DU Journal of Undergraduate Research and Innovation

# A Graph Based Ranking Strategy for Automated Text Summarization

Nitin Agrawal,Shikhar Sharma, Prashant Sinha, Shobha Bagai
nitin.cic@gmail.com , shikharcic23@gmail.com, prashantsinha94@gmail.com, shobhabagai@gmail.com
Cluster Innovation Centre, University of Delhi, Delhi, Delhi 110007

ABSTRACT

Text summarization is a process of capturing the idea and line of thought from an original text and inculcating the same into a short coherent text. Automated text summarization aims to meet this objective of retaining all the key ideas instilled in the text while skipping upon the redundant and repetitive bits of information. The reduced text thus compiled must be coherent in itself in order to meet the semantic and syntactic organization of the language. This work presents an extraction based automatic text summarization algorithm. The methodology proposed involves constructing of a directed weighted graph out of the original text wherein each sentences is taken to be a node. The weights for each of the edges are determined by using a suitable distortion measure which analyses the semantic relation between the two adjacent nodes / sentences. A ranking algorithm is used to compute the most important sentences in the text and that should be present in the summary based on the weighted graph. This technique has been employed on multiple data sets and has performed well on the evaluation parameters laid down for such applications.

Keywords: Graph, text-rank, automated summarizer, distortion, extraction

## INTRODUCTION

Automatic text summarization (ATS) is a process that enables a computer to summarize data/ information automatically. With massive growth in information, summarization has become more important for enlisting significant parts of a big corpus. It provides a non-redundant bits of information from an original article. The amount of information available today is tremendous and the problem of finding the relevant pieces and making sense of these is becoming more and more essential. Nowadays, a great deal of information comes from the Internet in a textual form. Text Summarization helps in various kinds of analysis and forms a base for different Natural Language Processing Algorithms. ATS has a wide range of applications such as summarization of news articles, search engines presenting summarized results, language translation, email thread summarization etc. Text Summarization is broadly divided into two categories. The first category is text abstraction which involves parsing the text on semantic grounds followed by a formal representation. This is followed by re-interpretation of the text into a different non-redundant segment which in turn is the summarized version of the original text. The second category is text

extraction wherein the process involves identification of most important / relevant aspects of text using statistical information techniques. We use the text extraction based method to summarize documents because of its less computationally intensive nature, ease of scalability and availability of various techniques for analysis. This moreover invokes minimal distortion in the original text. The algorithm basically involves preprocessing the words on a corpus followed by graph based text ranking based on relevance. It is an unsupervised graph based ranking algorithm. It takes into account the keywords, frequency, relationship between sentences and the distortion measure. The graph is connected, undirected/directed and represents the text. Each sentence is represented by a vertex. Distortion measure which determines the distinctively of sentences is filtered against a threshold to decide if an edge exists and computes its corresponding weight. The approach is a graph based extractive summarization. The sentences are split on punctuation marks followed by removal of stop words. Distortion measure is used to form the graph as explained earlier. Distortion measure is based on Squared Error which is a statistical way of quantifying difference between values.

Finally the Text Ranking algorithm is used to check relevance and summarize the text.The arena of text summarization has been investigated by the NLP community for the last half century. (1) defines a summary as "a text that is produced from one or more texts, that conveys important information in the original text(s), and that is no longer than half of the original text(s) and usually significantly less than that". Earliest instances of research on summarizing scientific documents for extracting salient sentences from text elaborated usage of using features like word and phrase frequency (2), position in the text (3) and key phrases (4) .The work on text summarization (5) which proposed combination of sentence extraction and trainable classifier using Support Vector Machine presented a reasonable goof summarization however the readability was an issue. Another work on text summarization (6) presents a sentence reduction system for automatically removing extraneous phrases from sentences that are extracted from a document for summarization purpose.

One of the works (2) stressed upon the significance of frequency of word in a text. The words were stemmed to their root forms, and stop words were deleted. Furthermore, the frequency was used to sort the words in a decreasing order. On a sentence level, a significance factor was suggested that reflected the number of occurrences of significant words within a sentence, and the linear distance between them due to the intervention of non-significant words. All sentences are ranked in order of their significance factor. (3) Provides early insight on a particular feature helpful in finding salient parts of documents: the sentence position. Towards this goal, the author examined 200 paragraphs to find that in 85% of the paragraphs the topic sentence came as the first one and in 7% of the time it was the last sentence. Thus, position based features are also important for text summarization. In one of the studies (4) a typical structure for an extractive summarization was experimented upon. Graph-based ranking algorithms for sentence extraction have also been applied to text summarization. (7) Had proposed a modified page rank algorithm for text summarization. (8) Had proposed a distortion measure based summarization methodology. In the current work the usage word frequency and position features was motivated from these works. Two other features which been used are the presence of cue words and the skeleton of the document with reference to sentence being a

heading. Weights were attached to each of these features manually to score each sentence.

## METHODOLOGY

This section explains the methodology adopted is detail. An extraction based approach is being described herein. The methodology involves various steps the requirement of which is a basic preprocessing. After using basic Natural Language Techniques (NLP) for the preprocessing, a distortion measure based graph is formed and a ranking procedure is carried out. Further, sentences are selected on the basis of summarization factor. Detailed methodology is as described in current section. The precise steps in the methodology involve preprocessing, stemming, stop word removal and the processed data is then subjected to mathematical analysis for summarization.

The preprocessing steps performed in an automatic summarization task are word stemming, stop words removal, text segmentation and query expansion. These steps are also used in various other NLP tasks such as document retrieval, information extraction and machine translation.

Stemming is the process of reducing the inflected forms of a word to a root form. For example words like kicks, kicked, kicking all have the same root form "kick".This enables non-redundant representation of words with almost same semantic value. It also contributes in reduction of overall feature space. The stemming is a process which is specific to a language. One of the ways is to use a set of predefined language-specific rules to transform a word into its baseform. Porter stemmer is the most well-known algorithm of this kind for English language. It is based on suffix elimination logic. In certain cases it may be more rough than required eg. "organizational" is stemmed to "organ" .The viable solution is to combine the two described approaches by introducing the list of frequently used exceptions to a rule based stemmer

Stop words are high-frequency words of a language that don't carry any particular information on their own. Such words are removed at the preprocessing phase to reduce the number of features. Closed class words such as pronouns, articles, prepositions and conjunctions are often included in stop words lists.

The text that is taken as an input is filtered to remove any non-identifiable characters. All other characters have non-ASCII characters are considered insignificance for current usage. The input text is then split in to its constituent sentences. These sentences are stemmed and have their stop words removed. The processed sentences are now taken through suitable ranking algorithm that can signify each sentence's importance to the entire paragraph. The algorithm suggested in this paper is a modification of the famous Page Rank algorithm (9) which was used to find the importance of a web link in terms of its connectivity to other pages. Unlike other graph based ranking algorithms, this takes into account both the incoming and outgoing connections of a single node and then proceeds to give a set of scores governed by the Page Rank equation. This equation has been modified to suit the

needs of assigning scores to sentences rather than links. The scores allocated to each sentence is now governed by the weighted graph text rank equation (7).

$$TR(V_i) = 1 - d + d\ \sum_{\mathrm{Vj}\in \mathrm{I}} \frac{TR(Vj)}{Out(Vj)})$$

where $TR(V_i)$ denotes the text rank of the i[th] sentence and $Out(Vj)$ represents the out degree of the j[th] sentence, d is a parameter set between 0 and 1.This parameter known as the damping factor is used to balance between distinct and redundant sets of information. Generally, it is taken to be 0.85 for optimal performance wherein non-specific text summarization is being undertaken. The equation runs on a set of random weights assigned initially to each edge and iterates a number of times till convergence condition is met. The final values denote the ranks of each of the nodes. The algorithm proposed in this current work takes the distortion measure between each of sentence as the weights for the edges. The distortion measure signifies the semantic difference between any two sentences. Every two sentences are examined by a distortion measure representing the semantic relation between them. This then represents the weight of the edge between those two nodes. The distortion measure (8) used in this model is based on the 'Squared Error' which statistically quantifies the difference between two sentences. This is done by squaring and adding the frequency of the words that are not common between the sentences. In case a word is common between the two sentences, its frequency is calculated in the second sentence and this is subtracted from the score of that word. The frequency is squared and added to a sum. The second sentence is then checked for its not common words with the first sentence. If the word is not common, the score is squared and added to the sum, and the number of not common words is incremented by 1. The final distortion measure is then calculated using the equation

$$\text{Distortion} = \frac{\text{Sum}}{\text{non common words}}$$

(Figure I) represents the distortion measure based graph for a simple data set of 6 sentences.

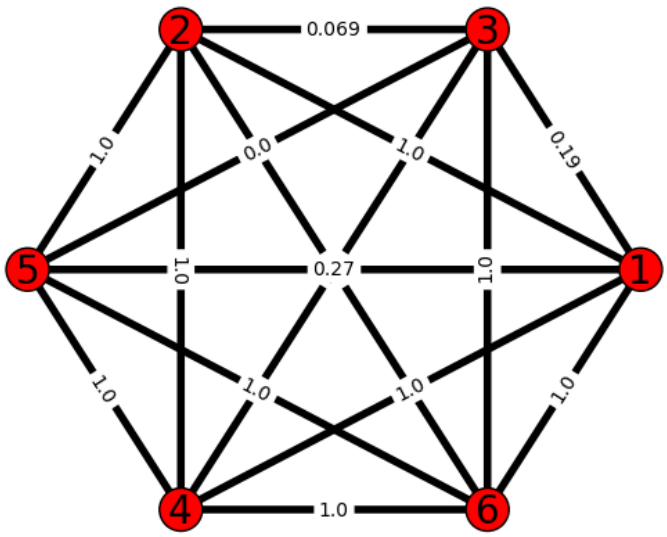



Figure I: An illustrative distortion graph for six sentences

The adjacency matrix of this distortion matrix graph is then passed to the text rank equation which computes the rank of each sentence. Once the ranks have been assigned, the highest ranked sentences taken in order form the summary. The number of sentences used in the summary depends on a summarization factor determines the number of sentences to be chosen from the ranked list of sentences.

*Pseudocode*

```
OPEN and READ file
filterASCII(text)                                   //Filter non ASCII characters
sentences = regex.split('(?<!\w\.\w.)(?<![A-Z][a-z]\.)(?<=\.|\?)\s', text)
FOR j in 0,length(sentences)                        //Split sentences
APPEND ([i for i in sentences[j].split() if i not in stop])
punctuationRemove(sentences)
PorterStemmer(sentences)                            //Apply Stemmer
ComputeFrequency(word in sentences)
fori in range(0,len(sentences)):                    //Compute Adjacency
  for j in range(i+1,len(sentences)):
     adjacency[i,j] = adjacency[j,i] = find_distortion(i,j)
adjacency=1-adjacency/float(adjacency.max())
fori in range(0,10):                                // 10 iterations for convergence
  text_rank = find_rank(text_rank,adjacency,0.85)
print sentences
```

Table I: A pseudocode depicting the proposed algorithm

The methodology was implemented in python with the help of NLTK module. The following text precisely explains the major algorithmic component's as outlined in (Table-I) involvingtext reading & preprocessing, frequency computation, matrix formulation & normalization and sentence rank computation.

**Reading and Pre-Processing**

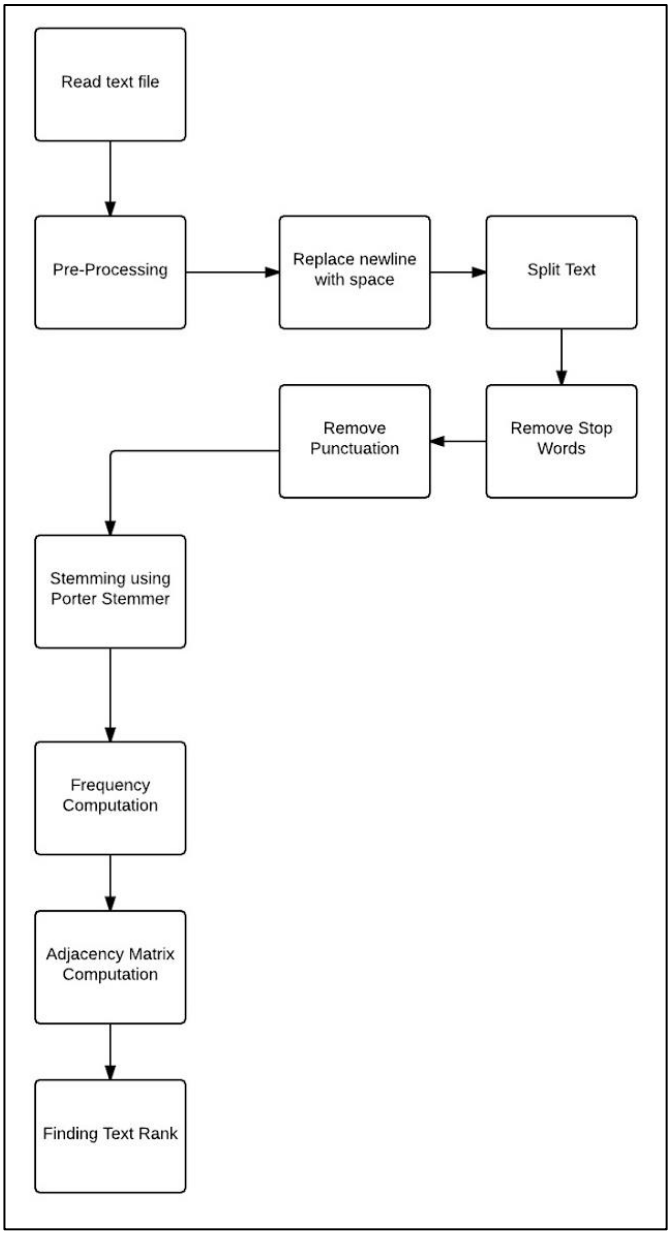
The file is opened, read and natural language processing based preprocessing is done as follows

1. Replacement of new line character by spaces
2. Splitting on symbols
3. Removal of stop words
4. Punctuation Removal
5. Stemming using Porter Stemmer

**Frequency Computation**
A dictionary based on key-value pair is formed with word as the key and corresponding frequency as its value. This frequency is used in later part of the algorithm for rank computation purposes.

**Adjacency Matrix and Normalization**
The distortion measure is a marker of dissimilarity between pairs of sentences. An adjacency matrix is formed with nodes being the sentence labels and edges representing a similarity parameter which is obtained from the distortion measure. For obtaining the similarity measure form the distortion measure, it is normalized in domain [0,1] and a complementary value is then obtained. A value of one signifies perfect similarity and a value of zero represents perfect dissimilarity. A threshold is decided upon in order to skip upon the edges with similarity values less than that of the threshold.

**Rank Computation using the weighted graph**
The text ranking algorithm is applied on the graph (vertices, distortion measure) to rank the sentences. The ranking is obtained and are sorted accordingly to get the summarized version of the text in coherence with the desired summarization factor.




FigureII: A flowchart representing the proposed approach

## RESULTS

Two data sets have been used to portray experimental results for the algorithmic approach proposed herein. Test set I consisted of 48 sentences while test set II consisted of 32 sentences. Both have been summarized at different summarization factors. The summarization factor determines the ratio of the original text to the summarized text. A standard damping factor of 0.85 was taken throughout the tests. (Table-II) illustrates a sample summary formulated by our proposed summarization strategy for one of the test sets.

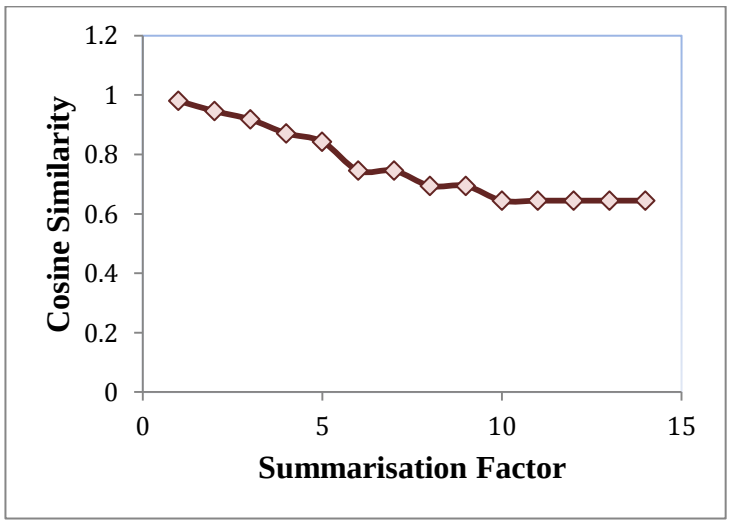
Sampling is the process by which inference is made to the whole by examining only a part of the population. Sampling is inevitable if the population selected is infinite and when the results are required in a short time. Sampling also becomes necessary when the area of survey is wide and the resources available are limited in terms of money and person (Web). During the information collection for the research it was mandatory to target consultants who have faced the situation which the dissertation tries to cover. According to Bunting (2005) the number of possible responses an interviewer can expect from an open ended question is more compared to any other technique. This gives an opportunity to the interviewer to choose an appropriate response based on the responses.

**Table II: An illustrative summarization for Data set II at a summarization factor of 4**

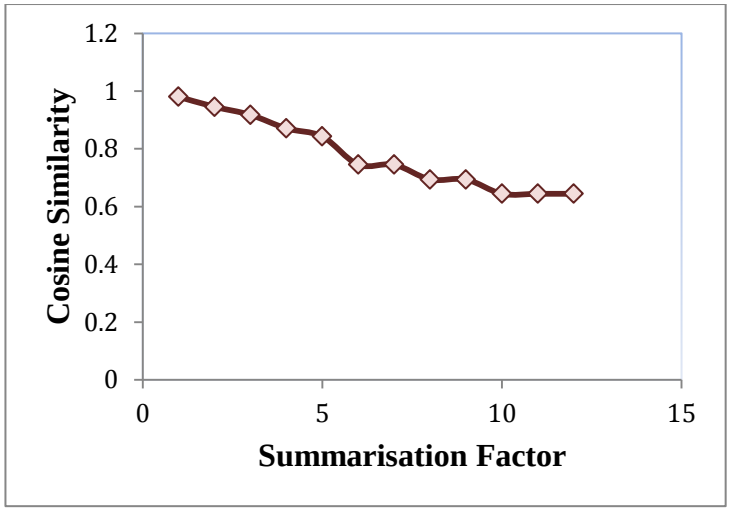
## DISCUSSION

In order to evaluate the coherency and agreement between the original text and the summarized text, cosine similarity was used as an evaluation measure.Cosine similarity measure is technique used to validate a text summarizer. It is based on similarity between two document vectors (10). The cosine between two document vectors is maximum that is it has a value of 1 when the angle between those vectors is null. Hence, a higher cosine similarity between documents signifies a positive relation (11).

The cosine similarity is calculated between the original and summarized document. It decreases as the summarization factor is increased. This measure inculcates a similarity approximately 0.9 when the summarization factor is kept at 2 for most data sets. As can be observed from (Figure III)&(Figure IV) cosine similarity doesn't fall rapidly and the fall is rather linear as we increase the summarization rate indicating high positive agreement of the summary to the original text. The agreement validates the credibility and effectivity of the approach proposed by us.




FigureIII : Text to summary agreement for Test set I




FigureIV : Text to summary agreement for Test set II

## CONCLUSION

The summarizer is tested on different text sets with different summarization factors. A fairly accurate summary was obtained in most of the cases. The summaries contained most important information sets that were essential to the original text for the summaries with practicable summarization factor. A high cosine similarity and reasonable fall on increasing the summarizationfactor has portrayed positive agreement between the original text and the summary. The summarized text sets have moreover been observed to be highly coherent in terms of quality.

## ACKNOWLEDGMENTS

We would like to express our gratitude to the Director, Cluster Innovation Centre Prof M.M. Chaturvedi and other faculty members at Cluster Innovation Centre for their support.

.

## REFERENCES

1. Radev, D. R., Hovy, E., &McKeown, K. (2002). Introduction to the special issue on summarization. *Computational linguistics*, *28*(4), 399-408.
2. Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of research and development*, *2*(2), 159-165.
3. Baxendale, P. B. (1958). Machine-made index for technical literature: an experiment. *IBM Journal of Research and Development*, *2*(4), 354-361.
4. Edmundson, H. P. (1969). New methods in automatic extracting. *Journal of the ACM (JACM)*, *16*(2), 264-285.
5. Ishikawa, K., Ando, S. I., Doi, S. I., & Okumura, A. (2002). Trainable automatic text summarization using segmentation of sentence. In *In Proc. 2002 NTCIR 3 TSC workshop*.

6. Jing, H. (2000, April). Sentence reduction for automatic text summarization. In*Proceedings of the sixth conference on Applied natural language processing*(pp. 310-315). Association for Computational Linguistics.
7. Mihalcea, R. (2004, July). Graph-based ranking algorithms for sentence extraction, applied to text summarization. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions* (p. 20). Association for Computational Linguistics.'
8. Samei, B., Eshtiagh, M., Keshtkar, F., &Hashemi, S. (2014, March). Multi-Document Summarization Using Graph-Based Iterative Ranking Algorithms and Information Theoretical Distortion Measures. In *The Twenty-Seventh International Flairs Conference*.
9. Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. *Computer networks and ISDN systems*, *30*(1), 107-117.
10. Xie, S., & Liu, Y. (2008, March). Using corpus and knowledge-based similarity measure in maximum marginal relevance for meeting summarization. In*Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on* (pp. 4985-4988). IEEE.
11. Donaway, R. L., Drummey, K. W., & Mather, L. A. (2000, April). A comparison of rankings produced by summarization evaluation measures. In *Proceedings of the 2000 NAACL-ANLPWorkshop on Automatic summarization-Volume 4*(pp. 69-78). Association for Computational Linguistics.