier [1, 8]

Kupeic et al. 1995 proposes a method of training a Bayesian classifier to recognize sentences that should belong in a summary [9]. The classifier estimates the

Manuscript received March 5, 2011
Manuscript revised March 20, 2011

probability that a sentence belongs in a summary given a vector of features that are computed over the sentence. It identifies a set of features that correspond to the absence/presence of certain words or phrases and avoids the problem of having to analyze sentence structure. Their work focused on analyzing a single document at a time. To be an informative summary, an abstract has to capture as much of the 'information content' as possible [10]. Keywords are useful tools as they give the shortest summary of the document. A frequently used multi document summarization system with user interaction that would extract a summary from frequently used documents using Naïve Bayesian Classifier with supervised learning is proposed.

Keyword Extraction Using Naive Bayesian Classifer [8]

Machine Learning techniques consider the keyword extraction as a classification problem. There are words (examples) in a document and the purpose is to identify whether a word belong to the class of keywords or ordinary words. As with other machine learning methods, we assume that there is a training set that can be used to learn how to identify keywords and using the knowledge gained from the training set, the unlabeled examples, which are the new documents in our case. Bayesian Decision Theory is a fundamental statistical approach based on the tradeoffs between the classification decisions using probability and the costs that accompany those decisions we obtain the metric TF *IDF (Term Frequency * Inverse Document Frequency) score, which is the standard metric used in Information extraction, and for a word W in document D, is defined as

TF*IDF(P,D)=P(word in D is W)*[-log P(W in a document)] **(5)**

The first term in this formula is calculated by counting the number of times the word occurs in the document and dividing it to the total number of words in it. The second term is calculated by counting the number of documents in the training set that the word occurs in except D and dividing it by the total number of documents in the training set.

Naive Bayes makes the assumption that the feature values are independent. With this assumption, we can compute the probability that a word is a key given its TF *IDF score (T), the distance to the beginning of the paragraph (D), paragraph where the word is present (PT) and the sentence that it exists in (PS) by using Bayes Theorem:

P(key|T,D,PT,PS)=((T|key)P(D|key)P(PT|key)P(PS|key)P(key)) / P(T,D,PT,PS) **(6)**

P(T,D,PT,PS)=Σ(P(T|key)P(D|key)P(PT|key)P(PS|key)P(key)) **(7)**

where P(key) denotes the prior probability that a word is a key (assumed to be equal for all words in our problem), P(T | key) denotes the probability of having TF * IDF score T given the word is a key, P (D | key) denotes the probability of having distance D , P(PT | key) denotes the probability of key with respect to the paragraph, P(PS | key) denotes the probability of key with respect to the sentence and P(T, D, PT, PS) denotes the probability that a word having TF * IDF score T, neighbor distance D, position in the text PT and position in the sentence PS. After calculating the probability value for each word, finally we have to calculate the score of each sentence. In order to calculate the score we have to add the probability value of each keyword in the sentence.

Score(S_i)=P(k1)+P(k2)+…….+P(k_n) **(8)**

Consider the below sentence "India is my Country". If we want to calculate the score for this sentence first find out the probability value for the keywords.

Assume P (India)= 0.2523 ,

P(Country)=(0.0232)

Score (s_i) = p (India) + p(Country) = 0.2755.

Finally Score value for the above sentence is obtained.

Frequent Documents [7]

Instead of taking up each sentence for comparison for summarization from all documents, it would be more than enough to summarize only the document which has been put to many numbers of readers. Since we track for the document which is read frequently by many people, it is supposed to provide all the necessary information about the topic to the user so the user need not surf through other documents for information as the document in hand would be satisfactory.

Timestamp [6]

The summary produced by MEAD contains the selected sentences from each document and output them in the order prevalent in the original document. Sentences selected from the first document will appear before the sentences selected from the second document, similarly selected sentences from the second document will appear before the sentences selected from the third document and subsequently. The order of the sentences in the summary may not be logical in occurrence. Hence to overcome this short coming the concept of Timestamp is implemented. The implementation of Timestamp is carried out by assigning a value to each sentence of the document depending on the chronological position in which it occurs in the document. Once the sentences are

selected they are arranged in the ascending order depending on the Timestamp. This gives the summary an ordered look, bringing out a coherent looking summary

Frequent Document Summarization with Timestamp Using MEAD and Naïve Bayesian Classifier [6, 7, 8]

Given the input as a cluster of documents on the same topic, the task of a frequently used document summarization system is to retrieve the frequently used documents and to generate a short paragraph that preserves the majority of information contained in the original documents. After performing the MEAD and Naïve Bayesian Classifier operation the final summary is generated based on Score and applying Timestamp. The number of sentences in the summary is dictated by the compression rate. For example if the compression rate is 10 percentages and the total number of sentences in all documents is equal to 100, then there will be 10 sentences in the summary.

Algorithm: Timestamp based summary generation

- For all the sentence in the cluster
- Begin
  - Sort the sentence in the descending order depending on the obtained score values after the reduction of the redundancy penalty.
- End
- Begin
  - Get the compression rate from the user
  - Select the required number of sentences based on the compression rate.
  - Sort the sentences in the ascending order depending on the timestamps
  - If the Timestamps are same
  - Begin
    - Compare the score values
    - Sentence with the higher score value will appear first
  - End
- End

Experimental Results and Discussion

20 Input documents are taken for summarization as sample inputs and are processed. The input documents are classified into sentences and they are processed using MEAD algorithm.

Automatic process has been developed and used to monitor and calculate the number of times the document has been read by the users. Input documents along with the number of times visited are calculated and the findings are given in Table 1.

Table 1 Input Documents and number of times visited

Documents Name	Number Of Times Visited
Input1.doc	11
Input2.doc	8
Input3.doc	7
Input4.doc	9
Input5.doc	11
Input6.doc	12
Input7.doc	12
Input8.doc	7
Input9.doc	10
Input10.doc	9
Input11.doc	8
Input12.doc	5
Input13.doc	6
Input14.doc	5
Input15.doc	10
Input16.doc	4
Input17.doc	11
Input18.doc	8
Input19.doc	10
Input20.doc	7

Frequent Document Selected For Processing

In the total of 20 documents, we have selected 10% (2 documents) of documents as frequently used documents for processing. Since the Input6.doc and Input 7.doc are visited more number of times they have been considered for further analysis. It is shown in Table 2

Table 2 Frequently visited documents

Document Name	Number Of Times Visited
Input6.doc	12
Input7.doc	12

Summary Generated By MEAD

The performance for summarization of the input documents using MEAD and Bayesian classifier has been analyzed and compared with frequent documents using MEAD and Bayesian classifier. Totally there are 100 documents. Among them 10% of documents are selected as frequent documents for processing using MEAD. The score table and the performance graph for frequent document summarization are shown below in table 3 and figure 1.

Table 3 Score table for frequent documents using MEAD

Number Of Documents	5	9	10
Time In Seconds	5	15	25

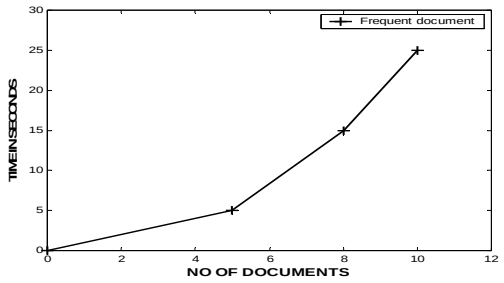

Figure 1 Frequent documents summarization using MEAD

From the above Figure 1, it is understood that when the MEAD is applied on the frequent documents the time taken to get the summary is only 25sec which is less than the time taken to summarize all the documents.

Totally there are 100 documents selected for processing using MEAD. For computational convenience and also for base of comparison of MEAD and Naïve Bayesian classifier the score table and the figure are shown only for 36 documents. The score table and the performance graph for multiple document summarization is shown below in Table 4 and figure 2.

Table 4 Score table for multiple documents using MEAD

Number Of Documents	10	25	30	34	36
Time in Seconds	10	40	45	70	110

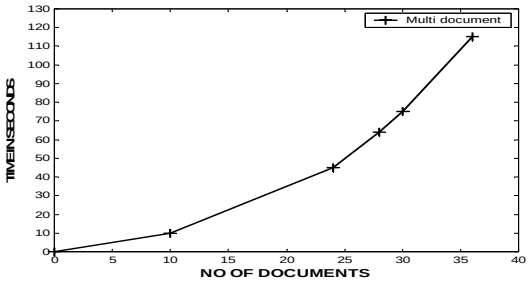

Figure 2 Multi document summarization using MEAD

From the above Figure 2, it is understood that when the MEAD is applied on the Multi documents the time taken to get the summary is 115sec which is higher than the time taken to summarize the frequent documents only.

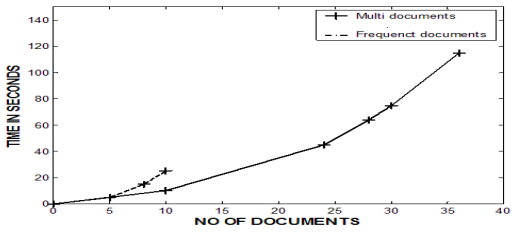

Figure 3 Comparison of frequent vs. multi document summarization using MEAD

Comparison of frequent vs multi document summarization using MEAD in Figure 3 shows that when the MEAD is applied on the Multi documents the time taken to get the summary is 115sec and the number of documents is 36 which is higher than the time taken to summarize only the 10 frequent documents within 25sec.So it is proved that there is improvement in summarization speed when the frequent document is considered instead of all the documents.

Summary Generated By Naïve Bayesian Classifier

Totally there are 100 documents. Among them 10% of documents are selected as frequent documents for processing using Naïve Bayesian classifier. The score table and the performance graph for document summarization is shown below in table 5 and figure 4.

Table 5 Score table for frequent documents using Naïve Bayesian Classifier

Number Of Documents	4	7	9	10
Time in Seconds	4	10	14	18

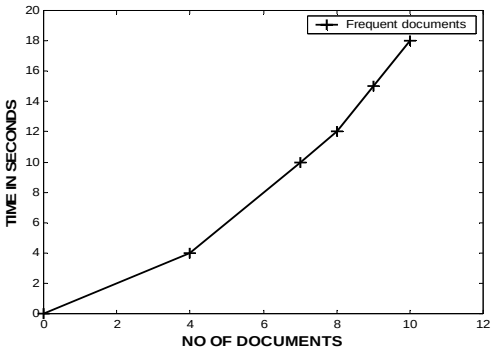

Figure 4 Frequent documents summarization using Naïve Bayesian Classifier

From the above Figure 4 it is understood that when the Naïve Bayesian classifier is applied on the frequent documents the time taken to get the summary is only 18sec which is less than the time taken to summarize all the documents. Totally 100 documents are selected for processing using Naïve Bayesian classifier. For computational convenience and also for base of comparison of MEAD and Naïve Bayesian classifier the score table and the figure are shown only for 36 documents. The score table and the performance graph for multiple document summarization using Naïve Bayesian classifier is shown below in table 6 and figure 5.

Table 6 Score table for multi documents using Naïve Bayesian Classifier

Number Of Documents	10	17	30	34	36
Time in Seconds	10	20	35	50	80

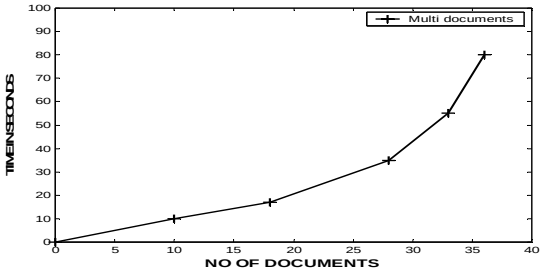

Figure 5 Multi document summarization using Naïve Bayesian Classifier

From the above Figure 5, it is understood that when the Naïve Bayesian classifier is applied on the Multi documents the time taken to get the summary is 80 sec which is higher than the time taken to summarize the frequent documents only.

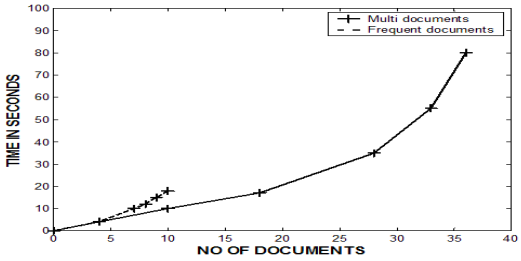

Figure 6 Comparison of frequent vs. multi document summarization using Naïve Bayesian classifier

The above Figure 6 shows that comparison of frequent and multidocument summarization using Naïve Bayesian classifier. Comparison of frequent vs. multi document summarization using Naïve Bayesian classifier shows that when the Naïve Bayesian classifier is applied on the Multi documents the time taken to get the summary is 80sec and the number of documents is 36 which is higher than the time taken to summarize only the 10 frequent documents within 18sec. So it is proved that there is improvement in summarization speed when the frequent document is considered instead of all the documents.

Comparison Of Frequent vs. Multi document Summarization using MEAD and NAÏVEBAYESIAN Classifier

The Figure 3 and 6 shows the comparison of frequent vs. multi document summarization using MEAD and Naïve Bayesian classifier. Comparison of frequent vs. multi document summarization using MEAD and Naïve Bayesian classifier shows that when the Naïve Bayesian classifier is applied on the Multi documents the time taken to get the summary is 80 sec and the number of documents is 36 which is less than the time taken by MEAD where the time taken is 115 sec for 36 documents. Also for summarizing only the 10 frequent documents the Naïve Bayesian classifier takes only 18 sec which is less than the time taken by MEAD where the time taken is 25 sec for 10 documents. So it is proved that Summarization of frequent documents using Naïve Bayesian classifier is better when compared to MEAD.

Table 7 Comparison of frequent vs. multi document summarization using MEAD and NAÏVEBAYESIAN classifier

Summarization Techniques	Multi Document Time In Seconds	Frequent Document Time In Seconds
Mead	115	25
Naïvebayesian Classifier	80	18

The above Table 7 compares the run time taken to summarize the documents using the two techniques. From the table it is found that run time for frequent document is faster than multi document using Naïve Bayesian Classifier.

Conclusion

Timestamp and Frequent Document Concept have been successfully implemented using MEAD and BAYESIAN CLASSIFIER to Generate the Multi Document summary. The results are compared and the summarizations using Bayesian classifier is found better than MEAD.

References

[1] Daume and Daniel Marcu (2005), "Bayesian Multi-Document Summarization", Proceedings of the ACL Workshop on Multilingual Summarization Evaluation, Ann Arbor, MI.

[2] Jade Goldstein, Vibhu Mittal, Jaime Carbonell, Mark Kantrowitz (2000),"Multi-Document Summarization By Sentence Extraction". Proceedings of the ANLP/NAACL Workshop, Seattle, WA,USA.

[3] Dragomir R. Radev, Hongyan Jing, and Malgorzata Budzikowska. (2000), "Centroid-based summarization of multiple documents: sentence extraction, utility based evaluation and user studies", Conference on ANLP/NAACL and Workshop on Summarization,Vol.4,pp21-30,Seattle, USA

[4] Dragomir R. Radev, Sasha Blair-Goldensohn, and Zhu Zhang (2001), "Experiments in single and multi document summarization using MEAD", First Document Understanding Conference, New Orleans, LA, USA

[5] Dragomir R. Radev, Hongyan Jing, Malgorzata Stys and Daniel Tam (2004), "Centroid based summarization of multiple documents", Journal of Information Processing and Management,Vol.40,No.6,pp 919-938.

[6] Nedunchelian R. (2008), 'Centroid based Summarization of Multiple documents Implemented Using Time stamps' proceedings of International Conference on Emerging Trends in Engineering & Technology, pp480-485, October 2008, Nagpur, India.

[7] Nedunchelian R.,Muthucumarasamy R. Saranathan E. (2009), 'Multi Document Text Summarization Techniques' International journal of Adv anced Research in Computer Engineering , Vol.3,No.1,pp-91-100.

[8] Nedunchelian R.,Muthucumarasamy R. Saranathan E. (2010), 'An Approach of the Naïve Bayesian classifier for the summarization of frequently used documents implemented using timestamps' International journal of Advanced Research in Computer Engineering, Vol.4,No.1,pp-53-60.

[9] Kupiec, J., J. Pedersen, and F. Chen, "A Trainable Document Summarizer," In Proceedings of the 18th ACM-SIGIR Conference, 1995, pp. 68-73.

[10] Otterbacher, J. C., A. J. Winkel, and D. R. Radev, The Michigan Single and Multidocument Summarizer for DUC 2002,

Nedunchelian Ramanujuam received the B.E Degree from Madras University in 1981 and M.E Degree from Anna University in 2004. Currently working as Professor in Saveetha University in Department of Computer Science and Engineering. He is doing doctoral research in the area of Data mining in SASTRA University.

R Muthucumaraswamy received B.Sc., Mathematics, from Gurunanak College, University of Madras in 1985. M.Sc., Applied Mathematics, from MIT, Anna University in 1987. M.Phil., Mathematics, from Pachaiyappa's College, University of Madras in 1991. Ph.D. (Theoretical and Computational Fluid Dynamics) from Anna university in 2001.Currently working as Professor and Head in the Department of Applied Mathematics at Sri Venkataswara College of Engineering.

E. Saranathan received B.Sc., Geology, from Madras University in 1983 M.Sc., Applied Geology, from Bharathidasan University in 1985. Ph.D. – Environmental Engineering and Science -Using Remote Sensing and GIS, from Indian School of Mines, Dhanbad, Jharkhand, INDIA in 2002. Currently Working as Senior Assistant Professor, School of Civil Engineering, SASTRA University, Thirumalaisamudram, Thanjavur.