International Journal of Pure and Applied Mathematics
Volume 118 No. 22 2018, 443-450
ISSN: 1314-3395 (on-line version)
url: http://acadpubl.eu/hub
Special Issue

ijpam.eu

EXTRACTIVE TEXT SUMMARISATION TECHNIQUES

[1]Kandukuru Sai Bhavana, [2]Aditya Deshmukh, [3]K. Deeba
Department of Computer Science and Engineering, SRM University
Kattankulathur, Kanchipuram District, Chennai, India
[1]ksaibhavana@yahoo.com, [2]adideshmukh17@gmail.com
[3]deeba.k@ktr.srmuniv.ac.in

Abstract: Text Summarisation is the process of extracting important information from large documents, and producing a shorter version that precisely summarizes the text. There are two types – Extractive Text Summarisation and Abstractive Text Summarisation. Extractive Text Summarisation extracts important words, key phrases, and sentences, and forms short summary using the same. On the other hand, Abstractive Text Summarisation uses techniques such as Natural Language Generation to create an abstract synopsis by paraphrasing the sentences. TextRank is the most common implementation for Extractive Text Summarisation. The idea of TextRank is to score the sentences in the document and use the top ranked sentences to form the summary. This paper presents an approach to perform Extractive Text Summarisation along with Question-Answering on a set of news articles under a closed domain. Question-Answering is a concept under the field of Natural Language Processing for generating appropriate answers to the various questions asked. The questions can be under either Closed Domain or Open Domain.

Keywords: text summarisation; extractive; abstractive; TextRank; unsupervised methods; question-answering; NLP;

1. Introduction

With the massive amount of data that is floating around the internet, a shorter, quicker source of information is required now more than ever. Text summarisation is a technique that gathers these huge blocks of text, and produces a concise version which delivers the most important information.

These summarisation techniques have been categorised based on various features. Based on the number of documents summarised, they are classified [1] as Single and Multiple Document Summarisation. Single Document Summarisation summarises a single document, whereas Multiple Document Summarisation summarises multiple documents, generally based on the same topic. Based on the purpose of summarisation, these techniques are classified as Generic, where the text is summarised in general without confining to a specific topic, Domain Specific, where the summaries are focused on the topic in focus, and Query-Based, where the summary tries to answer the question posed in the query. Finally, based on the output, these techniques are classified as Extractive and Abstractive text summarisation. Extractive text summarisation technique picks the linguistic objects (sentences, words, or phrases) that convey the message delivered by the text as a whole, and combines them to form an accurate summary. On the other hand, Abstractive Text Summarisation generates summary after comprehending the text using Natural Language Processing techniques. These techniques requires the use of well-trained sequential models and thus harder to implement when compared to Extractive Text Summarisation Techniques.

In general, all these techniques perform the following actions: [2]

i. To construct an intermediate representation that delivers the main aspects of the text. This can either be a topic representation, which focuses on the domain of the text, or indicator summarisation, which focuses on linguistic and statistical aspects of the test.

ii. Assign an importance score to sentences present in the text, which indicates how well the sentence represents the main idea of the text or how high the score is based on the indicators, and thus, how appropriate it is to use the sentence in the summary.

iii. Select k sentences to be used to form the summary for the given text by using greedy algorithms or optimisation techniques to increase accuracy and reduce redundancy and irrelevance.

Question-Answering (QA) Systems are developed to perform the task of responding to queries posed by the humans regarding any document. Thus, by being able to automatically answer a question that is posed by the human in the natural language form it simplifies the human work and only presents the necessary information rather than the entire document. Over the recent times, wide success and significant ability of these systems has been observed.

The objective of a Question-Answering System is to present a short, simplified and summarised response

to the input query rather than having to scan the complete document or the article. The user can either request for a direct answer to the query or get the most important, significant sentences related to the query in the document. These Question-Answering systems are classified into two types: Closed Domain and Open Domain. The Closed Domain Question-Answering System answers questions based on a specific domain on which it has been trained on. The Open Domain Question-Answering System is trained to answer questions related to any domain, and it is not related to a single field.

2. Proposed Work

In this paper, we present the implementation of a Closed Domain Question-Answering model and perform the Extractive Text Summarisation Technique to summarise a set of news articles. The implementation is carried out using the various tools and techniques of Natural Language Processing. Along with the implementation process, we present the experimental results of the model.

3. Related Work

Text summarisation was directed at making the task of understanding the documents easier. Mainly, the documents that are used for the summarisation are Legal Documents, New Articles, Contents of Text Books and Novels. In the past, most of the work in this field was directed towards Technical Documents. Hans Peter Luhn, a researcher for IBM and an innovator of Information Retrieval concepts, was the first to publish a paper in this field. The work done by Luhn suggested using the frequency of a word in the document to figure out its importance. As an initial step, the tokenization of the words was done to achieve the root words and remove the stop words from the document. Then the extracted words were arranged in an order of decreasing occurrence along with its frequency as the index. The occurrences of these words in a sentence helped in its ranking. The top ranked sentences were selected as the extracted summary of the document.

Over the years, Text Summarisation has been viewed under Natural Language Processing, Machine Learning and Statistical Modelling. While the Machine Learning methods have been using the Naïve-Bayes methods, recent improvements have seen the use of neural networks and algorithms that make no assumptions. [3]

The history of Question-Answering dates back to 1978 when Lehnert proposed a system based on semantics and reasoning. After the inception of the concept of Question-Answering the earliest systems to be developed and designed were BASEBALL and LUNAR. As the name suggested, BASEBALL answered questions related to the baseball leagues and LUNAR was developed to respond to queries about rocks examined in the Apollo missions.

More recent work in this field of computer is focussed on Open Domain. For example, answering open ended queries by accessing data from Wikipedia.

4. Summarisation Features

Various techniques have been devised in the past few years in the field of extractive text summarisation. The fundamental concept involves scoring sentences based on certain features. Commonly used features are described below. [4]

A. Keyword Feature

This feature enables the summariser to extract sentences that contain keywords, as keywords denote the main idea of the text.

B. Title Word Feature

Sentences that contain the words contained in the title are important. Thus, this feature helps extract such sentences to be included in the summary.

C. Cue Phrase Feature

Certain phrases that indicate the main idea or summary imply that those sentences could be used in the result.

D. Term Frequency

Sentences with words that most commonly occur could summarise the text well. This is accurately measured by TF-IDF frequency.

E. Location

Sentences that occur at common locations, such as the beginning or the end, could be a good choice for summarisation.

F. Similarity

Similarity between one sentence from the text and all other sentences or the title itself can be used as a measure. Sentences with high similarity scores can be used to form the summary.

5. Text Summarisation Implementation Techniques

The Extractive Text Summarisation can be done using both Supervised and Unsupervised methods. While the supervised learning methods can provide highly accurate results, they are difficult to implement. Unsupervised methods on the other hand, don't need pre-trained data and thus making the implementation process easier. They make use of algorithms that do not require an input to summarize text. These methods are best suited for complete automation. The following are some of the well-known and tested techniques under both Unsupervised and Supervised methods: [5]

A. Graph-Based Method

Sentences from the text, after pre-processing, are represented as nodes in an undirected graph. Algorithms such as Iterative Ranking algorithm and LexRank are implemented using this technique. LexRank, in specific, uses the concepts of Eigen Vectors and cosine similarities. The sentences are then ranked based on the similarity groups that are formed. [6]

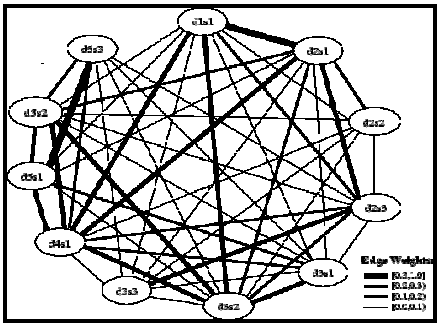


Figure 1. LexRank algorithm

B. Concept-Based Method

This approach uses an external concept knowledge-base. The sentences are matched against the concepts, thus building a vector model similar to graph-based approach. The sentences are then ranked and chosen for summary. Similarity measures are used to reduce redundancy and increase accuracy of the summary.

C. Fuzzy Logic-Based Method

The fuzzy system uses various text statistical inputs such as sentence length and similarities, and ranks sentences that can be used to summarize the text.

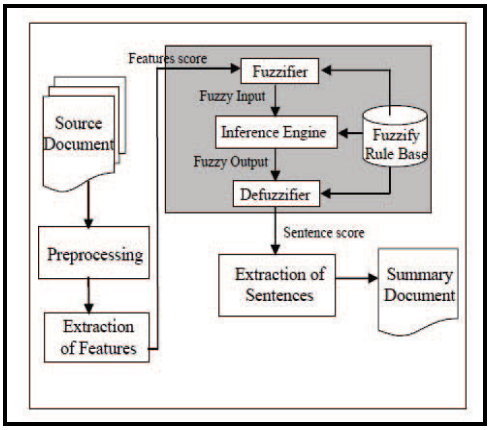


Figure 2. Fuzzy based approach [6]

D. Neural Network Approach

This approach uses multi-layered neural networks to firstly, label the data appropriately, and then learn to extract features from summary sentences. Some systems also make use of the statistical features, where the neural network learns to score sentences based on these features. Sentences that are extracted are used to form coherent summaries. This technique works much better than the other techniques due to its learning ability. [7]

6. Algorithms

Text Summarisation

Step 1: Create a closed domain dataset. And store the data in a dictionary or list.

Step 2: Segregate the article into sentences and tokenise the sentences.

Step 3: Create an undirected graph of sentences.

- Consider N sentences with links between each other
- P[i][j] is the probability of sentence i and j to be linked or similar
- If there is a link between sentence i and j, then P[i][j] is initialized with 1.0/ number of outbound links from the vertex of sentence i.
- If a sentence i has no outbound links, then P[i][j] = 1.0/N.

Step 4: Check for the similarity index of two sentences (I and j).

$$similarity(i,j) = \frac{|\{word|word\ in\ i\ and\ word\ in\ j\}|}{(|i|+|j|)/_2}$$

Step 5: Generate the summary based on the similarity index calculated.

Question-Answering

Step1: Create a dataset and tokenize its content.

Step 2: Get the input question from user and perform the pre-processing tasks.

Step 3: Use the spacy library to extract the subject from the question.

Step 4: Create a list of sentences which include the subject of the question.

Step 5: Select the most significant sentence by creating a dependency tree and parse the node until the question target is reached.

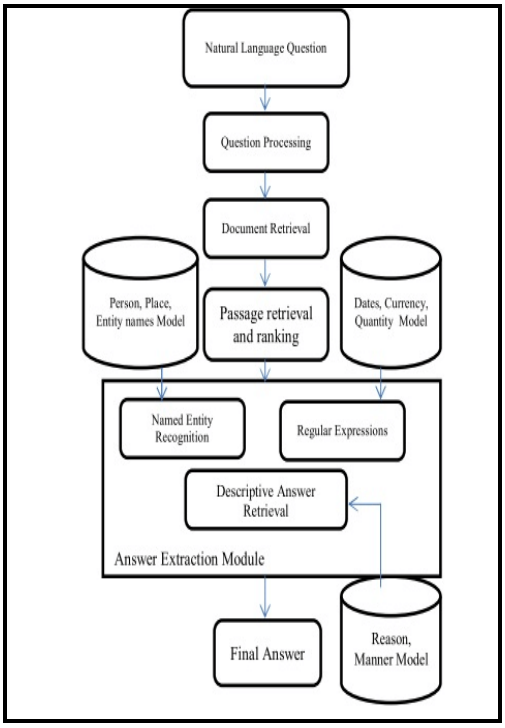


Figure 3. Question-Answering System Algorithm in a Flowchart

7. Explanation of Text Summarisation Algorithm

The Text Summarisation algorithm is very similar to TextRank, which uses the concept of PageRank algorithm. In TextRank algorithm, a graph is generated with sentences as the vertices and the similarity between the sentences is the weight of the edge.

A. Dataset

The dataset consists of around 4000 news articles collected from multiple news sources. The user is provided with the chance to read any article from the database. The summariser summarises the article chosen by the user.

B. Sentence Segregation

The article is retrieved from the dataset. From the retrieved article various sentences are identified and segregated. This is done using the NLTK's Punkt model.

C. Pre-processing Tasks

General pre-processing tasks are carried out on the segregated sentences. They include: Stop Word Removal and Stemming.

Stop removal is the process of removing all the stop words like "a, and, the, this, etc." from the sentence. Stemming is the process of converting a word to its root form. For example: Singing, sang is converted to sing.

D. Tokenising the sentences

The retrieved sentences are then tokenized into a collection of the words in the sentence by using the Scikit-learn text extraction feature. This creates a matrix of sentences and words.

E. Creation of sentence graphs

The matrix created is now normalised using a Tfid Transformer and then the graph with normalized weights is generated. The graph generated has each sentence as its node and the similarity between the two sentences becomes the weight of the edge connecting those two sentences.

$$TF\,(word) = \frac{\text{No. of times the word appears in a sentence}}{\text{Total number of words in the sentence}}$$

$$\text{IDF (word)} = \log * \frac{Total\ no.of\ sentences}{No.of\ sentences\ with\ the\ word\ in\ it}$$

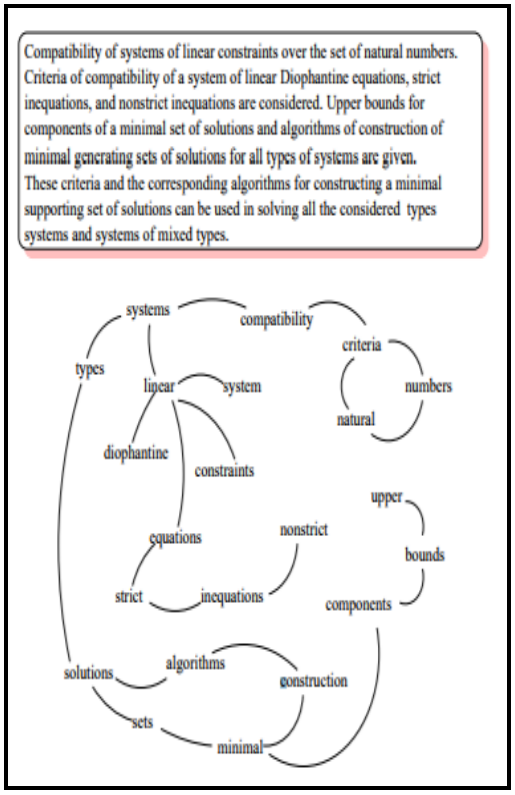


Figure 4. Sample graph for a given statement [8]

F. Similarity Index

The similarity index of sentences in Natural Language Processing is generally carried out by using the cosine similarity. The similarity is checked by comparing the tokenized and pre-processed words of the two sentences. Thus, more the number of common tokens, higher will the similarity index. The NLTK library uses the Cosine distance feature to check for the similarity. The cosine distance is the exact opposite of cosine similarity. Hence, it checks how far the two sentences from each other are. If the two sentences are completely similar, then they would have a cosine distance of 0. Formula for Cosine Distance is:

Cosine Distance = 1- Cosine Similarity

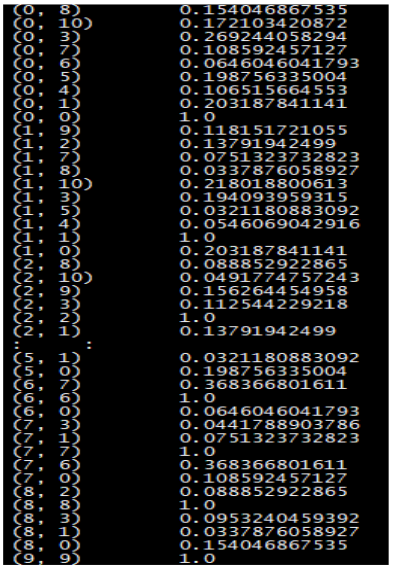


Figure 5. Similarities between two sentences

For the sentences, S1 and S2, to have a good similarity measure, any of the following conditions must be true:

- 0 <= Similarity(S1,S2) <= 1
- Similarity(S1,S1) = 1
- Similarity(S1, S2) != 1 if A != B

G. Summary Generation

The graph that is created is now stored as a dictionary, with each sentence being a key and the values are the similarity indices associated to that sentence.

To calculate the best possible sentence as the summary, the sum of the indices for each sentence is calculated and stored as a two-dimensional array. The sentence with the highest similarity index or the weights is chosen as the summarising sentence of that article.

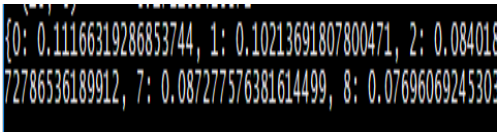


Figure 6. Sentence Scores

8. Explanation of Question-Answering Algorithm

A. Dataset, Query, and Pre-processing

The dataset is a collection of multiple news articles in a CSV format. The data set is designed as a closed domain QA System. The query is entered by the user depending on the article. The contents of the CSV file are extracted to various lists. For example: A list containing the articles is created. These articles are then pre-processed to be converted into a form that would support Answer Extraction. The article is classified into sentences and paragraphs.

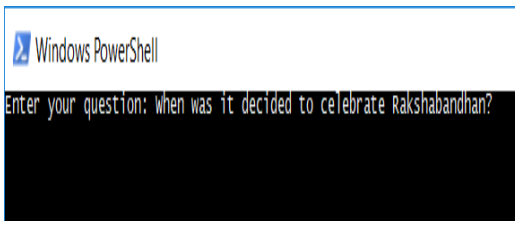


Figure 7. Question entered by user

B. Question Pre-processing

The user-entered query is pre-processed by performing pre-processing tasks of PoS tagging, stemming and stop word removal. [9]

1) Word Tokenisation: The words in the question sentence are split into individual words after removing the punctuations. This requires the use of NLTK's tokenize function.

2) PoS Tagging: Part of Speech (PoS) tagging is the process in which the tokenised words in the questions are classified based on its type. For example: Verb, Noun, and Pronoun. For the tagging, the Stanford PoS tagger library is used.

3) Stop Word Removal: The stop words in the question are removed. The stop words in the English language are: "A, An, The, That, etc"

4) Lemmatize or Stemming: The words are reduced to their root form before the process of indexing. For example: Ran, Running, and Runner all are converted to Run. The NLTK library's stem is used here.

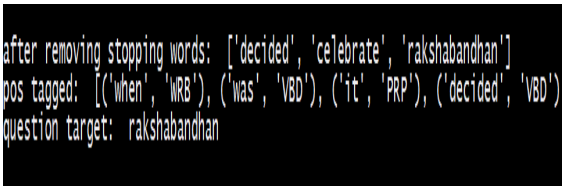


Figure 8. After pre-processing

C. Subject Extraction:

Using the spacy library, the target in the question is figured out. Spacy is a library built with pre-trained statistical models for advanced Natural Language Processing. It uses convolutional neural networks for the tasks of PoS tagging, parsing, and Named Entity Recognition (NER). It can also be used for Deep Learning Integration. The Named Entity Recognition is a pre-trained Machine Learning model/algorithm that finds and classifies entities such as Person, Location, Time, and Quantity in a sentence. The Named Entity Recognition systems have been formed to use linguistic grammar-based techniques and statistical models.

D. Sentence Selection:

After the subject extraction process in which the question target is identified, a list of sentences containing the identified target is created. Within this list, the sentence containing maximum occurrences of the words left in the question after pre-processing tasks is found. This sentence is the answer.

E. Answer Extraction

In order to extract the answer from this sentence, a dependency tree is created. This tree is parsed from the root until a node that is present in the question without the stop words is found. This node is considered as the new node and the sub-tree is parsed again in in-order traversal. Now an empty string is concatenated with the existing node value during the parsing. [9]

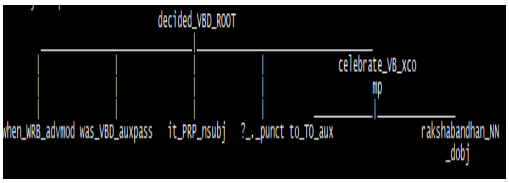


Figure 9. Tree Generation

F. Answer

The final concatenated string contains the precise answer from the sentences.

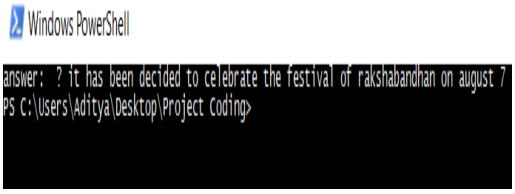


Figure 10. Answer

9. Evaluation and Results

Most methods for evaluating text summarizers are based on relative similarity to their corresponding gold-standard summaries. These gold-standard summaries are mostly written by humans. ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is one such approach, which is widely used as a standard measure. It accounts for the words that overlap between the gold-standard summary and the automatically produced summary, and also counts how much of the summary is overlapping with the standard one. These values are captured in two concepts: Precision and Recall. [10]

Recall = number of overlapping words/total words in reference summary

Precision = number of overlapping words/ total words in system summary

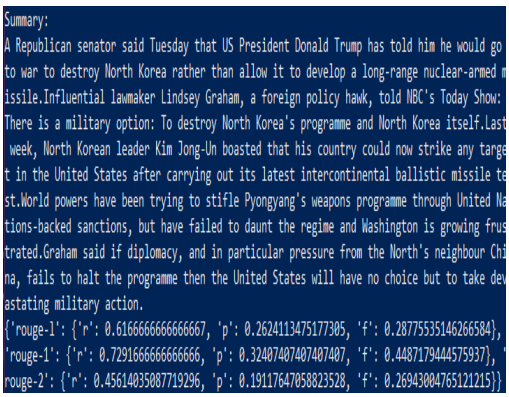
There are three major types of ROUGE measures: [11]

i. ROUGE-N: This is used to measure n-gram overlap between the gold-standard and the automated summary.

ii. ROUGE-L: This method measures similarity by comparing longest common subsequence between phrases/sentences.

iii. ROUGE-S: This method is also called as skip-gram co-occurrence. This is used to allow gaps between words, and hence comparing the temporarily formed phases.

Other evaluation measures include methods such as BLEU (Bilingual Evaluation Understudy) and SERA (Summarization Evaluation by Relevance Analysis).



The results have been represented in three different rouge measures, each denoting p,r,f values which stand for precision, recall, and f measure respectively.
The following are the average measures obtained over the summaries of various news articles:

A. Rouge-1
- r: 0.6022
- p: 0.3199
- f: 0.4929

B. Rouge-l
- r: 0.5302
- p: 0.2148
- f: 0.1973

C. Rouge-2
- r: 0.3031
- p: 0.1794
- f: 0.2361

10. Conclusion

It is evident that text summarization is a growing and a much-needed research area. Its uses are numerous – summarizing newspaper articles, medical reports, large documents, etc. The fact that a large amount of substantial information is needed in a shorter time in everyday life motivates this area to be developed further. Text summarization has further scope of integrating the above mentioned techniques to produce better results and perfect summaries. Along with text summarisation, Question-Answering is a very valuable part of Natural Language Processing with a great scope in the future.

In this paper, we presented an approach using a graph based model similar to TextRank to implement Extractive Text Summarisation. We also explained the working of a simple Question-Answering System implemented using various tools of Natural Language Processing.

With our algorithm, we have successfully achieved results by generating excellent summaries and almost accurate answers to the questions asked to the Question Answering system. In the future we would like to perform Text Summarisation using Unsupervised Deep Learning methods to improve the efficiency and accuracy of the summariser.

References

[1] Yogan Jaya Kumar, Ong Sing Goh, Halizah Basiron, Ngo Hea Choon and Puspalata C Suppiah, A Review on Automatic Text Summarization Approaches, Journal of Computer Science.

[2] Sumya Akter, Aysa Siddika As, Md. Palash Uddin, Md. Delowar Hossain, Shikhor Kumer Roy, and Masud Ibn Afjal, An Extractive Text Summarization Technique for Bengali Document(s) using K-means Clustering Algorithm, IEEE.

[3] Sukriti Verma, and Vagisha Nidhi, Extractive Summarization using Deep Learning.

[4] Deepali K. Gaikwad and C. Namrata Mahender, A Review Paper on Text Summarization, International Journal of Advanced Research in Computer and Communication Engineering, Vol. 5, Issue 3, March 2016.

[5] N. Moratanch, Dr. S. Chitrakala, A Survey on Extractive Text Summarization, IEEE International Conference on Computer, Communication, and Signal Processing (ICCCSP-2017).

[6] Nazreena Rahman, Bhogeswar Borah, A Survey on Existing Extractive Techniques for Query-Based Text Summarization, International Symposium on Advanced Computing and Communication (ISACC), 2015.

[7] N. Moratanch, Dr. S. Chitrakala, A Survey on Abstractive Text Summarization, International Conference on Circuit, Power and Computing Technologies [ICCPCT], 2016.

[8] Rada Mihalcea, Paul Tarau, TextRank: Bringing Order into Texts

[9] Sweta P. Lende, M. M. Raghuwanshi, Closed Domain Question Answering System Using NLP Techniques, International Journal of Engineering Sciences & Research Technology.

[10] Josef Steinberger and Karel Jezek , Evaluation Measures For Text Summarization, Computing and Informatics, Vol. 28, 2009, 1001–1026, V 2009-Mar-2.

[11] Arman Cohan and Nazli Goharian, Revisiting Summarization Evaluation for Scientific Articles.

[12] Padmapriya, V.Saminadan, "Performance Improvement in long term Evolution-advanced network using multiple imput multiple output technique", Journal of Advanced Research in Dynamical and Control Systems, Vol. 9, Sp-6, pp: 990-1010, 2017.

[13] S.V.Manikanthan and K.Baskaran "Low Cost VLSI Design Implementation of Sorting Network for ACSFD in Wireless Sensor Network", CiiT International Journal of Programmable Device Circuits and Systems,Print: ISSN 0974 – 973X & Online: ISSN 0974 – 9624, Issue : November 2011, PDCS112011008.