EXTRACTIVE TEXT SUMMARIZATION BY FEATURE-BASED SENTENCE EXTRACTION USING RULE-BASED CONCEPT

Siya Sadashiv Naik
Department of Computer Engineering
Goa College of Engineering
siyanaik08@gmail.com

Manisha Naik Gaonkar
Department of Computer Engineering
Goa College of Engineering
manisha@gec.ac.in

Abstract— World Wide Web is a tremendous source of knowledge. Vast amount of information available over the internet has made the humans suffer from problem of information explosion. Therefore, a good mechanism is required to extract relevant information. This calls a need for an automatic and significant tool that converts lengthy documents into concise form by extracting relevant information from it. Automatic Text Summarizer serves as one of the best tool for interpreting lengthy textual content. It represents the shorter version of the original document by choosing most important part of text, thus generating its summary. It is classified into two categories: abstraction and extraction. This paper highlights on extractive approach. Main aim is to select best sentences by weighting them. We carried out our experiment on 15 documents from DUC 2002 dataset. Each test document was first pre-processed. Then, all the sentences were represented as attribute vector of features by calculating their scores. Rule-based method was proposed to select the best sentences. Results were compared with GSM summarizer and conclusion was drawn that best average recall, precision and f-measure values was obtained for Rule-Based Summarizer.

Keywords— Automatic Text Summarization, Pre-processing, Keyword Extraction, Feature Extraction, Rule-Based Method.

I. INTRODUCTION

Nowadays, all are highly dependent on internet. We use search engines to find information regarding an item. If a user is searching by entering a certain keyword, he will be overloaded with information which will waste his time and hence he will tend to lose his track. Automatic text summarization technology serves as the best solution for this problem. It generates a compressed version of source document by preserving its important text content and helps a person to quickly grasp large content of information. Automatic text summarization is an important area of research in field of Natural Language Processing (NLP) and Data Mining (DM).

We can summarize single document as well as multiple text documents. Summarizing a single document will take only one document as input. Whereas, summarizing multiple documents takes group of documents related to the same domain as input. Two techniques to summarize a document are abstraction and extraction. Abstractive summarization interprets the original text content, understands semantic relationship between sentences and produces summary. Whereas, extractive summarization retrieves only the most relevant sentences from source document thus maintaining low redundancy. Automatic text summarization finds its application in mass media area, search engines, news area, stock market area etc.

The main focus of this paper is to summarize a single text document and create its extractive summary. We have proposed a Rule-Based Summarizer. The paper is organized as follows: Section II investigates about the extractive summarization approaches proposed in the past. Section III states the proposed work. Section IV gives the results analyzed. Finally, Section V states the conclusion and future aspects.

II. EXTRACTIVE SUMMARIZATION APPROACHES

In this section, we summarize previous and current work going on in the field of text summarization and analyze various techniques to produce extractive summary of a single text document. Extractive Summarization extracts relevant sentences by assigning weights to the important regions of a document like sentences, paragraphs. These regions are then combined and ranked accordingly, to minimize redundancy.

Jasmeen Kaur and Vishal Gupta [1] proposed a statistical method which highlights how important a term is in a document by finding its term frequency (tf) and inverse document frequency (idf). Tf calculates frequency of a particular term and idf indicates its importance by dividing all documents to the number of documents in which that term occurs and then calculating the log of the quotient. Thus, tf-idf value increases proportionally to total number of times the word occurs in the document.

Vishal Gupta and Gurpreet Lehal [2] proposed a clustered based method that generates summary if the documents are of complete different topics. Documents are represented using tf-idf measures. Clustered documents are given as input to the summarizer. Once the documents are clustered, sentences are selected from the clusters to give the final summary.

Suanmali, Salem, Binwahlan and Salim [3] proposed fuzzy logic approach which calculates feature values for each sentence thus representing them as attribute vector of features. Sentences are then fed to fuzzy system where each sentence gets a score ranging from zero to one. Input membership function values are calculated. Rules required

978-1-5090-3704-9/17/$31.00 © 2017 IEEE

for summarization are also entered into the knowledge base. Top n sentences with high scores forms the final output summary.

Khosrow [4] proposed neural network approach. Neural networks are first trained to determine which sentences to include in final summary. This is done by a three-layered feed forward neural network. Once the network has learned about the features that must be present in summary sentences, next step is to discover relationship among the features. This is fulfilled by feature-fusion phase. All connections which consist of very small weights are pruned off without affecting the networks performance. Hidden layer activation values are clustered by using adaptive clustering technique for each hidden layer neuron. Once the above two steps are combined, feature effects are generalized and parameters to rank the sentences are provided.

Suanmali, Salim and Mohammed [5] stated about the General Statistical Method (GSM) to generate the summary. Pre-processed document is passed through feature extraction phase where feature scores are calculated thus representing each sentence as an attribute vector of features. Next, overall sentence score is calculated by summation of all feature values using the equation given below. Finally, sentences are sorted in descending order and top n sentences are selected as summary sentences.

$$\text{Score } (S) = \sum\nolimits^{n}_{k=1} S\_Fk(S) \text{ ------------------------(1)}$$

where Score (S) - score of the sentence S
S_Fk(S) - score of the feature k.

Pembe and Gungor [6] proposed query based approach where sentences scores are calculated based on the frequency of terms. Sentences which have query phrases are given higher scores than the ones which contain only single query words. These sentences which have high scores are part of final output summary.

Canasai and Chuleerat [7] proposed graph theoretic approach that provides a method for identifying key topics from document. Once the document has been pre-processed, each sentence from the document is represented as a node in an undirected graph. There is an edge between two nodes only if these nodes share some common words between them. If a node has more number of edges connected to it, then all the sentences present in that node have high preference to be included in the summary.

III. PROPOSED WORK

A. Problem Definition

Given a document, how will you generate a short summary of the entire document so that the user won't spend time reading the full document?

Suppose there is a document containing text. A software program will take this document as input and process it. During the course of processing, the document will be passed through pre-processing steps. Keywords will be extracted from the document and based on the threshold calculated, keywords will be pruned off. Further, the document will be subjected to various feature extraction methods where the sentences are represented as vector of features. A rule will be written and all the sentences will be passed through this rule. Finally, the sentences will be sorted and only the topmost sentences will be selected based on the extent of summarization defined to produce the summary.

B. Design

The proposed design for the Rule-Based Summarizer is explained below.

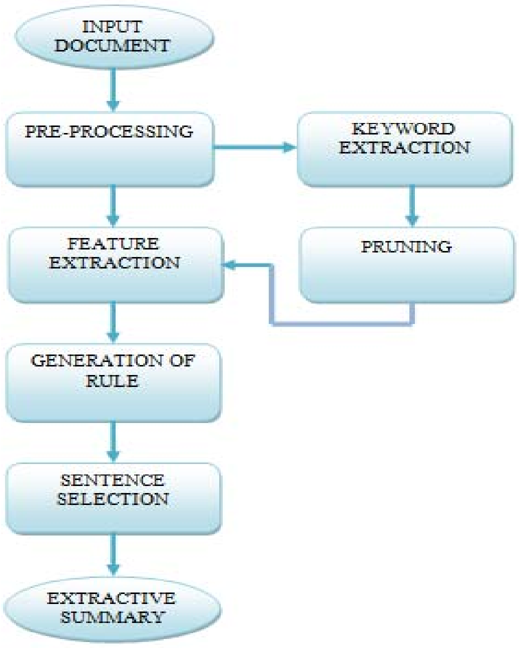


Figure 1. Text Summarization Architecture based on Rule-Based Concept

Each module of "Fig. 1"is explained below:

1) INPUT DOCUMENT: Input to summarizer are documents from Document Understanding Conferences (DUC) 2002 dataset [8]. Each document consists of 15 – 60 sentences with an average of 25 sentences. The dataset consists of human generated summaries provided by two different experts for each document.

2) PRE-PROCESSING: Pre-processing is the most primary step in any summarization method. Pre-processing is carried out to clean, remove noisy data and grammatical errors from document. Pre-processing methods applied are tokenization, stop word removal and stemming.

- TOKENIZATION: Tokenization [9] breaks down paragraphs into sentences and each sentence is broken down into individual words or tokens.
- STOP WORD REMOVAL: Data obtained after tokenization is analyzed and commonly occurring words or stop words [9] like a, an, the etc are removed from the document.
- STEMMING: In stemming [9], all words are reduced to its root form. For example, words like 'accept', 'acceptable', 'accepting', 'acceptances', 'acceptance', 'acceptation', 'accepted' are reduced to their root form 'accept'.

3) KEYWORD EXTRACTION: In keyword extraction [1] phase, frequency count of each word or a term in a document is calculated in order to find out its importance.

This is achieved by calculating (tf-idf) scores of the document.

4) PRUNING: In this stage, a threshold is defined. This value is calculated, by summing the term with lowest frequency and the term with the largest frequency thus taking their mean. Once threshold is calculated, all terms with tf less than the threshold value are pruned off from the document.

5) FEATURE EXTRACTION: After pre-processing step, each sentence of document is represented as attribute vector of features. Seven features are calculated for each sentence and each feature is given a value from 0 to 1 after normalization. Features considered are:-

- Sentence Position: This feature [10] [11], identifies the position of a sentence and calculates its importance in the document. Suppose the paragraph consists of 5 sentences than each sentence gets a value as follows:

$$F1\ (S_i) = 5/5 \text{ for 1st sentence,}\text{---------------------}(2)$$

$$4/5 \text{ for 2nd sentence,}$$

$$3/5 \text{ for 3rd sentence,}$$

$$2/5 \text{ for } 4^{th} \text{ sentence,}$$

$$1/5 \text{ for 5th sentence,}$$

$$0/5 \text{ for other sentences.}$$

- Title Feature: If a sentence consists of a word that is also present in title, than its score is given by the formula [10] [11]

$$F2\ (S_i) = \text{number of title words in the sentence/ number of words in the title}\text{-----------}(3)$$

- Numerical Value: If the sentences in the document consists of numerical data [10] [11], then these sentences reflect more important statistics and are likely to be selected for the summary.

$$F3\ (S_i) = \text{number of numerical values in the sentence/ length of sentence}\text{ ---------------------}(4)$$

- Keyword Weight: In this feature, keywords are all the words which are present after pruning stage. If the sentence consists of these keywords, that sentence weight is given by:

$$F4\ (S_i) = \text{number of keywords in the sentence/ length of sentence}\text{----------------------------------}(5)$$

- Proper Noun: If the sentence contains named entities like name of a person, place, week days etc then, that sentence might be important. Proper noun [10] [11] is calculated as:

$$F5\ (Si) = \text{number of proper nouns in the sentence/ length of sentence.}\text{---------------------------------}(6)$$

- Sentence To Sentence Similarity: This feature [10] [11], calculates the similarity of one sentence with all other sentences from the document. This is achieved by using cosine similarity measure.

$$Sim\ (S_i,S_j) = \textstyle\sum^n_{t=1} w_{it} \times w_{jt} \,/\, (\text{sqrt} \sum^n_{t=1} w_{it}{}^2) \times (\text{sqrt}\sum^n_{t=1} w_{jt}{}^2)\text{-------------------}(7)$$

where numerator - vector product of S_i and S_j
denominator - square root of square of terms of S_i and S_j

This score for a sentence is calculated by taking the ratio of summation of sentence similarity S to every other sentence over the maximum summation.

$$S(S_i)=\textstyle\sum Sim\ (S_i,S_j)\ /max(\sum Sim(S_i,S_j))\text{----------}(8)$$

where numerator - summation of sentence similarity S_i to every other sentence S_j
denominator - maximum of summation

The above equation is further normalized by diving it with maximum similarity value and final score of the sentence is calculated.

$$F6\ (Si) = S\ (Si)/\text{max similarity value}\text{--------}(9)$$

- Sentence Length: Short sentences in the document, are not expected to be in the summary since they will contain very less information. This feature [10] [11], is calculated as

$$F7\ (Si) = \text{length of sentence/length of longest sentence in document}\text{----------------------------}(10)$$

6) GENERATION OF RULE: Low and high values for each of the seven features are calculated. Next, single rule is written where all feature values are set to high except for sentence to sentence similarity as we need less similar sentences in the summary. Once the rule is written, all the sentences are passed through this rule. Each feature value of the sentence is mapped with each feature value of this rule. If there is a match, 0 is the output else output is 1. Finally, all 1's are counted which gives us number of mismatching features for that sentence, with reference to the rule. Each sentence now has a particular single score value.

7) SENTENCE SELECTION: All sentences are sorted in ascending order based on their scores. Extent of summarization is defined to retrieve sentences. Sentences are based on 20% extent; as it has been proved that extraction of sentences based on this percentage gives more informative summary as that of full document [12].

8) EXTRACTIVE SUMMARY: Final extractive summary of the document will be displayed.

C. Algorithm

Step 1: Document pre-processing.

Step 2: Calculate term frequency of all the terms

$$tf_{\ ij} =(n_{i,j})/\ \Sigma_k\ n_{k,j}\text{--------------------------}(11)$$

where $n_{i,j}$ - number of times term t_i occurs in document d_j
$\Sigma_k n_{k,j}$ - sum of total number of words occurring in a document d_j

Step 3: Define threshold

$$\text{threshold} = (\text{low+high})/2\text{----------------}(12)$$

where low - value of the term with smallest term frequency
high - value of the term with largest term frequency

Step 4: For each sentence S_i calculate

- Sentence position for each sentence
  for each sentence (Si) in paragraph of n sentences
  SP = n/n for 1st, (n-1) /n for 2nd, (n-2) /n for 3rd sentence and so on. ----(13)Title weight for each sentence

  for T_{ij} in S_i do,
  $TW= \Sigma T_{ij}(S_i)/\Sigma T_{ii}$--------------(14)
- Numerical value for each sentence
  for N_{ij} in S_i do,
  $NV= \Sigma N_{ij}(S_i)/\Sigma S_{iw}$------------ (15)
- Keyword weight for each sentence
  for K_{ij} in S_i do
  $KW = \Sigma K_{ij}(S_i)/\Sigma S_{iw}$------------(16)
- Proper noun for each sentence
  for P_{ij} in S_i do
  $PN = \Sigma P_{ij}(S_i)/\Sigma\ S_{iw}$-------------(17)
- Sentence to sentence similarity for each sentence
  for sentence S_i w.r.t. sentence S_j
  $\mathrm{Sim}\ (S_i,S_j) =\sum^{n}_{t=1} w_{it}\times w_{jt} /$ (sqrt $\sum^{n}_{t=1} w_{it}{}^{2}$) ×(sqrt$\sum^{n}_{t=1} w_{jt}{}^{2}$)------(18)

  Sentence score of sentence S_i
  $S(S_i)=\sum \mathrm{Sim}(S_i,S_j)/ \max(\sum \mathrm{Sim}(S_i,S_j))$--------------(19)

  $SS(S_i) = S(S_i)/$max similarity value --------(20)

- Sentence length for each sentence
  for each sentence S_i in document D
  $SL = \Sigma S_{iw}/\Sigma S_{Dw}$----------------(21)

Step 5: a) Calculate low and high value of each feature (F_i).

Low=(a+b)/2 (22)

where a - lowest value of a particular feature for all sentences
b - largest value of a particular feature for all sentences

High =   > Low-----(23)

b) Output 0 if Fi =Low
else
Output 1

Step 6: Represent each sentence as vectors of 0's and 1's.

Step 7: a) Write the following rule

If (F1= 1, F2=1, F3=1, F4=1, F5=1, F=0, F7=1) then sentence is important.

b) Map value of each feature of sentence S_i to value of each feature of rule.

c) If there is match,
Output =0
else
Output =1

Step 8: a) For each sentence S_i,

Score (S_i) =∑ no of 1's------------(24)

Step 9: Sort sentences in ascending order.

Step 10: a) Define extent of summarization.

(% of output summary)/100 ×N-----(25)

where N - total number of sentences present in the original document

b) Select topmost n sentences.

Step 11: Output extractive summary.

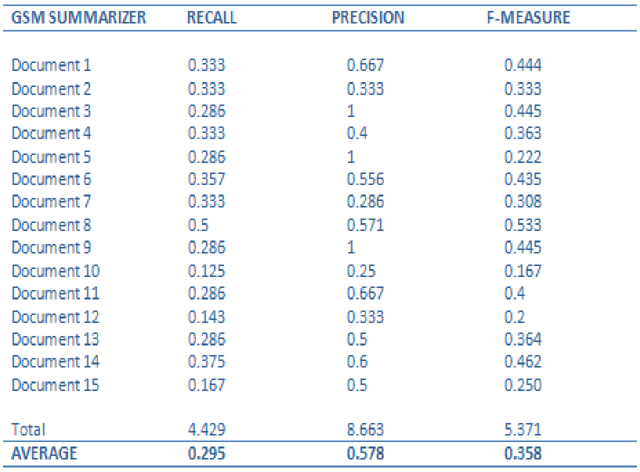
IV. EXPERIMENTAL RESULTS

Experiments were carried out on Intel core i5 processor, 64-bit OS and 4GB RAM. Programming Language used was JAVA and the platform was NetBeans IDE 8.0.2.

The proposed method and the existing GSM are implemented. Summaries produced by proposed approach are compared with summaries of GSM.

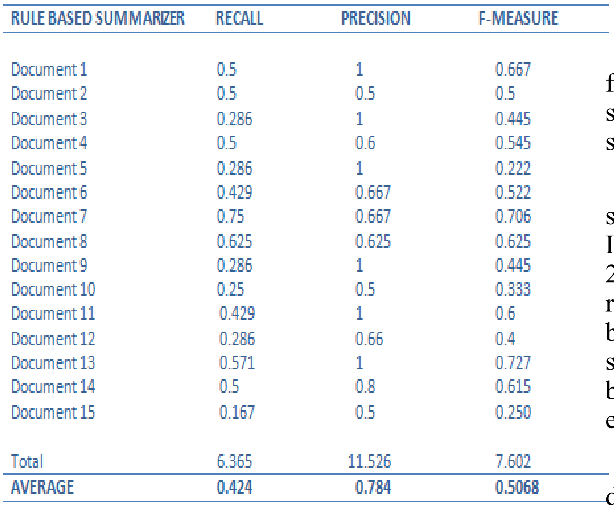
Three major methods for evaluating the performance of a system are recall, precision and f-measure [13]. Recall evaluates [13] summaries by measuring the proportion of relevant sentences present in summary. Precision measures [13] the proportion of correct sentences in summary. F-measure is calculated [13] by taking weighted harmonic mean of recall and precision values.

Recall, Precision and F-Measure values for 15 sets of documents from DUC dataset were calculated using GSM Summarizer and Rule-Based Summarizer. Results are shown below in the form of tables.



GSM SUMMARIZER	RECALL	PRECISION	F-MEASURE
Document 1	0.333	0.667	0.444
Document 2	0.333	0.333	0.333
Document 3	0.286	1	0.445
Document 4	0.333	0.4	0.363
Document 5	0.286	1	0.222
Document 6	0.357	0.556	0.435
Document 7	0.333	0.286	0.308
Document 8	0.5	0.571	0.533
Document 9	0.286	1	0.445
Document 10	0.125	0.25	0.167
Document 11	0.286	0.667	0.4
Document 12	0.143	0.333	0.2
Document 13	0.286	0.5	0.364
Document 14	0.375	0.6	0.462
Document 15	0.167	0.5	0.250
Total	4.429	8.663	5.371
AVERAGE	**0.295**	**0.578**	**0.358**

TABLE I. RESULT OF GSM SUMMARIZER



RULE BASED SUMMARIZER	RECALL	PRECISION	F-MEASURE
Document 1	0.5	1	0.667
Document 2	0.5	0.5	0.5
Document 3	0.286	1	0.445
Document 4	0.5	0.6	0.545
Document 5	0.286	1	0.222
Document 6	0.429	0.667	0.522
Document 7	0.75	0.667	0.706
Document 8	0.625	0.625	0.625
Document 9	0.286	1	0.445
Document 10	0.25	0.5	0.333
Document 11	0.429	1	0.6
Document 12	0.286	0.66	0.4
Document 13	0.571	1	0.727
Document 14	0.5	0.8	0.615
Document 15	0.167	0.5	0.250
Total	6.365	11.526	7.602
AVERAGE	0.424	0.784	0.5068

TABLE II. RESULT OF RULE- BASED SUMMARIZER

Following are the graphs obtained for recall, precision and f- measure values.

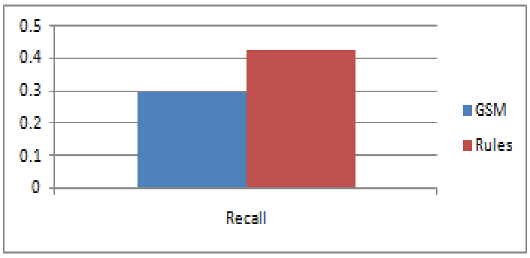


Figure 2. Recall

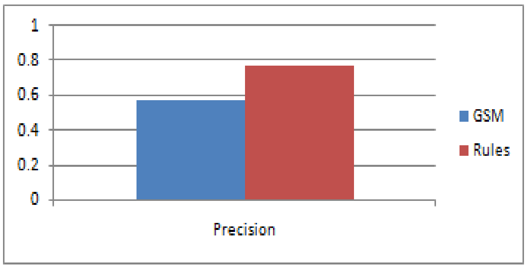


Figure 3. Precision

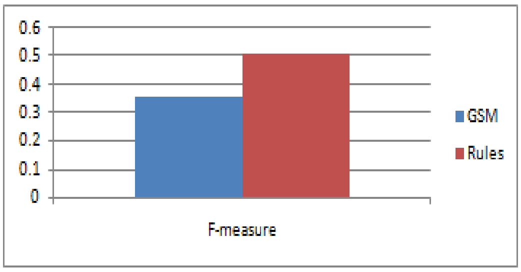


Figure 4. F-Measure

V. CONCLUSION AND FUTURE WORK

Automatic Text Summarization is a complex task in the field of information retrieval. In extraction based text summarization, important part is identification of relevant sentences from source document.

A Rule-Based Summarizer was proposed which gives summary of the document with better information coverage. Inputs to the summarizer were 15 news articles from DUC 2002 dataset. Each sentence of the document was represented as attribute vector of features. Result produced by this summarizer was compared with existing GSM summarizer. It was seen that proposed summarizer gives better average recall, precision and f-measures vales than existing summarizer.

Proposed method was used to summarize only a single document text. As future work, this method can be extended to summarize multiple documents. The proposed method could also be combined with existing learning methods for large dataset.

REFERENCES

[1] Jasmeen Kaur and Vishal Gupta, "Effective Approaches for Extraction of Keywords". International Journal of Computer Science Issues (IJCSI), vol. 7, issue 6, November 2010.

[2] Vishal Gupta and Gurpreet Singh Lehal, "A Survey of Text Summarization Extractive Techniques". Journal Of Emerging Technologies In Web Intelligence, vol. 2, no. 3, August 2010.

[3] Ladda Suanmali, Mohammed Salem, Binwahlan and Naomie Salim, "Sentence Features Fusion for Text Summarization using Fuzzy Logic". Ninth International Conference on Hybrid Intelligent Systems, IEEE, 142-145, 2009.

[4] Khosrow Kaikhah, "Text Summarization using Neural Networks". Proceedings of Second International Conference on Intelligent Systems, IEEE, 40-44, Texas, USA, June 2004.

[5] Ladda Suanmali, Naomie Salim and Mohammed Salem Binwahlan, "Fuzzy Genetic Semantic based Text Summarization". International Conference on Dependable, Autonomic and Secure Computing, IEEE, 2011.

[6] F.Canan Pembe and Tunga Gungor, "Automated Query-Biased and Structure-Preserving Text Summarization on Web Documents". Proceedings of the International Symposium on Innovations in Intelligent Systems and Applications, Istanbul, June 2007.

[7] Canasai Kruengkari and Chuleerat Jaruskulchai, "Generic Text Summarization using Local and Global properties of Sentences". Proceedings of the IEEE/WIC International Conference on Web Intelligence (WI'03), 2003.

[8] DUC. Document understanding conference 2002 (2002), http://wwwnlpir. nist.gov/projects/duc

[9] Dharmendra Hingu, Deep Shah and Sandeep S. Udmale, "Automatic Text Summarization of Wikipedia Articles". International Conference on Communication, Information & Computing Technology (ICCICT), IEEE, 2015.

[10] Ladda Suanmali, Naomie Salim and Mohammed Salem Binwahlan, "Feature-Based Sentence Extraction using Fuzzy Inference Rules". International Conference on Signal Processing Systems, IEEE, 2009.

[11] Sravani Chintaluri, R. Pallavi reddy and Kalyani Nara, "Fuzzy Approach for Document Summarization". Journal of Information, Knowledge and Research In Computer Engineering, Nov 14 - Oct 15, Volume – 03, Issue – 02.

[12] G. Morris, G.M. Kasper and D.A. Adam, "The Effect and Limitation of Automated Text Condensing on Reading Comprehension Performance". Information System Research, 3 (1), pp.17-35. 1992.

[13] Saeedeh Gholamrezazadeh, Mohsen Amini Salehi and Bahareh Gholamzadeh , "A Comprehensive Survey on Text Summarization Systems".IEEE,2009