A Major Project-II Report

On

Automatic Text Summarization using Soft-Cosine Similarity and Centrality Measures

Submitted in fulfilment of the Requirement for the Degree of

Master of Technology

in

Computer Science and Engineering

Submitted By

Harshita Rastogi

2K18/CSE/07

Under the Guidance of

Ms. Minni Jain

Assistant Professor

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahabad Daulatpur, Main Bawana Road, Delhi-110042

August 2020

CERTIFICATE

This is to certify that Project Report entitled **"Automatic text summarization using soft-cosine similarity and centrality measures"** submitted by **Harshita Rastogi** (2K18/CSE/07) in fulfilment of the requirement for the award of degree Master of Technology (Computer Science and Engineering) is a record of the original work carried out by her under my supervision.

Minni Jain

Project Guide
Ms. Minni Jain
Assistant Professor
Department of Computer Science & Engineering
Delhi Technological University

DECLARATION

I hereby declare that the Major Project-II work **"Automatic text summarization using soft-cosine similarity and centrality measures"** which is being submitted to Delhi Technological University, in partial fulfilment of requirements for the award of the degree of Master of Technology (Computer Science and Engineering) is a bona fide report of Major Project-II carried out by me. I have not submitted the matter embodied in this dissertation for the award of any other degree or diploma.

Harshita Rastogi

Harshita Rastogi
2K18/CSE/07
M. Tech (Computer Science & Engineering)
Delhi Technological University

ACKNOWLEDGEMENT

First of all, I would like to express my deep sense of respect and gratitude to my project supervisor Ms. Minni Jain for providing the opportunity of carrying out this project and being the guiding force behind this work. I am deeply indebted to him for the support, advice and encouragement he provided without which the project could not have been a success.

Secondly, I am grateful to Dr. Rajni Jindal, HOD, Computer Science & Engineering Department, DTU for her immense support. I would also like to acknowledge Delhi Technological University library and staff for providing the right academic resources and environment for this work to be carried out.

Last but not the least I would like to express sincere gratitude to my parents and friends for constantly encouraging me during the completion of work.

Harshita Rastogi

Roll No – 2K18/CSE/07

M. Tech (Computer Science & Engineering)

Delhi Technological University

ABSTRACT

Automatic text summarization is one of the major problems in the field of machine learning. The approach used in this project is significantly different from all previous works done in this field in the respect that it uses two major concept called soft-cosine similarity and centrality measures. Soft cosine similarity takes into account the semantic relationship between the words thereby reducing the ambiguity caused by words with similar meanings. It also helps to realize how similar two sentences are which could be used to reduce redundancy and hence improve the quality of the final summary produced. There are dictionaries present to get the semantic relations between words. We are using WordNet which is an English linguistic dictionary containing 8 different relation types.

Centrality measures is another widely used concept for graph-based approaches. We have discussed 40 different centrality measures and analyzed there impact and usage in 30 different real world networks. Finally studied 4 basic and most widely used centrality measures in order to decide which measure derives best results. EigenVector has shown to outperform other centrality measures.

We have used two types of datasets single text documents from BBC news articles and and multi-text documents from DUC 2007 dataset. We have used the renowned ROUGE measure to compare the results and found that our approach performs better than all other state-of-the-art automatic text summarization methods namely TextRank, LexRank, Luhn and LSA.

CONTENTS

Page No.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

S. No	Abbreviation	
1	NLP	Natural Language Processing
2	DTM	Document Term Matrix
3	VSM	Vector Space Model
4	LSA	Latent Semantic Analysis
5	RNN	Recurrent Neural Network
6	DUC	Document Understanding Conference
7	BBC	British Broadcasting Corporation
8	ROUGE	Recall-Oriented Understudy for Gisting Evaluation
9	FBI	Federal Bureau of Investigation

CHAPTER 1: Introduction

Text summarization refers to the creation of a compact, fluent and coherent version of one or more text documents. The primary intention is to provide an outline of the information in the original documents quickly and efficiently using fewer words. The internet is the biggest source of information today, it is flooded with huge amount of textual data in the form of web pages, blogs, news articles, emails, tweets etc., causing information overload. Moreover with no proper authorization, it is hard to determine the authenticity of the information present. A summary can not only provide an overview of the source but also its genuineness before being studied in detail. It is expensive, subjective and infeasible to manually extract the summary from such large amount of documents. Hence there is a need for automatic text summarization technique to process the data. In spite of the progress over the years, there is still scope of better results. The main challenge is how to determine the most crucial information and how to express this information efficiently in the final summary without exceeding the size limit.

Automatic text summarization could be classified into four types based on the type of input, output, context and external resources:

a) *Input-based text summarization* is categorized as single-document text summarization methods such as TextRank, LexRank both based on PageRank algorithm, SumGraph etc and multi-document text summarization methods like LexPageRank, feature-based summarization, statistical extractive summarization etc.

b) *Output-based text summarization* is of two types- Extractive text summarization creates summary from the original document choosing the most important phrases, sentences or paragraphs without any modification while Abstractive text summarization is more human-like i.e. a conceptual conclusion of the given document with some or all new words, phrases or paragraphs. Extractive summaries are easier and faster to generate as they can be generated entirely from the original document with no further knowledge of any kind. Abstractive summaries on the other hand need a good domain and linguistic knowledge so as to understand the essence of the document and present it in its own words.

c) *Context-based text summarization* can be of three types- Domain-specific text summarization which uses domain knowledge related to the document it is dealing with so as to identify important and unimportant words and phrases. Querybased text summarization which forms the final summary using the answers to a set of pre-decided natural language questions. Generic text summarization is the most basic and the most explored type of summarization where there is no prior assumption or context taken into account .

d) *External resources-based text summarization* can be knowledge-poor in which case no external source of ontology is used. While in knowledge-rich text summarization a sizeable knowledge base is used. It is domain-independent and thus applicable across domains and applications . Fig 1. shows the classification of automatic text summarization and the techniques used in our paper are highlighted in green boxes i.e. our approach provides a multi-document, extractive, generic and knowledge-rich type of automatic text summarization technique.

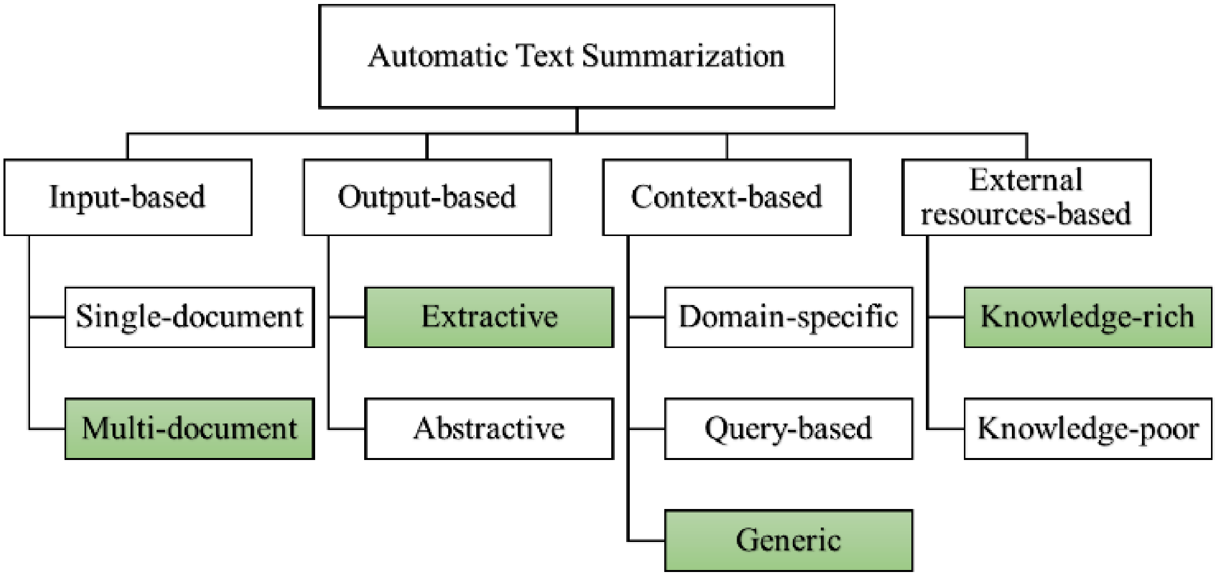


Fig. 1. Classification of automatic text summarization

Previously, a lot of generic text summarization methods have used cosine similarity vector space model in order to calculate relatedness between the sentences directly or in a modified form, as the tf-idf modified cosine similarity . The concept of centrality is useful to find important nodes in a graph and has been readily used in social, biological, electrical, technological and information networks . There are over 200 types of centrality measures for different types of networks, structures and requirements . We are using the four most widely used centrality measures namely degree, betweenness, closeness and eigenvector centrality . Since former three measures do not originally deal with weighted networks, we are using their weighted versions .

1.1 Related work

The main objective of behind the emergence of the concept of automatic text summarization has been the increasing rate of data which was impractical and infeasible to be processed by humans with desired time and efficiency. Thus various works have been presented in the past to overcome this problem as listed below:

a) ***Positional method:*** A method introduced in 1959 in which while analyzing 200 paragraphs from scientific documents. The author came across that the fact that topic sentence is generally the first or the last sentence of the document.

b) ***Luhn's method:*** Luhn discovered that the most and the least frequent words are generally the least important words in the document. He also introduced the concept of data pre-processing in order to filter out such words and the concept of stemming used to remove suffixes from words e.g. cats → cat.

c) ***Edmundson's method:*** Edmundson used a linear combination of features namely position, frequency, cue of words and document structure. Cue words are the manually chosen words correlated to the sentences. Cue words are of three types- bonus words (refer important sentences), stigma words (negatively impact sentence importance) and null words (have no impact on sentences).

d) ***FRUMP:*** The first knowledge based text summarization was named as the Fast Reading Understanding and Memory program (FRUMP) . In this method the relevant sentences are selected by filling the pre-determined template. Semantic and pragmatic knowledge are used for summarization. It also uses a data structure called sketchy scripts to predict events.

e) ***Maximal marginal relevance:*** A query based summarization approach which takes in user query and using a predefined similarity matrix. It chooses sentences to be included in the summary based on the fact that how similar is the sentence to the query as well as the set of sentences already added to the summary. It is a diversity-based method used for reranking documents and generating summaries.

f) ***Classification:*** The first trainable method designed with a training set of original documents and manually created extracts used for the classification model to predict the probability of a sentence to be selected for the summary. This probability was determined usin the Naïve-Bayes classification assuming that the features are statistically independent.

g) ***LexRank****:* In , The LexRank is inspired by Google's PageRank algorithm which is used to rank the webpages. It is a graph-based approach and uses the concept of Lexical centrality. A similarity matric is formed using sentences as vertices and similarity score as edge weights between them. It was mainly designed for single document summarization but could be extended to multi-document as well without much alterations.

h) ***TextRank****:* In , another approach similar to LexRank is introduced known as TextRank, another graph-based algorithm using the PageRank approach. The nodes can be words, phrases or entire sentences and the relation between them can be contextual overlap, semantic relations or lexical relations etc. The graph generated can be weighted or unweighted and directed or undirected. If the nodes are considered to be words or phrases, it can also be used for keyword and keyphrase extraction.

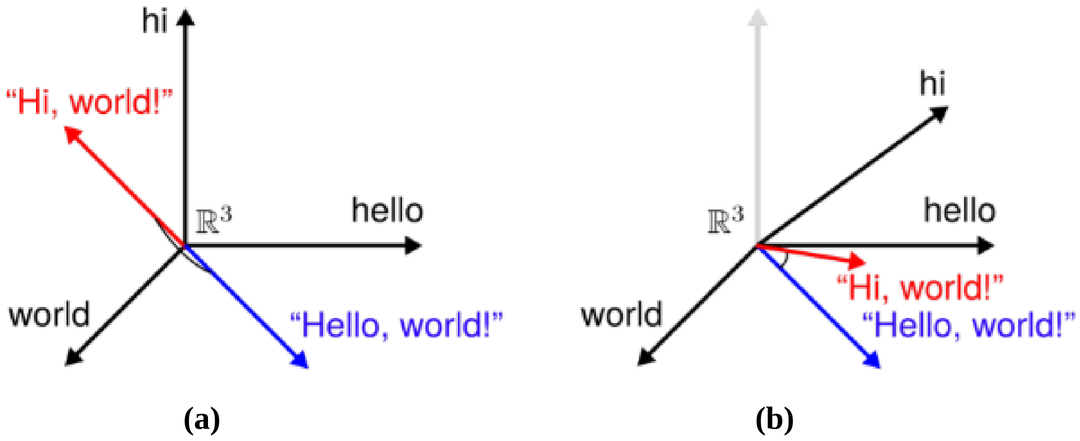
i) ***Normalized Google Distance****:* In , the authors use clustering mechanism to divide sentences and then apply Normalized Google Distance as the similarity measures on each cluster to determine the relatedness between the sentences in each cluster. The Differential Evaluation method which works on real valued parameters is used here but is modified using the mutation from the genetic algorithm. The cluster containing the most similar information as the document is chosen to form the summary.

j) ***MEAD****:* In , the open source toolkit for a centroid-based approach called MEAD was presented which works for single as well as multi-documents. Its publicly available and can be extended to apply other features. It has three major components: feature extractor, combiner and reranker. MEAD distribution also has three basic features- Centroid, Position and Length. The MEAD policy works on the combination of command lines used for the features, formula used to convert vector to scalar and command line used for reranking.

- ***k)*** ***Sequence to sequence:*** A deep learning technique introduced by Google which takes sequence of words as an input and generates sequence of words as an output. The words from the original document are sent to encoder which is a stack of several recurrent neural networks (RNN) . Each RNN accepts one input element i.e. one word and propagates it to the next recurrent unit after collecting information from them. The output produced by all these RNN is an encoder vector which is then fed to into another stack of RNN called the decoder. It uses the collective information present in the encoder vector to make predictions and produce final summary,

CHAPTER 2: Soft-Cosine Similarity

The soft-cosine similarity measure also known as the soft similarity measure is a variation of the well-known cosine similarity measure. It is different from the cosine similarity measure in the respect that it takes into consideration the various relationships words possess such as synonyms, antonyms, hyponyms, meronyms etc. Based on the similar concept of vector space model as used the regular cosine similarity matrix where the features are treated as independent entities, it takes into account the fact that even though the sentences may not possess same words but may hold similar meaning due to the interdependence between the words.
This difference between the two Vector Space Models (VSM) is demonstrated in Fig 3.



(a) **(b)**

Fig. 2. Comparison of vectors' representation in a) Cosine similarity VSM b) Soft-cosine similarity VSM in 3-dimensional space.

The Vector space model presents the objects as n-dimensional vectors . The soft-cosine similarity between two sentence vectors *a* and *b* can be calculated as:

$$Cosine_{soft}(a,b) = \frac{\sum_{i.j}^{n} S_{ij}a_i b_j}{\sqrt{\sum_{i.j}^{n} S_{ij}a_i a_j}\sqrt{\sum_{i.j}^{n} S_{ij}b_i b_j}}$$

where S_{ij} is the semantic relation between features *i* and *j*. But if there is no similarity between the features $S_{ii} = 1$ and $S_{ij} = 0$ for $i \neq j$ which is equal to cosine similarity:

$$Cosine(a,b) = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2}\sqrt{\sum_{i=1}^{n} b_i^2}}$$

The quadratic time complexity of the similarity measure makes it feasible to be used in real-world scenarios. Noteworthily the complexity could be further improved to sub-quadratic.

CHAPTER 3: Centrality Measures

Centrality can be interpreted as- ‘influence’, ‘prestige’ or ‘control’. With more than 200 different types of centralities proposed so far for different types of networks and based on different kinds of roles the nodes play in them. The centrality of a node in a network may be determined by the three most basic structural attributes of that node: its degree, betweenness, or closeness. The choice of the structural attribute and its measure depends upon the context to which its application is intended. Degree based measure can be used for communication activity. Measure based on betweenness can be used for interest in control of communication. And for either independence or efficiency we can choose measure based upon closeness.

3.1 Types of centrality measures

1) **Degree Centrality:** One of the most basic approach to find a central node. Applicable in almost all the areas where we can generalize the problem in the form of a network and seek to determine important nodes based on their degree. In this measure importance of nodes in a graph depend on its number of links with other nodes i.e. direct friendship connections. It is be formulated as:

$$\sigma_D(x) = \sum_{i=1}^{n} a_{ix}$$

where a_{ij} is adjacency matrix and $\sigma_D(x)$ is centrality score.

2) **Betweenness Centrality:** One of the most basic approach to find a central node. Applicable in almost all the areas where we can generalize the problem in the form of a network and seek to determine important nodes based on path lengthCommunication takes place only along the geodesic paths. Here importance of nodes is based on the its relevance for the control and mediation for the information flow. It is formulated as:

$$\sigma_B(x) = \sum_{i=1, i\neq x}^{n} \sum_{j=1, j<i, j\neq x}^{n} \frac{g_{ij}(x)}{g_{ij}}$$

where g_{ij} is shortest path between i and j and $g_{ij}(x)$ is paths passing through x node.

3) **Closeness Centrality:** One of the most basic approach to find a central node. Applicable in almost all the areas where we can generalize the problem in the form of a network and seek to determine important nodes based on topology of the network. This measure uses minimum geodesic distance i.e. minimum number of edges that need to traversed to reach from one node to another. It is formulated as:

$$\sigma_C(x) = \frac{1}{\sum_{i=1}^{n} d_G(x, i)}$$

where $d_G(x,i)$: distance between the two nodes.

4) Eigen-vector Centrality: Power measure for a variety of exchange measures. Importance of a node depends on how important are the nodes connected to it.

$$\sigma_E = \frac{1}{\lambda_1}\sum_j A_{ji} v_j$$

where ƛ1is maximum eigen value for A, A is adjacency matrix and v is maximum eigenvector for A.

5) Page Rank: Ranking webpages by simulating user behavior where most linked and visited webpage is ranked highest. It is determined using the citation graph of hyperlinks in webpages the importance of webpages is derived. A page possesses high page rank if a no. of pages point to it or is pointed by some high page ranked pages. It is formulated as:

$$PR = (1 - d) + \left(d(\frac{PR(T_1)}{C(T_1)} + \cdots + \frac{PR(T_n)}{C(T_n)} \right)$$

where T_1...T_n are pages, C(A) is no. of links leaving page A and d is damping factor.

6) HITS: Ranking websites based on user importance of mutual reinforcement. A hub is a positive version of authority. It uses an iterative relation between authorities- pointed by many hubs and hubs- pointing to authorities to determine good-quality information in broad-level search. It is formulated as:

$$h_{i+1} = a_i A^T,$$
$$a_{i+1} = h_{i+1} A$$

where h is hub and a is authority.

7) Katz Centrality: Used in wide variety of directed networks like web network, citation network and biological networks. It's a generalized form of degree centrality. It is determined by the no. of nodes accessible via a certain path. It is formulated as:

$$C_K^W(i) = \alpha \sum_j W\,(i,j).\, C_K^W(j) + \beta$$

where W is adjacency matrix, α is attenuation constant and β is positive constant.

8) Radiality Centrality: Influential in studying network reachability and connectedness. It is determined by the extent to which a node can access a network as provided by its neighbor nodes. High radiality means lesser time to reach other nodes in the network thus faster sending of information. It is formulated as:

$$C_{Rad}^{W}(i) = \frac{\sum_{j \in V}(\Delta_G^W + 1 - d^w(i,j))}{g_{jk}^{w}}$$

where Δ_G^W:diameter of the graph G, $d^w(i,j)$ is weighted geodesic distance between nodes i and node j.

9) Integration Centrality: Influential in studying network reachability and connectedness. How well a node is connected in the network. Being closely linked integrated nodes get early access of information. It is formulated as:

$$I(k) = \frac{\sum_{j \neq k} R\, D_{jk}}{N - 1}$$

where RD_{jk} is reverse distance calculated using geosedic distance between nodes j and k and N is network size.

10) Time-scale Degree Centrality: Consideration of presence as well as duration of nodes in network. Time-variant approach of degree centrality. It is formulated as:

$$tsdc_0 = \sum_j x_{ij}(T - t_i) \ldots$$
$$tsdc_1 = \sum_j x_{ji}(T - t_i) \ldots$$

where i is input and o is output.

11) Edge-betweenness: Centrality Topology control based on Quality of service for sensor networks to achieve high quality services. Evaluating relation between edges of network using this weighted, bidirectional topology-control algorithm. It is formulated as:

$$EB(e) = \sum_{v_i \in V} \sum_{v_j \in V} \frac{\sigma v_i v_{j(e)}}{\sigma_{v_i v_j}}$$

$\sigma_{(vi\ vj\ )}$= no. of shortest paths between nodes v_i and v_j.

12) Mint Centrality: Coin based transactions in bitcoin transaction graph. Spot important pseudonymous addresses in bitcoin transaction graph with linear time complexity. It is formulated as:

$$MC(A, h) = \frac{\left|\cup_U S_{u_i}\right|}{h}$$

where U is set of all transactions, h is height and S is set of coin-base.

13) Feature Centrality: Areas requiring prediction of human behavior towards an object features. Rational model for behavior of human categorization. First optimize the system then making some assumptions regarding the environment and costs optimal behavioral functions are derived. It is formulated as:

$$C_i^F = \frac{\sum_{j,k \in G} m_{jk}(i)}{\sum_{j,k \in G} m_{jk}}$$

where m_{jk} is maximum flow between j and k and $m_{jk}(i)$ is maximum flow through i.

14) Information/Delta Centrality: Useful in finding community structures, characterizing planar graphs and importance of soluble mediators in human immune system. Importance of a node is determined by the response of the network after the node is deactivated. Based on network efficiency and network flow. It is formulated as:

$$C_i^\Delta = \frac{(\Delta P)_i}{P}$$

where P is cohesiveness measure and $(\Delta P)_i$ is variation of P after deactivation of i.

15) Flow Betweenness Centrality: Betweenness of nodes with continuous flow of maximal amount of information between all source and destination nodes. Based on maximum flow in network. It is formulated as:

$$C_i^F = \frac{\sum_{j,k \in G} m_{jk}(i)}{\sum_{j,k \in G} m_{jk}}$$

where m_{jk} is maximum flow between j and k and $m_{jk}(i)$: maximum flow through i.

16) Current Flow Betweenness Centrality: Estimation of pedestrian flow in large urban networks. Estimation of status of traffic flow in spatial networks. Considering the network topology and data associated with it. Resized Approximation CFB version is used to provide more efficient and fast results in large network. It is formulated as:

$$b_i = \frac{1}{N_b} \sum_{s<t} I_i^{(st)}$$

where s, and t are current routes, N_b is normalization constant and $I_i^{(st)}$ is average current flow.

17) Second-order Centrality: Provide signature of graphs to determine their health i.e. how robust a graph is against attacks. Find critic nodes in a complex network. Regularity of visit through a random walk of a node determines its importance. Standard deviation of return time is used to determine critical nodes in the network. Distributed by design. It is formulated as:

$$\tau(j) = \inf\{n \geq 1 | X_n = j\}$$

where n is the number of nodes, X_n is the Markov chain on finite space S.

18) Alpha Centrality: Positive and negative bargaining systems. Total number of paths from a node to other nodes, exponentially attenuated by the length between them. Power comes from being connected to powerless. It is formulated as:

$$\lambda e_i = \sum\nolimits_j R_{ij} e_j$$

where R is the relationship matrix and e is the eigen vector.

19) Temporal Centrality: Analyze the vulnerability, time-varying centrality and traffic management in highly vulnerable and time-critical networks. Time varying Graph is made to study temporal features of network. Then using the modified version Matrix multiplication algorithm temporal centrality metric is obtained to analyze network vulnerability and improve traffic management system. It is formulated as:

$$TC_t(k) = \sum_{u \neq v \neq k \in V} \frac{|d'_t(u,v,k)|}{|d_t(u,v)|}$$

where $|d_t(u,v)|$ is no. of shortest path between u and v and $|d'_t(u,v,k)|$= among them paths passing through k.

20) Subgraph Centrality: Measure network bipartivity, an important topological characteristic of complex networks. Weighted sum of spectral moments of adjacency matrix. It is formulated as:

$$SC(G) = \frac{1}{N} \sum_{l=1}^{\infty} \frac{\mu_l}{l!}$$

where $\mu_l \sum_{j=1}^{N} (\lambda_j)^l$, μ_l is no. of closed paths of l length and λ is eigen values.

21) Dominating Centrality Set: Determine influential set of center nodes in large scale graphs without overlapping. Efficient network coverage to select central nodes. Using 3- and 4- cycles containing specific node. Is is formulated as:

$$R_G(S) = |N[S]|/|V(G)|$$

where $R_G(S)$ is covering rate of DCS on set of S nodes in graph G.

22) Cumulative Neighboring Relationship: Find central nodes effectively in rapidly changing Mobile Social Networks. First find pair-wise relationship between neighbors mobile social networks. Then break the dynamic network into a set of time-ordered networks. Later using 3 aggregation methods they are combined to form CNR metric and find central nodes in the given time interval. It is formulated as:

$$CNR(i) = \frac{1}{N-1} \sum_{j \in V, j \neq i} NR(ij)$$

where $NR_{(ij)}$ is the neighbor relationship.

23) Straightness Centrality: Spatial analysis of urban street patters and as a part of Multiple centrality assessment for street centrality measurement and efficient information transfer in network. Efficiency of communication between two nodes in a graph is inversely proportional to the shortest path. Captures the deviation of connecting route from virtual straight route between two nodes. It is formulated as:

$$C_i^S = \frac{1}{N-1} \sum_{j \in G, j \neq i} d_{ij}^{Eucl} / d_{ij}$$

where d_{ij} is shortest path length and d_{ij}^{Eucl} is Euclidean distance.

24) Ranking Betweenness: Mainly designed for complex urban street networks. Combination of random-walk betweenness and Page Rank centralities. After converting the network into geographical space node importance is found by considering manually decided set of features based on the problem along with network connectivity. It is formulated as:

$$C_{rb}(i) = \frac{\sum_{s<t} I_i^{st}}{\frac{1}{2} n(n-1)}$$

where $I_i^{st} = \frac{1}{2} \sum_j K_{ij} |T_{is} - T_{it} - T_{js} + T_{jt}|, i \neq s, t$ and K,T are metrices.

25) Clustering Coefficient : Check if a graph is small world network. Measures how cliquish is the neighborhood of a node Ratio of existing edges between two nodes within their neighborhood to the total number of possible edges. It is formulated as:

C(G)= 6n/l

where n: is no. of triangles and l is no. of paths of length two.

26) Local Bridging Centrality: A bridging node is a node connecting two densely connected parts of graph. It's a product of a global measure i.e. betweenness centrality that tells among all the shortest paths how many of them pass through the node and a local measure i.e. bridging coefficient telling how well the node is located between the nodes with high degrees. It is formulated as:

$$C_R(v) = BC(v) \times C_B(v),$$
$$BC(v) = \frac{d(v)^{-1}}{\sum_{i \in N(v)} \frac{1}{d(i)}}$$

where BC(v) is bridging coefficient, d(v) is degree of v node and N(v) is set of node v's neighbors.

27) Community Centrality: Optimization of modularity in technological, social and information networks. Some vertices ,as a result of the situation they are in, have positive or negative power to substantially contribute to the overall modularity of the network. It is formulated as:

$$|X_k| = \sum_{i \in G_k} \hat{X}_k^T x_i$$

where X_k is community vector and x_i is vertex vector.

28) Relative Centrality: Local community detection problems Importance of a set of vertices in a network with respect to another set of vertices instead of the set of all nodes of the network i.e. the whole network. It is formulated as:

$$C(S_1|S_2) = P(W \in S_1|V \in S_2$$

where S1 and S2 are set of nodes, and W and V are randomly selected nodes.

29) Hubble-index: The scientific achievement of a scholar is determined by the no. of papers and citation times of those papers measured using h-index. Used for scientometric purposes. Importance of a node is based on how many other high degree nodes it is connected to. It is equal to the largest no. h for a node n such that n has atleast h neighbors with degree not more than h. It is formulated as:

$$H_i = max_{1\leq h\leq d_l} \min\left(\left|\begin{matrix}\mathcal{N}_{\geq h}\\(i)\end{matrix}\right|\right), h)$$

where $\mathcal{N}_{\geq h}(i)$ is neighbor of the node I with at least h degree.

30) Lobby-index: An h-index inspired measure used for general networks. It is equal to the largest no. k for a node n such that n has at least k neighbors with degree not more than k. Many high degree neighbors means more influence. Used mainly in scale-free artificial networks. It is formulated as:

$$l(x) = \max\{k: \deg(y_k) \geq k$$

where deg(y_k) is degree of neighbor y with at least k degrees.

31) Communication Centrality: Inspired by h-index, takes into consideration degree, communication ability and edge weight of node. It is equal to the product of the edge weight and the h-degree of neighbor node. Used in scale-free weighted networks to study communication ability of nodes with their neighbors. It is formulated as:

$$c(x) = \max\{k: w_k d_h(k) \geq k\}$$

where k is minimum no. of neighbor nodes of x, w_h is h-degree of neighbor node k.

32) Diffusion Centrality: Approximation of communication centrality. Highly correlated to it and possesses its predictive properties but requires much less data. Measure the effectiveness of alternative injection points for microfinancing loans. It is formulated as:

$$DC(g; q, T) = \left[\sum_{t=1}^{T}(qg)^t\right]$$

where g is adjacency matrix, q is probability and is the no. of iterations.

33) Gravity Centrality: Inspired from the classic formula of gravity. Taking k-shell value as mass and shortest path distance as distance. Finding node importance and influential spreaders in the complex networks. It is formulated as:

$$G(i) = \sum_{j\in\rho\psi_i} \frac{ks(i)ks(j)}{d_{ij}^2}$$

where ks(i) is the k-shell value of node i.

34) DIL Centrality: Node importance is determined based on local information. Firstly, importance of edge is computed then we compute contribution of node in its importance. Finally degree and contribution of nodes is used to determine their importance. Used mainly for bridge nodes in large scale complex networks with less computation cost. It is formulated as:

$$L_{vi=k_i} + \sum_{v_j \in \Gamma_i} W_{v_i v_j}$$

where Γ_i is set of neighbors of node v_i, k_i is degree of node v_i and W_{vivj} is the contribution of v_i in importance of e_{ij}.

35) Density Centrality: Based on the formula of Area density, it is computed by taking degree and distance between two nodes in neighborhood of order r=1,2,3 etc. it impacts connectivity as well as transmission of the node in the network. Used in Complex networks but can be extended to weighted and directed networks. It is formulated as:

$$D_c(i) = \sum_{j \in \xi_i} \frac{k_i}{\pi d_{ij}^2}$$

where ξi is neighborhood set with at most r distance to node i, d_{ij} is shortest path distance between i and j and k_i is degree of node i.

36) Laplacian Centrality: Defined as the amount of drop seen in the Laplacian energy when a node is removed from the network. Laplacian energy is the sum of the squared eigen values of the Laplacian matrix. The Laplacian matrix is obtained by scalar subtraction of weight matrix from the degree matrix of the graph. Used for faster results in weighted networks as it has linear computation time. It is formulated as:

$$E_L(v_i, G) = \frac{E_L(G) - E_L(G_i)}{E_L(G)}$$

where $E_L(G)$ is Laplacian energy with node i and $E_L(G_i)$ is Laplacian energy without node i.

37) Percolation Centrality: Considering the percolation state of nodes in a network along with topological connectivity. Node importance is measured by how impacting it can be in aiding the percolation in the network. In complex networks while percolation scenarios such as infection or disease spreading. It is formulated as:

$$PC^t(v) = \frac{1}{(N-2)} \sum_{s \neq v \neq r} \frac{\sigma_{s,r}(v)}{\sigma_{s,r}} \frac{x_s^t}{[\sum x_i^t] - x_v^t}$$

where $\sigma_{(s,r)}(v)$ is shortest distance between s,r via v and $\sigma_{(s,r)}$ is shortest distance between s and r.

38) Opinion centrality: Designed specifically for multiplex networks such as online social networks, marketing etc. Firstly, opinion diffusion in the central node is done in the network followed by optimization to achieve maximal influence. This could be used to invest right amount of external influence in a node to maximize the opinion of the entire social group. It is formulated as:

$$\lambda_i^0 = RI^{-1}$$

where R is resource budget and I is matrix containing initial index assigned to each node.

39) Perturbation Centrality: Measure centrality in dynamic network. It is defined as the reciprocal of silencing time derived using a Dirac delta type of starting perturbation with 10*n units, where n is the no. of nodes. It is formulated as:

$$PC = 1/t$$

where t is silencing time.

40) Valued Centrality: For valued networks with varying strengths. Similar to closeness centrality with a correlation factor of 0.97. It is the average of the reciprocal of the sum of the maximum path distance between nodes in the network. It is formulated as:

$$C_V(x) = \frac{1}{n-1}\left(\sum_{y \neq x} \frac{1}{d(x,y)}\right)$$

where d(x,y) is maximum distance between node x and node y and n is number of nodes.

3.2 Centrality measures and real-world networks

Centrality measures have been designed to treat different networks differently. Networks may vary in terms of size, complexity, layers, topology etc. and based upon these factors the criteria which decides importance of nodes in these networks may also vary. In order to appropriately understand the network and provide best results different set of centrality measures have been used throughout the large variety of networks. This application of centrality measures across various networks is mentioned in the table below along with the approach behind using them, the domain these networks can be categorized into, datasets used and the results achieved as shown below:

S.no	Name	Approach used	Centrality measures used	Domain	Datasets	Results
1.	Urban Rail Networks	Analysis of centrality characteristics of Shanghai Rail network based on total passenger flow	Degree, betweenness, closeness	Transportation	Shanghai Urban Rail Network with 12 lines and 296 stations	Closeness centrality outperforms others
2.	Air Networks	Analysis of airline network performance against other airlines and identify competitors	Degree, Closeness	Transportation	Shri Lanka airline network of 2017 with 41 destinations and 820 routes	Removing critical point effects both degree and closeness centrality
3.	Airport Network	Analyse airport network in India for planning more efficient, secure and robust traffic, tourism and controlling spreading of a disease	Degree, Closeness, Betweenness	Topology	Airport Network of India with 84 airports and 13909 weekly direct flights and number of swine flue cases as per the Health Ministry of India	Developing more local hubs can reduce traffic and promote healthy competition among the airports
4.	Space Satellite Networks	Time-varying graph model and temporal centrality metrics to identify critical components in high-dynamic space central network	Temporal	Satellite Traffic Management	Leo satellite network with 20 randomly distributed low orbit satellites at semi-major axis range	Balanced traffic dispersion reduces vulnerability and improves robustness
5.	Distance vector routing	Next hop selection based on least centrality measurement in a 2-hop local graph	Degree, betweenness, subgraph	Wireless communication	Simulation model with 100 randomly allocated nodes and undirected links with geodesic distance below the threshold in a squared area	Betweenness centrality outperforms others in providing minimum of maximum number of relay

					squared area.	
6.	Spam SMS Detection	Compare centrality measures to detect spam SMS. Degree centrality achieves highest precision and recall among all	Degree, Closeness, Eccentricity	Spam detection	A real world dataset containing 1115 messages out of which 150 are spam and 965 ham messages	Degree centrality performs best among all centrality with precision and recall of 0.81 and 0.76 respectively and 94.4% accuracy
7.	Human Metabolic Core Analysis	Topological analysis of genome-scale metabolic networks. Hub based centrality measures to find important metabolites, therapy design and drug target	Degree	Metabolic network	Ma’s model to relate human metabolism and diseases. Database used is Kegg Ligand. The metabolic core used contains 256 vertices and 649 edges	Top 10 metabolites are studied for their therapeutic and biological significance
8.	Microblog network	Find central nodes in microblog network using the combination of the three centrality metrices	Degree, closeness, betweenness	Social network	Sina weibo API and user data from Fudan University, Dec 2012. Microblog network graph created contains 3131 nodes and 47376 edges.	All three centralities combined achieve the best results

9.	Doctor-patient Network	Considering length of relation and number of links in the network. The time based relation represented as weighted graph used to analyze network instability and inconsistency	Time-scale degree (TSDC)	Time-variant approach to degree centrality	Doctor-patient network with 2556 inpatient entries of patients over 60 years as provided by Hospital Contribution Fund over a period of 2005 to 2009	TSDC approach provides same results as degree centrality at macro level but at micro-level it proves to perform better
10.	Content-Centric Network	Effective location selection using centrality for content-centric network router using shortest-path and network topology information	Degree, closeness, betweenness, eigenvector	Network routing	Barabási–Albert model with 100 randomly generated nodes connected by 291 edges	Betweenness centrality provides best results in location selection for CCR routers
11.	Petri Nets	Static and dynamic network centrality analysis using General-purpose Petri-net simulator to see the bottlenecks and the reason behind them in the system	Degree, betweenness, closeness	Industry	Flexible manufacturing system used as Petri Net model in General-Purpose Petri Net Simulator	The three centrality measures help in performance analysis of the model along with an overall i.e. static and dynamic analysis of the system.
12.	Vehicular Ad-hoc networks	Data dissemination protocol for better traffic management and faster and more	Edge-betweenness	Wireless communication	Veins and OMNeT++ framework with IEEE 802.11p as	A 95% confidence interval is obtained by SUMO.

		secure communication among vehicles with up to 90% network coverage.			base protocol. Simulation of Urban Mobility (SUMO) simulator is used to run 33 simulations.	DDBC performs better than CARRO and UV-CAST in terms of number of transmissions, coverage and delay.
13.	Wireless Sensor Networks	Variant of betweenness centrality for quality of service based topology control by evaluating edge to edge relationship hence controlling energy dissipation, latency, message delivery and information flow among nodes	Edge-betweenness, closeness	Wireless communication	A simulation model based Java-based simulation environment. AODV routing protocol is deployed. IEEE 802.11 is used as MAC protocol with wireless bandwidth of 2Mbps.	Wireless Sensor Networks [69]
14.	Wavelength Division Multiplexing Networks	Load balancing by finding possible traffic concentrated nodes for uneven traffics and perform link-cost adjustment, the recalculation results in modified betweenness centrality. Drastic reduction in blocking probability	betweenness	Network routing	20 nodes and 60 links are used in a random network. 16 wavelengths are used in random wavelength assignment algorithm. The request pattern is Poisson working	Wavelength Division Multiplexing Networks [70]

					on OSPF routing algorithm.	
15.	Protein interaction network	Studying the mutational divergence-and-duplication and dependence of the two centrality measures in different protein evolutionary stages in the protein network of unicellular organism eukaryote (a type of yeast).	Degree, betweenness	Biology	A high confidence dataset with 2561 protein links and 5996 interactions. [92] The model has two parameters alpha and beta for removing and forming interactions.	Protein interaction network [83]
16.	Multiplex Networks	Analyze the influence on eigenvector centrality by the complex networks with heterogenous interactions through multiple layers network structures. The author tends to introduce how differently the centrality measures perform in multiplex networks and how they are potentially applicable.	Uniform eigenvector	Multi layered networks	The measure of multiplexity of the network is determined by the heterogenous centralities. Randomly generated multiplex networks are tested against different types of centrality measures under different influences.	Multiplex Networks [85]
17.	fMRI data of human brain	Functional magnetic resonance data of	Betweenness, eigenvector	General	Two experiments were made with	Eigen vector centrality has

		human brain analysis proven to be more efficient using eigenvector centrality over betweenness centrality for capturing the intrinsic neural architecture on voxel-wise levels.			52000 and 40000 voxels respectively. Functional EPI/MRI data of 35 and 22 normal volunteers are used respectively.	shown parameter free, fast and independent of presumptions which makes it a valuable model free toolbox.
18.	EEG based Biometric system	Estimate the importance of a node in a functional brain network by spectral decomposition of weighted connectivity matrices. The proposed method takes place in four steps: 1. Band pass filter the EEG data 2. Estimate functional interdependence 3. Reconstruct brain network 4. Characterize the network topology	eigenvector	General	The EEG signals dataset open access is used. EEG recordings are obtained from 64 channel BCI2000 system. Scikit-learn for Python is used for analysis.	The resting-state functional brain network provides better results for classification than using only functional connectivity
19.	Bitcoin Transaction Graph	Main objective is to link the address with the coin based transactions via associated unspent transaction outputs thereby identifying most central addresses in	mint	General	Public bitcoin blockchain of 200,00 blocks are used. 50 addresses with high mint centrality are	Out of 50 addresses, 18 belong to satoshidice, 25 belong to luckybit and satoshidice and 7 belong to

		in bitcoin transaction graph.			inspected and labeled using bitcoinwhoswho. com and blackchain.info	Gavin's original bitcoin faucet and Wikileaks donation address
20.	Urban Street Networks	Spatial analysis of 18 different cities around the world providing visual characterization of a city's structural properties and proving that planned and self-organized cities belong to separate classes identify scale-free properties as seen in degree distribution of relational networks	Betweenness, closeness, straightness, information	General	Taking 18 1-square mile samples of different cities, constructed spatial graphs of street networks. In this graph nodes are intersections and edges are streets.	Self-organized cities show scale free properties unlike planned cities.
21.	Community discovery	Based on significance of edges thus converting centrality metrices for nodes to those for edges. Information centrality used to determine the network efficient before and after certain edge removal. Remaining measures are compared based on the methodology that one edge removal causes a drop of 2 degrees in the graph.	Information, closeness, betweenness, straightness, clustering coefficient	General	The proposed method is tested on 4 different networks namely Zachary's karate club, American college football teams, food webs and Primate networks.	Information centrality outperforms other centrality measures. Edge centrality for walks show high time complexity

22.	Cytoscape	A new plug-in called ModuLand for the determination of overlapping network modules and key network positions in biological networks. Centrality is used for the measure of influence of the network as a whole on its nodes and edges	Weighted degree, betweenness, community	Bioinformatics	The plug-in is written in C++ language with a Java language based graphical interface. It is supported by all major OS like Linux, Windows and Mac-OS.	The plug-in is helpful in identifying largely overlapping modular structures and providing simpler functional annotation.
23.	Mobile Social Networks	A time-ordered aggregation model used to convert dynamic network to a series of time-ordered networks. Then using the combination of average, linear and exponential time-ordered aggregation methods to determine node importance	Cumulative Neighboring Relationship	Mobile Social Networks	Proposed TCNR metric is compared with metrices in Infocom 06 and MIT Reality real mobility traces. Message routing is done by flooding.	TCNR provides more accurate results and hence outperforms than the other two existing aggregation methods
24.	Smart Cities	Dealing security and efficiency issues in smart city networks by - determining specialized nodes - detecting DoS attacks in real time optimizing network traffic	betweenness	IoT	Gephi is used to run the Ulrik Beande's algorithm which obtains the normalized and unnormalized score for all 21 nodes	The algorithm takes O(n+m) space and O(nm) time for unweighted graph and O(nm+n^2logn) time for weighted graph with n nodes and m links

25.	Electrical networks	Compensating reactive power for minimizing losses at different buses in generalized electrical networks using centrality indices of different test systems	Electrical closeness, electrical betweenness	Electrical power system	5-bus, IEEE 14-bus and IEEE 30-bus test systems are used. Shortest path searching is done through Johnson's algorithm inbuilt in MATLAB	Electrical networks [73
26.	Call graph of Java source code	Centrality analysis of complex network model derived from call graph of java program with nodes representing functions and edges representing call events between the functions	Betweenness, closeness, eccentricity, PageRank, HITS	Complex network, Small-world graph	The call graph contains 724 nodes and 1025 links. Pejak Program is used for its topological characteristics. Gephi toolbox is used for computing centers of network	Call graph of Java source code [74
27.	Hidden Link Prediction	Importance of common neighbors depend on their centrality. Centrality measures are used to find connection probability between nodes. Prediction accuracy is further enhanced by combining the result with weak-tie theory	Degree, closeness, betweenness	General	5 real world networks are considered namely USAir, NetScience, PowerGrid, Yeast and C. elegans	Hidden Link Prediction]

28.	BBS Reply Networks	Determine central nodes in network and analyze effect by their removal from the graph. There is degree and betweenness centralization reduction thus showing their significance and power in the complex network	Degree, betweenness	Complex Networks	MATLAB is used for analysis. There is increase in path length and decrease in clustering coefficient	Central nodes possess high prestige in the network like hub in shared network.
29.	Unsupervised Feature Selection	Unsupervised graph based feature selection first involves feature selection and their clustering using subspace. Followed by ranking them as per their centrality score after applying Page Rank algorithm. A multivariate method which effectively discriminates features.	Page Rank	Artificial Intelligence	High dimensional Low data size dataset is used with 7000 features and 10000 objects for the process. 4 data sets used are Colon, ovarian cancer, leukemia and CLL_SUB_111.	High classification for 150 colon and 500 ovarian cancer leukemia datasets. Slow but more accurate results are obtained.
30.	Co-citation Networks	In the undirected and weighted co-citation network nodes are authors and edges are papers and co-citation frequencies are weights for respective edges. Weighted Page Rank algorithm designed for undirected graph is used to rank the authors based on the citations.	Degree, betweenness, closeness, Page Rank	General	108 most cited authors between 1970 to 2008 are selected inn the area of Information Retrieval. The Page Rank damping factor is kept in the range of 0.05 to 0.95.	High correlation of Citation Rank with Page Rank of different damping factors. Low correlation of Page Rank and Citation Rank with other centralities.

Table 1. Description of real world networks

CHAPTER 4: Implementation

The implementation of the approach involves six phases namely pre-processing of data to filter out noise, vectorization to convert data into VSM form, application of soft-cosine similarity measure on the generated VSM to get the similarity matrix, similarity graph generation for the the matrix, centrality computation using the best centrality measure among the four centrality measures compared and finally the generation of the ordered list of sentences as per the derived centrality scores.

4.1 Methodology

Our method involves sentence boundary discrimination followed by sentence ranking and finally sentence selection. Fig 2. shows the visual overview of our approach. Detailed description of our approach is given below:

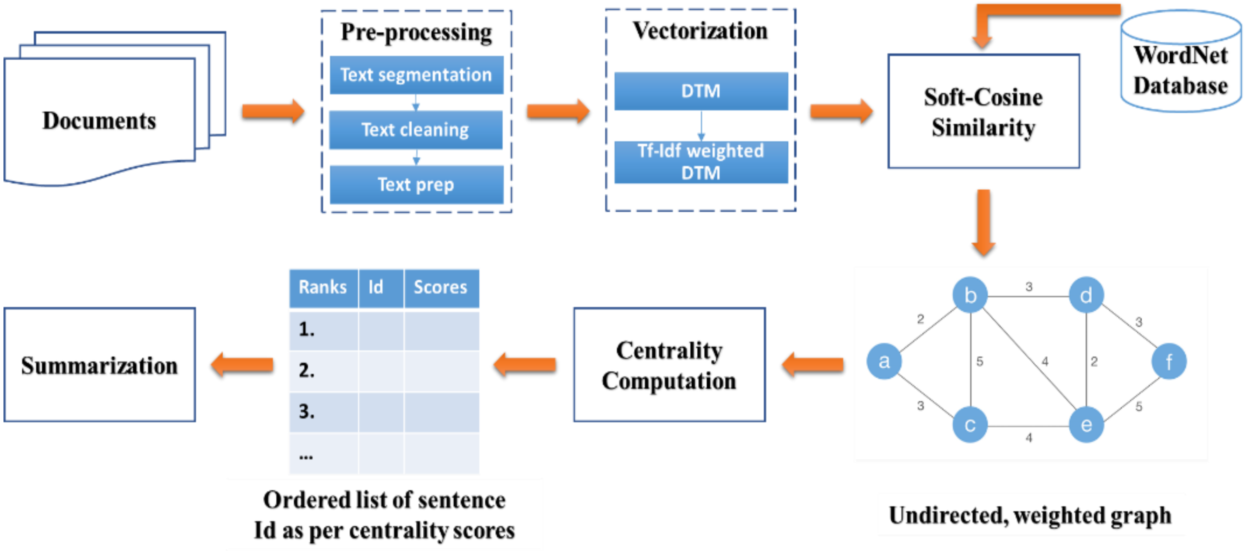


Fig. 3. Architecture of our text summarization approach

4.1.1 Pre-processing

Before the actual process of text summarization, the data is filtered. This is divided into three major tasks:

2. *Text segmentation*- the text is segmented into sentences using '.', '?' and '!'. Each sentence is further broken into words called tokens. Short sentences with less than four words in it are removed in this step as they are least likely to hold much importance .
3. *Text cleaning*- it includes removal of noise e.g. tags (<TEXT>, <P> etc.), emoticons(😆, ☹ etc.), punctuations (;,:- etc.) and other symbols (~`"#^* etc). Certain symbols which add meaning to the sentence are replaced by their word equivalents e.g. '@' as at, '$' as dollar etc. Abbreviations, contractions and ordinals are also converted to their long forms.
4. *Text prep*– it involves lower casing, stop-words (e.g. 'a', 'the' etc.) removal, Parts-of-Speech tagging and stemming using Porter's stemming algorithm .

4.1.2 Vectorization

The tokens are then used to form a sparse matrix called document-term matrix (DTM) which holds the frequency of words per document. In order to fix distortion due to less important terms and provide an unbiased estimate of importance of the terms the DTM is weighted using term frequency-inverse document frequency (tf-idf) . It can be expressed as:

$$Weight_{DTM} = \left(1 + \log tf_{t,d}\right). \log \frac{n}{df_t}$$

where n is the number of sentences. Each row in the weighted DTM forms vectors for each sentence in an n-dimensional vector space.

4.1.3 Similarity Calculation

The tf-idf weighted DTM is then used to find similarity between the sentences using the soft-cosine similarity measure . Just like cosine similarity measure, soft-cosine similarity measure also determines the similarity between two sentences based on the orientation of the sentence towards the other in a multi-dimensional array where each sentence represents a dimension. But unlike cosine similarity measure, soft-cosine similarity doesn't consider sentences to be independent of each other rather it takes into account the semantic relatedness between the sentences .

Clearly soft-cosine similarity works as cosine similarity when there is no similarity between the features of the terms, making it a special case of cosine similarity. Since there is almost always some similarity between the two terms which can be identified using dictionary , soft-cosine similarity is a considerable alternative to regular cosine similarity. A semantic relation is a relation between two words based on their meanings. Here we are using the WordNet taxonomy to derive such relations . The types of semantic relations in WordNet are discussed in detail in Table 1.

Semantic Relation	Description	Examples
Synonymy	Similar sense relation or synonyms (synsets)	true, genuine
Antonymy	Opposite sense relation of antonyms	liability, asset
Hyponymy, Hypernymy	Super-subordinate (ISA) relation	lion, animal
Meronymy, Holonymy	Part-whole (HASA) relation	book, library
Troponomy	Manner relation among verbs	nibble, eat
Entailment	Propositional relation among verbs	dawn, morning

Table 2. WordNet taxonymy of semantic relations

4.1.4 Graph Generation

The soft-cosine matrix obtained for the sentences is used to construct a weighted, undirected graph. There is almost always some similarity between sentences thus the similarity graph behaves as a highly connected graph. In order to show only relevant sentences, low valued links can be eliminated using threshold. An appropriate threshold value can reduce noise while preserving important links. LexRank with threshold method has clearly proven 0.1 to be a good threshold value . Since our approach works on similar grounds we are using the same threshold value.

4.1.5 Centrality Measures

Important nodes in the similarity graph are identified using the centrality measures. Since we are using weighted graph to calculate centrality measures, both number of links i.e. degree and weights of those links i.e. strength of the nodes need to be considered. We are comparing four centrality measures namely weighted degree, closeness and betweenness centralities and eigenvector centrality. Former three measures use the concept of tuning parameter to adjust the trade-off between degree and strength. This tuning parameter α is a positive real number and its effect on the measure is as illustrated below:

$$\alpha = \begin{cases} 0 : Considers\ degree\ only \\ < 1 : Degree\ contributes\ proportionally \\ 1 : Considers\ strength\ only \\ > 1 : Degree\ contributes\ inversally \end{cases}$$

The VSM created by soft-cosine similarity holds explicit knowledge which can be easily codified, thus many weak ties are more relevant than fewer strong ties and number of intermediary nodes is more important than the link weights while calculating distance between two nodes . For such knowledge the value of α < 1 is more preferable . Thus we are taking α = 0.5 for calculating centrality measures:

5. ***Weighted Degree Centrality:*** it identifies centrality of a node with its degree i.e., number of edges connected to that node. Degree centrality for a node in a weighted network can be defined as the product of the number of nodes it is connected to and the α-adjusted average of their weights.

$$C_D^{w\alpha}(i) = \sum_{j=1}^{n} a_{ij} \times \left(\frac{\sum_{j=1}^{n} w_{ij}}{\sum_{j=1}^{n} a_{ij}} \right)^{\alpha}$$

6. ***Weighted Closeness Centrality:*** it considers the sum of the geodesic distances which is the shortest path length along the manifold. Using inverted weights transformed by α helps involve both number of intermediary nodes and link weights to find the length of the path.

$$C_C^{w\alpha}(i) = \frac{1}{\sum_{j=1}^{n} min\left(\frac{1}{(w_{ih})^{\alpha}} + \cdots + \frac{1}{(w_{hj})^{\alpha}} \right)}$$

7. ***Weighted Betweenness Centrality:*** it considers all possible geodesic paths between pairs of nodes. The centrality measure of the given node is then obtained by counting the number of such paths. Based on the method used above it can also use generalized shortest distance.

$$C_D^{w\alpha}(i) = \frac{g_{jk}^{w\alpha}(i)}{g_{jk}^{w\alpha}}$$

8. ***Weighted Eigenvector Centrality:*** it is determined by the eigen vector of the largest eigen value of the adjacency matrix. It is the weighted sum of direct and indirect connections of every length. Unlike conventional centrality measures where each link possesses equal weight, the eigen vector weighs links according to their centralities.

$$\sigma_{E_j} = \frac{1}{\lambda_1}\sum_i a_{ij}\, v_i)$$

where *i,j,k,h*: nodes of the similarity graph
n: total number of nodes
a_{ij}: adjacency matric
w_{ij}: weighted adjacency matrix
$g_{jk}^{w\alpha}$: shortest path between nodes i and j
$g_{jk}^{w\alpha}(i)$: paths passing through node i
λ_1: maximum eigen value for A
v_j: maximum eigen vector for A

4.1.6 Ranking

The best centrality measure out of the four is applied on the similarity graph to identify central nodes. The nodes are ranked as per their centrality scores and top ranked nodes are taken to form the summary such that it doesn’t exceed the specified limit of words.

4.2 Datasets and Metrices

4.2.1 Data Sets

We are using two datasets for evaluation of our method:

- *BBC Dataset:* The British Broadcasting Corporation (BBC) news dataset contains 2225 news articles from year 2004-2005 divided into different domains- Business (510 articles), Entertainment (386 articles), Politics (417 articles), Sports (511 articles) and Tech (401 articles) . Each articles is accompanied by one human generated gold standard summary.

- *DUC Dataset*- The Document Understanding Conferences (DUC) 2007 dataset is also used containing 45 topics each with 25 articles. There are 2 to 4 human-written gold standard summaries for each articles written by one of the human assessors out of the 10 human assessors. Each summary is kept under 250 words .

4.3 Hardware and Software

For the computation purpose the I have used hardware with following specifications:

Category	Specification
Processor	Intel core i5
Hard Drive	20 GB
Memory	8GB

Table 3. Hardware specifications for the approach

The source codes are run using the software specifications given below:

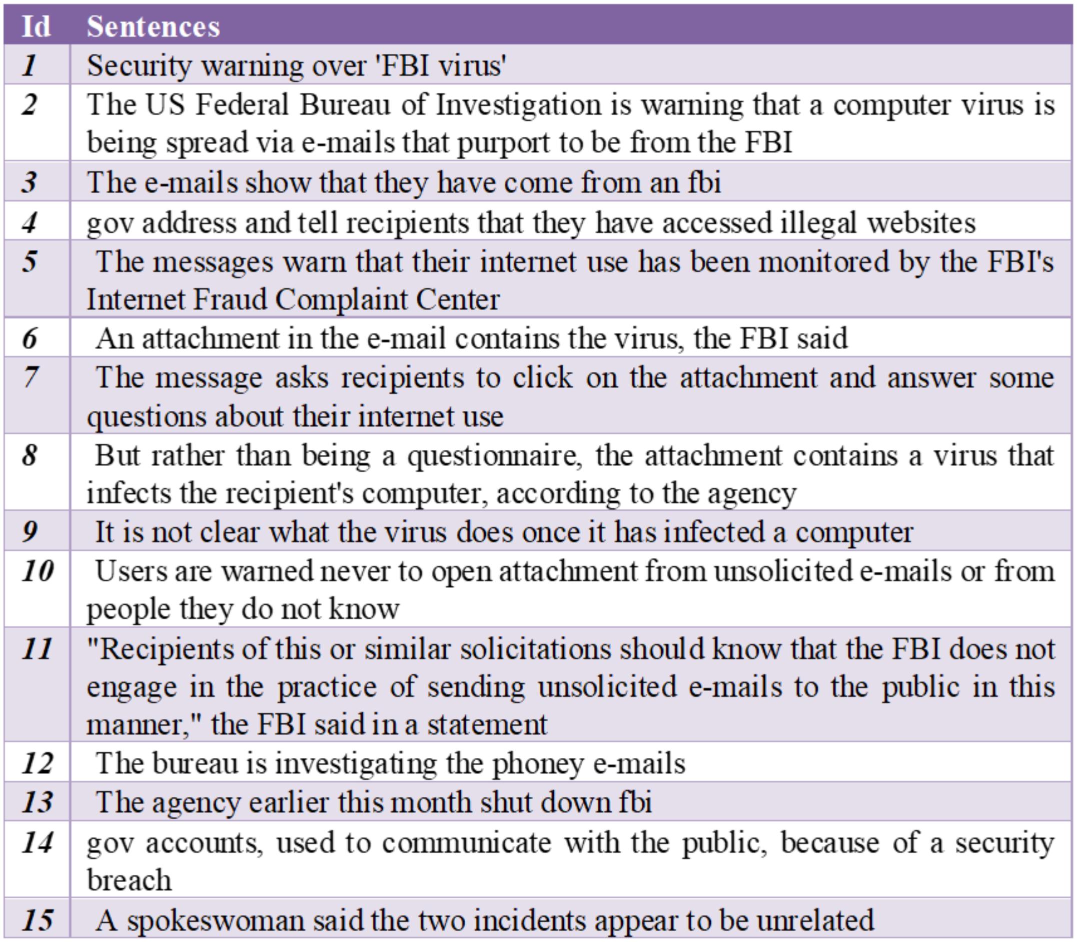
Category	Specification
Operating System	Windows 10
Programming Software	RStudio, Jupyter notebook
Programming languages	C++,R, Python

Table 4. Software specifications for the approach

Furthermore the major packages used in the RStudio are CINNA, igraph, SnowballC, tm, word2vec and the major modules used in the Jupyter notebook are networkx, rouge, sumy.

CHAPTER 5: Demonstration

Due to the limiting space we are taking a small sample document 334.txt from BBC news articles in tech section with 15 sample sentences as shown in Table 2. Soft-cosine similarity measures are applied on the tf-idf weighted DTM of the dataset. The similarity matrix obtained is represented as a weighted undirected graph with weights written over the corresponding edges in the Fig 4. In order to remove noise, the less important edges are removed through elimination by weights using threshold value of 0.1. The filtered graph is then used to calculate centrality scores for each node. Fig 5 shows the resultant graph with edge weight represented by its thickness and centrality scores of each node mentioned over it. The resultant summary is displayed against the human-generated gold-standard summary in Fig 6.



Id	Sentences
1	Security warning over 'FBI virus'
2	The US Federal Bureau of Investigation is warning that a computer virus is being spread via e-mails that purport to be from the FBI
3	The e-mails show that they have come from an fbi
4	gov address and tell recipients that they have accessed illegal websites
5	The messages warn that their internet use has been monitored by the FBI's Internet Fraud Complaint Center
6	An attachment in the e-mail contains the virus, the FBI said
7	The message asks recipients to click on the attachment and answer some questions about their internet use
8	But rather than being a questionnaire, the attachment contains a virus that infects the recipient's computer, according to the agency
9	It is not clear what the virus does once it has infected a computer
10	Users are warned never to open attachment from unsolicited e-mails or from people they do not know
11	"Recipients of this or similar solicitations should know that the FBI does not engage in the practice of sending unsolicited e-mails to the public in this manner," the FBI said in a statement
12	The bureau is investigating the phoney e-mails
13	The agency earlier this month shut down fbi
14	gov accounts, used to communicate with the public, because of a security breach
15	A spokeswoman said the two incidents appear to be unrelated

Table 5. Sample sentences from BBC news articles dataset in tech section

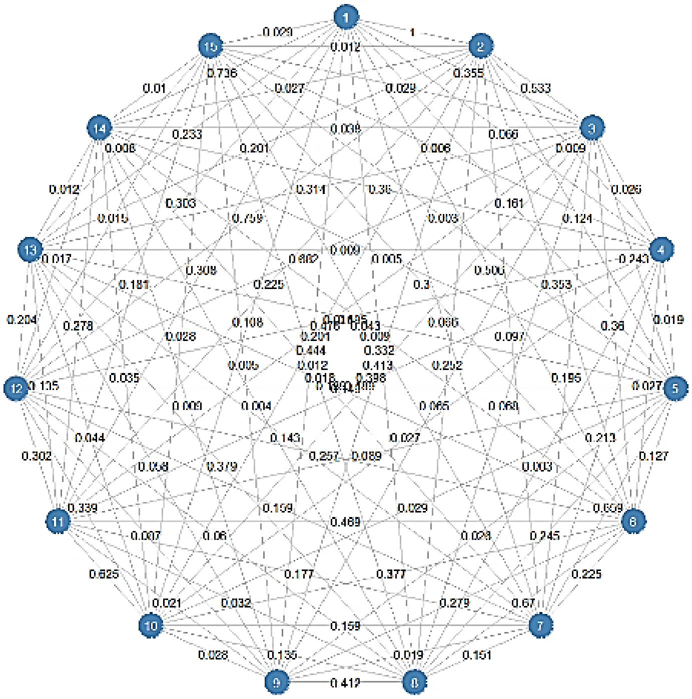


Fig. 3. Similarity graphs for sentences in Table 2. displays the weighted similarity graph with respective edge weights,

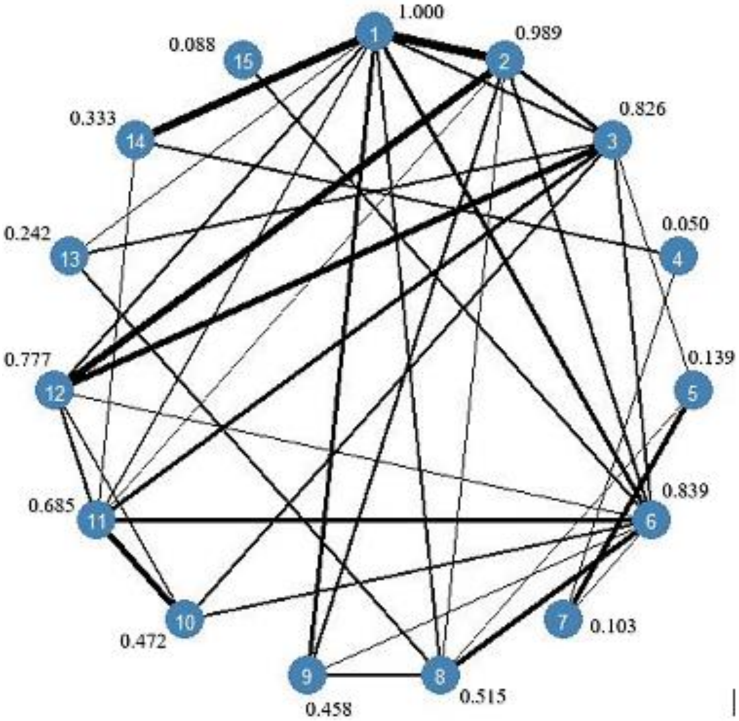
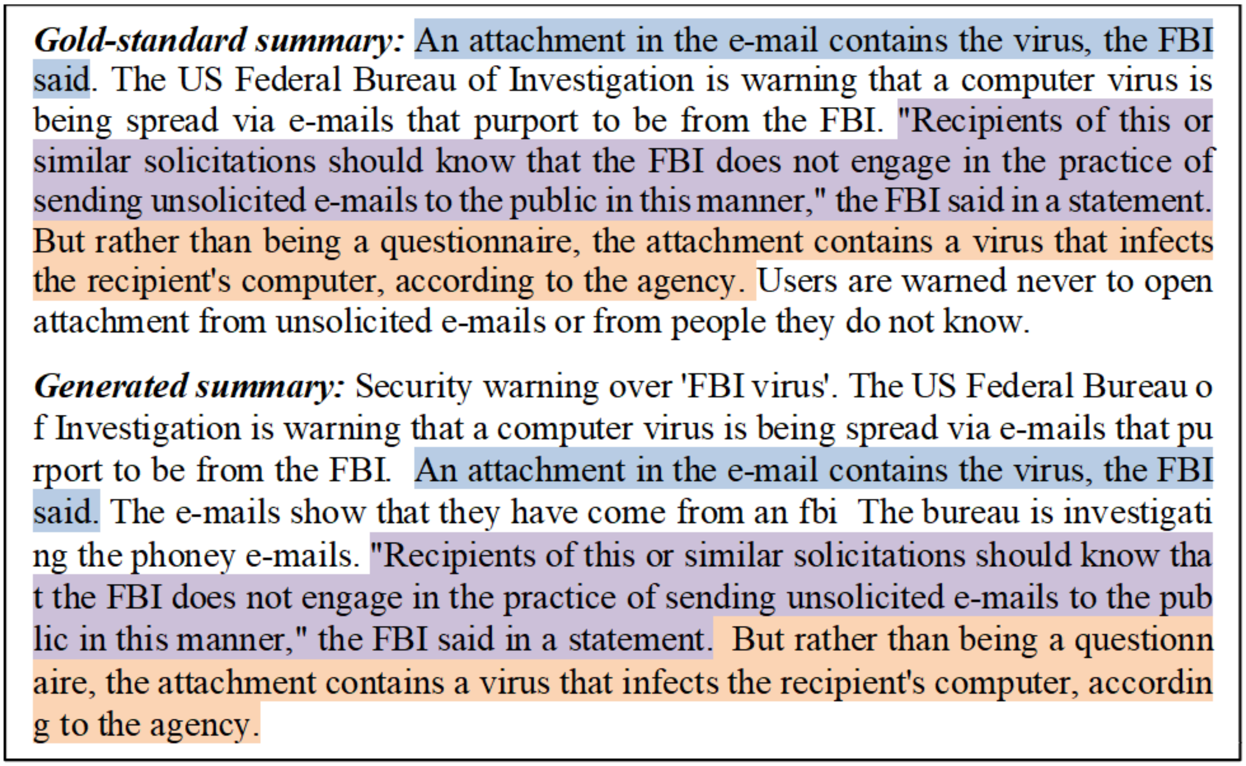


Fig 5. displays the centrality measures for each node with edges weights after threshold-elimination of 0.1.



Gold-standard summary: An attachment in the e-mail contains the virus, the FBI said. The US Federal Bureau of Investigation is warning that a computer virus is being spread via e-mails that purport to be from the FBI. "Recipients of this or similar solicitations should know that the FBI does not engage in the practice of sending unsolicited e-mails to the public in this manner," the FBI said in a statement. But rather than being a questionnaire, the attachment contains a virus that infects the recipient's computer, according to the agency. Users are warned never to open attachment from unsolicited e-mails or from people they do not know.

Generated summary: Security warning over 'FBI virus'. The US Federal Bureau o
f Investigation is warning that a computer virus is being spread via e-mails that pu
rport to be from the FBI. An attachment in the e-mail contains the virus, the FBI said. The e-mails show that they have come from an fbi The bureau is investigati
ng the phoney e-mails. "Recipients of this or similar solicitations should know tha
t the FBI does not engage in the practice of sending unsolicited e-mails to the pub
lic in this manner," the FBI said in a statement. But rather than being a questionn
aire, the attachment contains a virus that infects the recipient's computer, accordin
g to the agency.

Fig. 6. Human-generated gold-standard summary and summary generated through our method

CHAPTER 6: Results

5 Experimental Results

For the evaluation purpose, we are using the automatic summary evaluation metric ROUGE- Recall-Oriented Understudy for Gisting Evaluation . It is available for a variety of scoring criteria such as 1, 2, 3, 4, N-gram comparisons, -L for the longest common subsequence, -W which is similar to -L but is weighted by length, and -S for skip-bigram co-occurrence, -SU an extension of -S. We are using the unigram ROUGE-1, bi-gram ROUGE-2 and ROUGE-L metrices.

5.1 Comparison of centrality measures

For comparing the four centrality measures we are using Rouge-1 scores on DUC-2007. Only those 22 out of 45 clusters of DUC 2007 dataset are considered for which exactly four gold-standard summaries are available so that all centrality measures are tested for equal size of test data and against largest set of gold-standard summaries.

Results show that all centrality measures provided similar outcomes in Fig 6(a). But using dimensionality reduction of the centrality measures through Principle Component Analysis (PCA) it is clear that Eigenvector centrality outperforms others as shown in Fig 6(b).

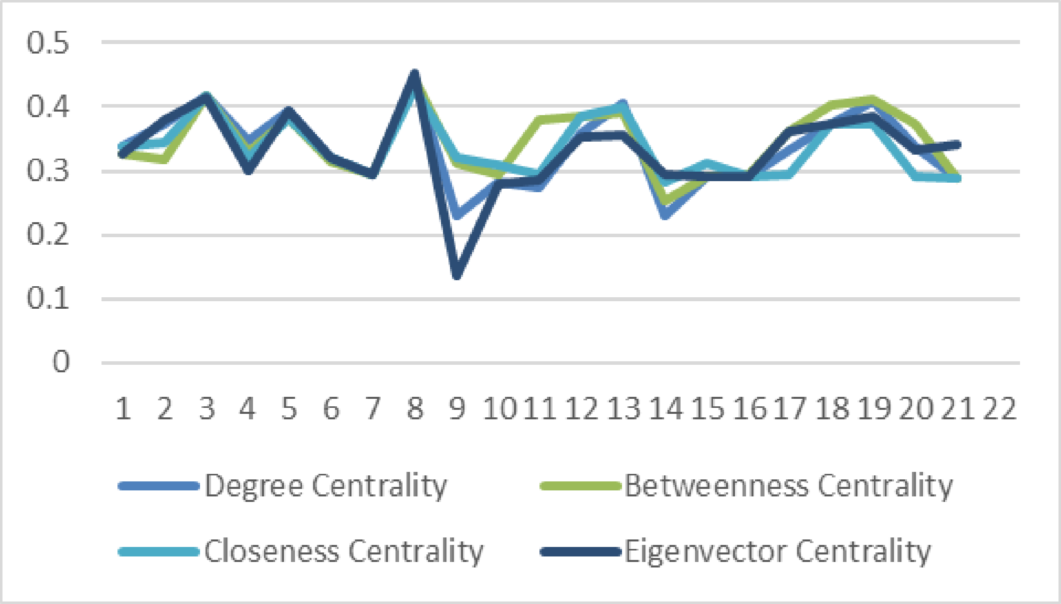


Fig 7. Comparison of ROUGE-1 scores obtained by centrality measures for clusters of DUC 2007 dataset with 4 gold-standard summaries.

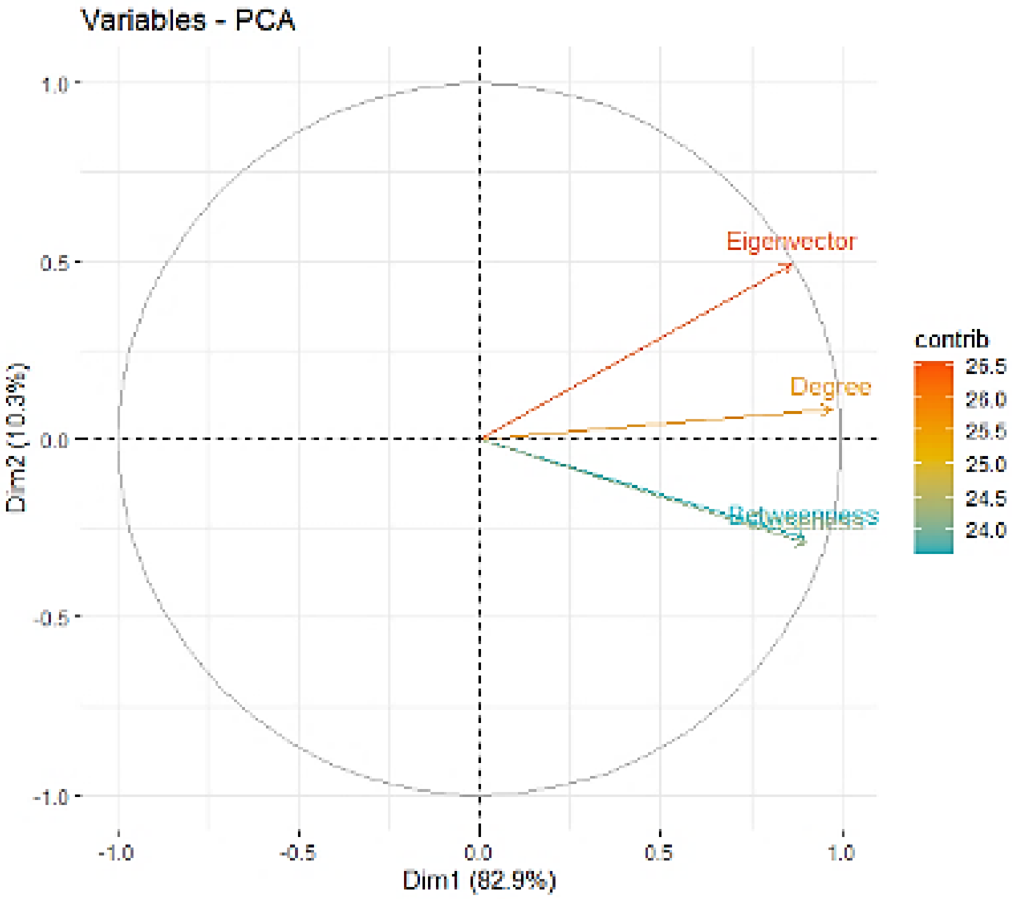


Fig. 8. PCA results comparison for all four centrality measures

5.2 ROUGE results

The quality of a summary is based on two main factors- readability and retainability. Extractive summarizers pick text directly from the original document thus its readability is as good as the original document. In order to detect how much information is retained from the original document we are using Recall-Oriented Understudy for Gisting Information (ROUGE). This metric statistically evaluates the summary against human-generated gold standard summary in order to determine its quality. We are using ROUGE-1 for unigram, ROUGE-2 for bi-gram and ROUGE-L for longest common subsequence in order to evaluate the results obtained. Our results are compared against the standard baseline text summarizers namely LexRank, TextRank, Luhn and LSA as displayed in Table 3 and 4.

The results in Table 3 and Table 4 are represented in graphical form in Fig 8 and Fig 9 respectively. This clearly depicts that our proposed method performs better than other automatic summarizers in most of the cases.

	BBC Dataset		
	ROUGE-1	***ROUGE-2***	***ROUGE-L***
Eigenvector	0.989619	0.989547	0.987952
LexRank	0.811534	0.803653	0.821275
TextRank	0.815859	0.804943	0.818737
Luhn	0.848293	0.828383	0.805217
LSA	0.825665	0.865065	0.875695

Table 6. ROUGE scores for BBC news articles dataset.

	DUC 2007 Dataset		
	ROUGE-1	***ROUGE-2***	***ROUGE-L***
Eigenvector	0.452504	0.128641	0.350685
LexRank	0.096475	0.053856	0.177198
TextRank	0.296244	0.103285	0.347298
Luhn	0.328156	0.120523	0.349871
LSA	0.432195	0.121325	0.356153

Table 7. ROUGE scores for DUC 2007 dataset.

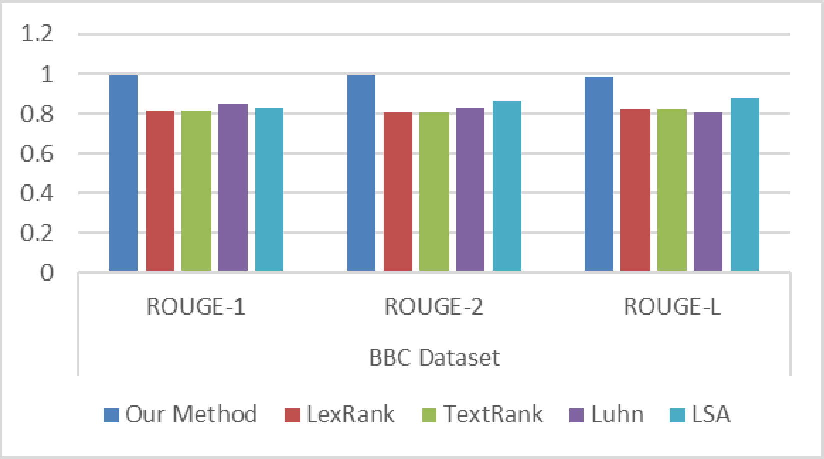


Fig 9. Graphical representations of ROUGE Scores for BBC news articles dataset

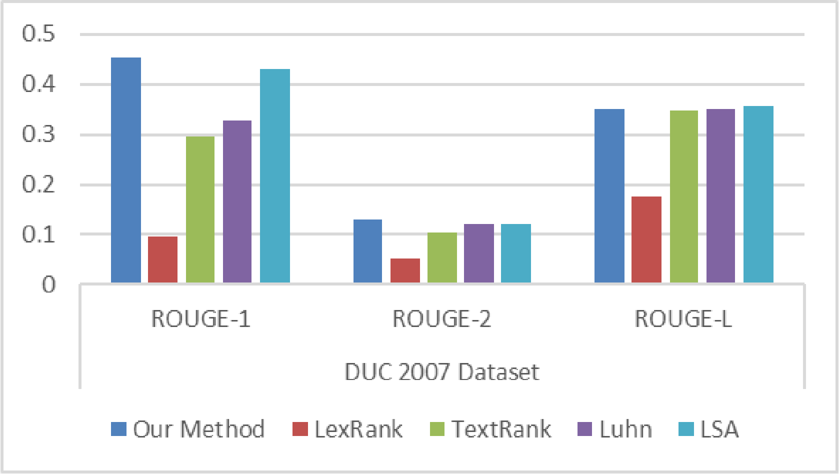


Fig. 10. Graphical representation of ROUGE scores for DUC 2007 dataset

Conclusion

I have presented an extractive summarization technique based on soft-cosine similarity and centrality measures. Soft-cosine similarity uses semantic knowledge to find relatedness between the sentences. Our method performs better than other text summarizers in general and at par with the LSA summarizer. This can be clearly understood as LSA also utilizes learning from data in order to provide better results. As mentioned earlier, LSA has the drawback that it need larger and less diverse dataset to provide decent results while our method doesn't has any such constraint. Also it learns from the data every time it is used whereas soft-cosine similarity only needs to load semantic knowledge once before applying. Thus our method holds an upper hand over LSA.

The major breakthrough achieved in this process is how we reduce the domain in which words were previously categorized just by realizing that several words could represent one and the same entity regardless or the context. Using this idea we have been able to achieve a good jump in results and compared to the ones produced by the previous state-of-the-art automatic text summarization techniques. Our approach has shown to give good results for both short and lengthy text documents. It has also proved to possess less overall time complexity as it needs to access semantic dictionary only once and can work using same knowledge base for all documents.

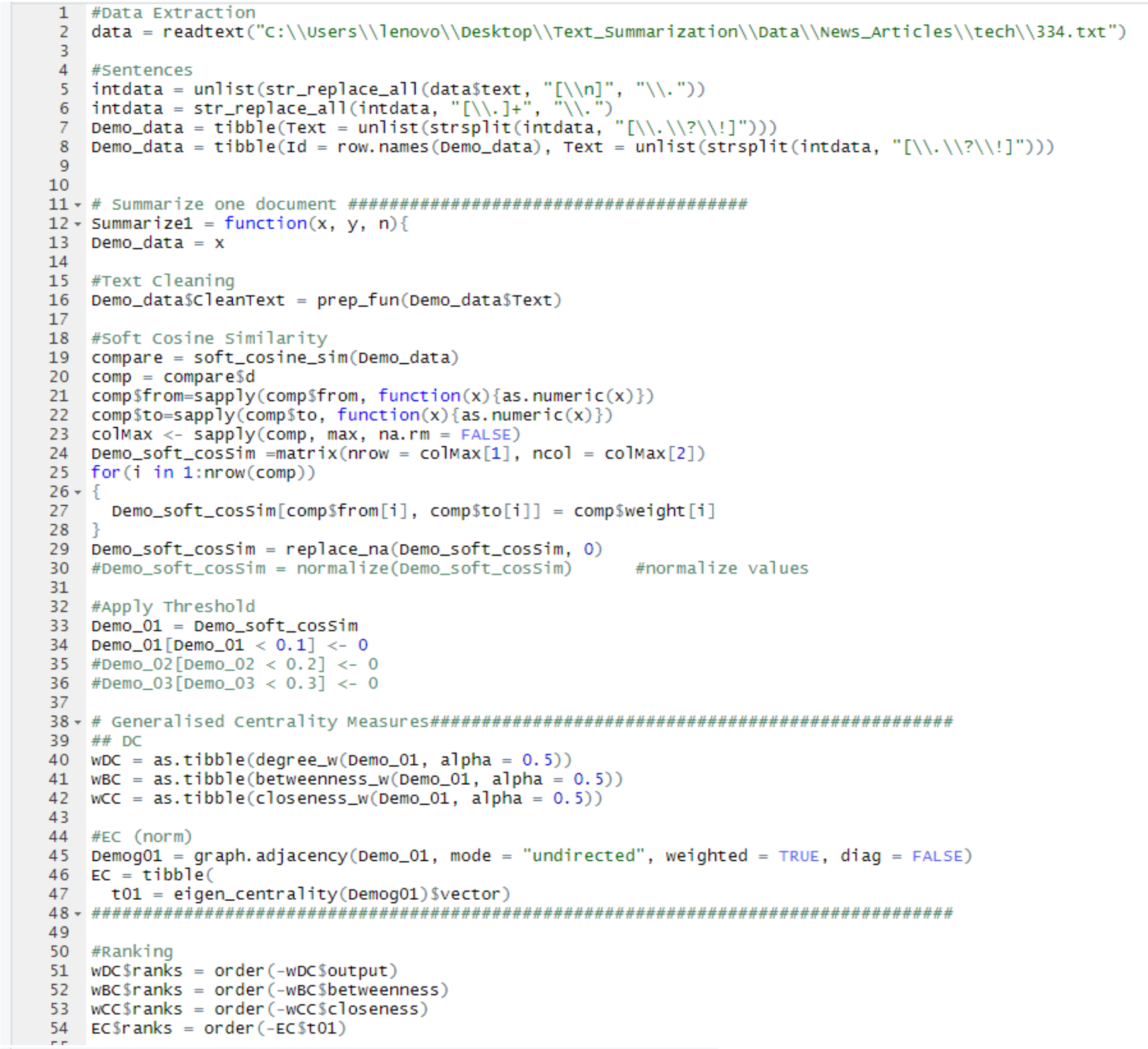
The limitation of our approach is that despite the improvement in results it is not as good as human generated summary. It does not take into account the fact that several sentences could be combined to produce one sentence depicting the essence of all those sentences. Moreover the initial training for the approach is dependent on external source i.e. dictionary which means it is not completely self-reliant.

Future Work

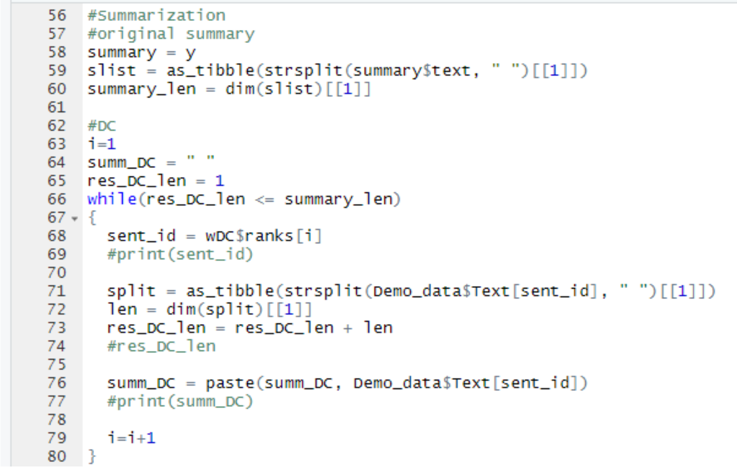
In future we would like to incorporate more types of semantic relations. We would also like to incorporate other methods of sentence scoring such as length, position, cue, frequency, co-occurrence etc. We would try to extend our method for other languages as well. We would also try to include other centrality measures to further tune the results. A linear or weighted combination of centrality measures could also prove to be helpful as different centrality measures may possess varying degree of impact on the results. Further we would like to extend our work to other languages using the respective set of dictionaries to get the semantic relation set present between the words in that language. This could be further improvised to work for multiple languages in one go and can be used in real world scenarios such as twitter, facebook, Instagram etc. where users generally use more than one language in same paragraph.

Appendices

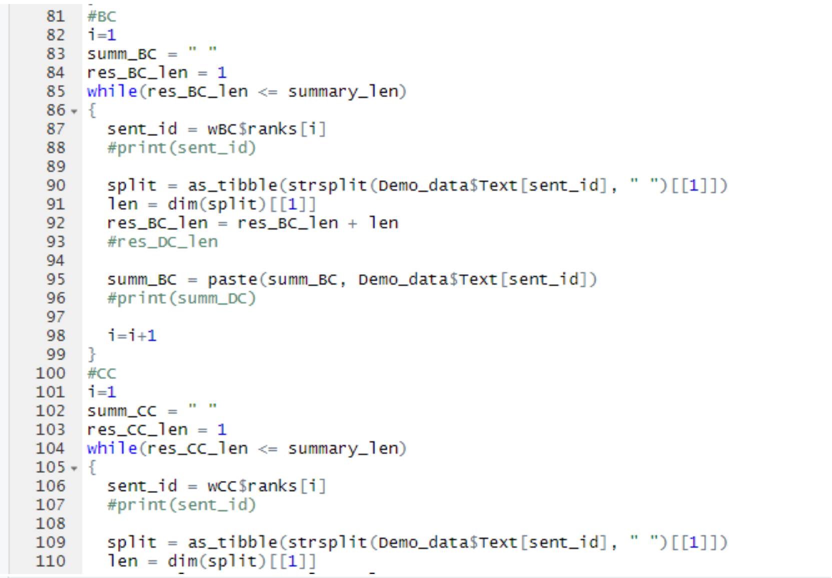
R code snippets:



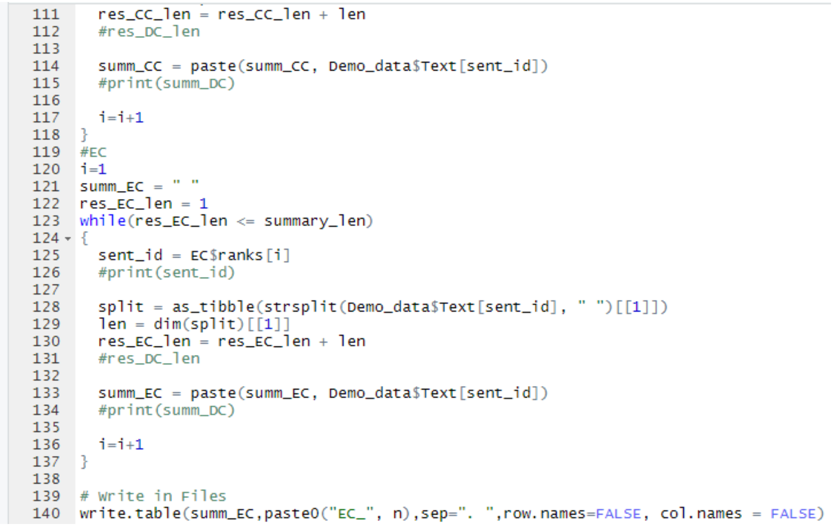
```r
#Data Extraction
data = readtext("C:\\Users\\lenovo\\Desktop\\Text_Summarization\\Data\\News_Articles\\tech\\334.txt")

#Sentences
intdata = unlist(str_replace_all(data$text, "[\\n]", "\\."))
intdata = str_replace_all(intdata, "[\\.]+", "\\.")
Demo_data = tibble(Text = unlist(strsplit(intdata, "[\\.\\?\\!]")))
Demo_data = tibble(Id = row.names(Demo_data), Text = unlist(strsplit(intdata, "[\\.\\?\\!]")))


# Summarize one document ######################################
Summarize1 = function(x, y, n){
Demo_data = x

#Text Cleaning
Demo_data$CleanText = prep_fun(Demo_data$Text)

#Soft Cosine Similarity
compare = soft_cosine_sim(Demo_data)
comp = compare$d
comp$from=sapply(comp$from, function(x){as.numeric(x)})
comp$to=sapply(comp$to, function(x){as.numeric(x)})
colMax <- sapply(comp, max, na.rm = FALSE)
Demo_soft_cosSim =matrix(nrow = colMax[1], ncol = colMax[2])
for(i in 1:nrow(comp))
{
  Demo_soft_cosSim[comp$from[i], comp$to[i]] = comp$weight[i]
}
Demo_soft_cosSim = replace_na(Demo_soft_cosSim, 0)
#Demo_soft_cosSim = normalize(Demo_soft_cosSim)      #normalize values

#Apply Threshold
Demo_01 = Demo_soft_cosSim
Demo_01[Demo_01 < 0.1] <- 0
#Demo_02[Demo_02 < 0.2] <- 0
#Demo_03[Demo_03 < 0.3] <- 0

# Generalised Centrality Measures##################################################
## DC
wDC = as.tibble(degree_w(Demo_01, alpha = 0.5))
wBC = as.tibble(betweenness_w(Demo_01, alpha = 0.5))
wCC = as.tibble(closeness_w(Demo_01, alpha = 0.5))

#EC (norm)
Demog01 = graph.adjacency(Demo_01, mode = "undirected", weighted = TRUE, diag = FALSE)
EC = tibble(
  t01 = eigen_centrality(Demog01)$vector)
###################################################################################

#Ranking
wDC$ranks = order(-wDC$output)
wBC$ranks = order(-wBC$betweenness)
wCC$ranks = order(-wCC$closeness)
EC$ranks = order(-EC$t01)
```



```r
#Summarization
#original summary
summary = y
slist = as_tibble(strsplit(summary$text, " ")[[1]])
summary_len = dim(slist)[[1]]

#DC
i=1
summ_DC = " "
res_DC_len = 1
while(res_DC_len <= summary_len)
{
  sent_id = wDC$ranks[i]
  #print(sent_id)

  split = as_tibble(strsplit(Demo_data$Text[sent_id], " ")[[1]])
  len = dim(split)[[1]]
  res_DC_len = res_DC_len + len
  #res_DC_len

  summ_DC = paste(summ_DC, Demo_data$Text[sent_id])
  #print(summ_DC)

  i=i+1
}
```

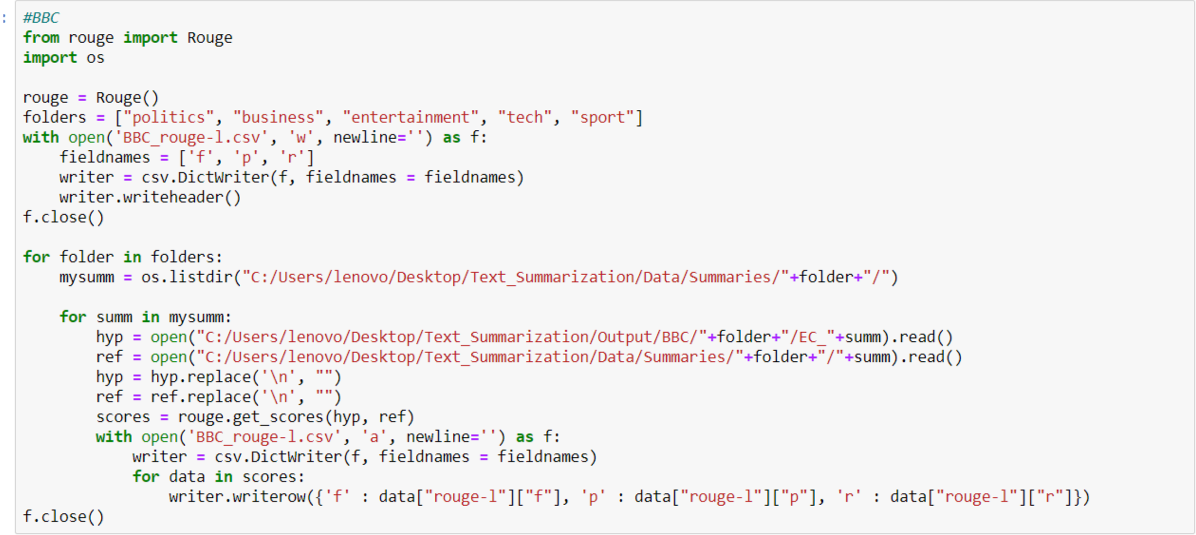


```r
#BC
i=1
summ_BC = " "
res_BC_len = 1
while(res_BC_len <= summary_len)
{
  sent_id = wBC$ranks[i]
  #print(sent_id)

  split = as_tibble(strsplit(Demo_data$Text[sent_id], " ")[[1]])
  len = dim(split)[[1]]
  res_BC_len = res_BC_len + len
  #res_DC_len

  summ_BC = paste(summ_BC, Demo_data$Text[sent_id])
  #print(summ_DC)

  i=i+1
}
#CC
i=1
summ_CC = " "
res_CC_len = 1
while(res_CC_len <= summary_len)
{
  sent_id = wCC$ranks[i]
  #print(sent_id)

  split = as_tibble(strsplit(Demo_data$Text[sent_id], " ")[[1]])
  len = dim(split)[[1]]
```

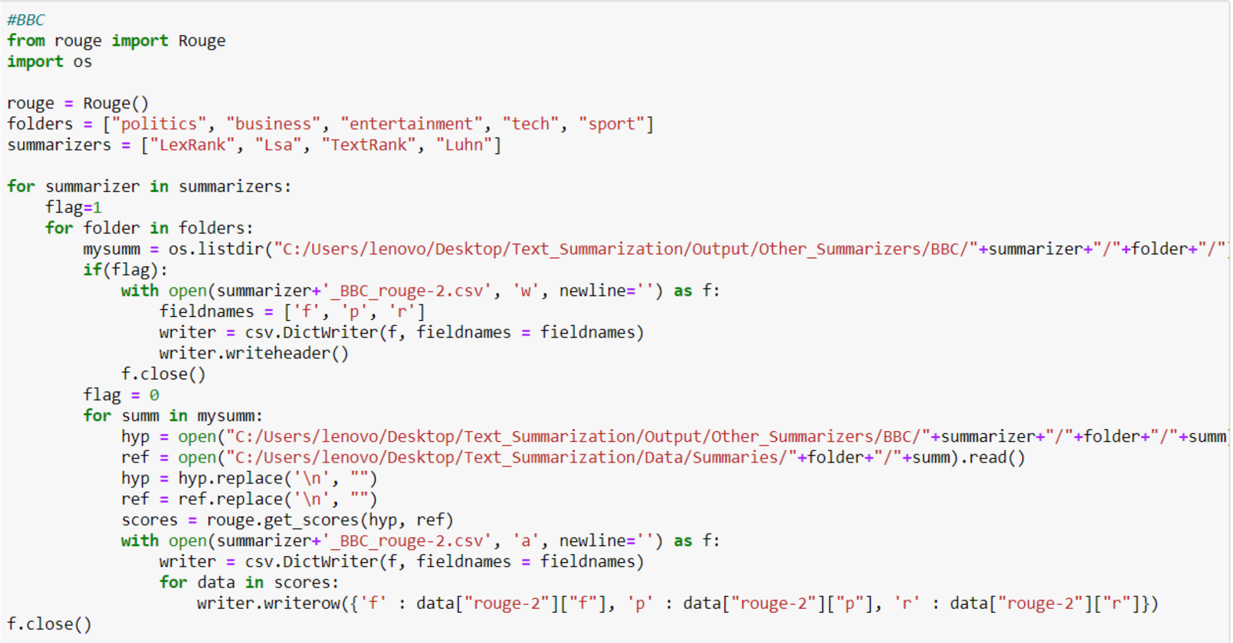


```r
  res_CC_len = res_CC_len + len
  #res_DC_len

  summ_CC = paste(summ_CC, Demo_data$Text[sent_id])
  #print(summ_DC)

  i=i+1
}
#EC
i=1
summ_EC = " "
res_EC_len = 1
while(res_EC_len <= summary_len)
{
  sent_id = EC$ranks[i]
  #print(sent_id)

  split = as_tibble(strsplit(Demo_data$Text[sent_id], " ")[[1]])
  len = dim(split)[[1]]
  res_EC_len = res_EC_len + len
  #res_DC_len

  summ_EC = paste(summ_EC, Demo_data$Text[sent_id])
  #print(summ_DC)

  i=i+1
}

# Write in Files
write.table(summ_EC,paste0("EC_", n),sep=". ",row.names=FALSE, col.names = FALSE)
```

Python code snippets:



```python
#BBC
from rouge import Rouge
import os

rouge = Rouge()
folders = ["politics", "business", "entertainment", "tech", "sport"]
with open('BBC_rouge-l.csv', 'w', newline='') as f:
    fieldnames = ['f', 'p', 'r']
    writer = csv.DictWriter(f, fieldnames = fieldnames)
    writer.writeheader()
f.close()

for folder in folders:
    mysumm = os.listdir("C:/Users/lenovo/Desktop/Text_Summarization/Data/Summaries/"+folder+"/")

    for summ in mysumm:
        hyp = open("C:/Users/lenovo/Desktop/Text_Summarization/Output/BBC/"+folder+"/EC_"+summ).read()
        ref = open("C:/Users/lenovo/Desktop/Text_Summarization/Data/Summaries/"+folder+"/"+summ).read()
        hyp = hyp.replace('\n', "")
        ref = ref.replace('\n', "")
        scores = rouge.get_scores(hyp, ref)
        with open('BBC_rouge-l.csv', 'a', newline='') as f:
            writer = csv.DictWriter(f, fieldnames = fieldnames)
            for data in scores:
                writer.writerow({'f' : data["rouge-l"]["f"], 'p' : data["rouge-l"]["p"], 'r' : data["rouge-l"]["r"]})
f.close()
```



```
#BBC
from rouge import Rouge
import os

rouge = Rouge()
folders = ["politics", "business", "entertainment", "tech", "sport"]
summarizers = ["LexRank", "Lsa", "TextRank", "Luhn"]

for summarizer in summarizers:
    flag=1
    for folder in folders:
        mysumm = os.listdir("C:/Users/lenovo/Desktop/Text_Summarization/Output/Other_Summarizers/BBC/"+summarizer+"/"+folder+"/"
        if(flag):
            with open(summarizer+'_BBC_rouge-2.csv', 'w', newline='') as f:
                fieldnames = ['f', 'p', 'r']
                writer = csv.DictWriter(f, fieldnames = fieldnames)
                writer.writeheader()
            f.close()
        flag = 0
        for summ in mysumm:
            hyp = open("C:/Users/lenovo/Desktop/Text_Summarization/Output/Other_Summarizers/BBC/"+summarizer+"/"+folder+"/"+summ
            ref = open("C:/Users/lenovo/Desktop/Text_Summarization/Data/Summaries/"+folder+"/"+summ).read()
            hyp = hyp.replace('\n', "")
            ref = ref.replace('\n', "")
            scores = rouge.get_scores(hyp, ref)
            with open(summarizer+'_BBC_rouge-2.csv', 'a', newline='') as f:
                writer = csv.DictWriter(f, fieldnames = fieldnames)
                for data in scores:
                    writer.writerow({'f' : data["rouge-2"]["f"], 'p' : data["rouge-2"]["p"], 'r' : data["rouge-2"]["r"]})
f.close()
```

References

1. Extractive and Abstractive Text Summarization Techniques.: Regular Issue, 9(1), 1040–1044 (2020). doi:10.35940/ijrte.a2235.059120
2. Debnath, A., Pinnaparaju, N., Shrivastava, M., Varma, V., Augenstein. I.: Semantic Textual Similarity of Sentences with Emojis. Companion Proceedings of the Web Conference 2020 (WWW'20). Association for Computing Machinery, New York, NY, USA, 426–430 (2020). doi:https://doi.org/10.1145/3366424.3383758
3. Mihalcea, R., Rada, Tarau P.: TextRank: Bringing Order into Texts (2004)
4. Erkan, G., Radev, D.R.: LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research, 22,* 457-479 (2004). doi:10.1613/jair.1523
5. Brin, S., Page, L.: The PageRank Citation Ranking: Bringing Order to the Web, Google Search Engine (1998)
6. Patil, K., Brazdil, P.: SUMGRAPH: Text summarization using centrality in the pathfinder network. In: International Journal on Computer Science and Information Systems (2007)
7. Erkan, G., Radev, D.R.: LexPageRank: Prestige in Multi-Document Text Summarization. Conference on Empirical Methods in Natural Language Processing, vol. 2, pp. 365-371 (2004).
8. Mutlu, B., Sezer, E.A., Akcayol, M. A.: Multi-document extractive text summarization: A comparative assessment on features. Knowledge-Based Systems, vol. 183 (2019).
9. Goldstein, J., Mittal, V., Carbonell, J., & Kantrowitz, M.: Multi-document summarization by sentence extraction. NAACL-ANLP 2000 Workshop on Automatic Summarization (2000). doi:10.3115/1117575.1117580
10. Reeve, L.H., Han, H., Brooks, A.D.: The use of domain-specific concepts in biomedical text summarization. Information Processing & Management, 43(6), 1765–1776 (2007). doi:10.1016/j.ipm.2007.01.026
11. Sarker, A.: Extractive summarization of medical documents using domain knowledge and corpus statistics. Australasian Medical Journal, 5(9), 478–481 (2012). doi:10.4066/amj.2012.1361
12. Bounhas, M., Elayeb, B.: Analogy-based Matching Model for Domain-specific Information Retrieval. Proceedings of the 11th International Conference on Agents and Artificial Intelligence (2019). doi:10.5220/0007342104960505
13. Carbinell, J., Goldstein, J.: The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. ACM SIGIR Forum, 51(2), 209–210 (2017). doi:10.1145/3130348.3130369
14. El-Haj, M. O., Hammo, B. H.: Evaluation of Query-Based Arabic Text Summarization System. 2008 International Conference on Natural Language Processing and Knowledge Engineering (2008). doi:10.1109/nlpke.2008.4906790
15. Afsharizadeh, M., Ebrahimpour-Komleh, H., Bagheri, A.: Query-oriented text summarization using sentence extraction technique. 2018 4th International Conference on Web Research (ICWR) (2018). doi:10.1109/icwr.2018.8387248
16. Mohamed, A., Rajasekaran, S.: Improving Query-Based Summarization Using Document Graphs. 2006 IEEE International Symposium on Signal Processing and Information Technology (2006). doi:10.1109/isspit.2006.270835

17. Meena, Y.K., Gopalani, D.: Evolutionary Algorithms for Extractive Automatic Text Summarization. Procedia Computer Science, 48, 244–249 (2015). doi:10.1016/j.procs.2015.04.177
18. Yerimbetova, A.S., Batura, T.V., Murzin, F.A., Sagnayeva, S.K.,: Automatic text summarization based on syntactic links (2020).
19. Hahn, U., Mani, I.: The challenges of automatic summarization. *Computer*, *33*(11), 29-36 (2000).
20. Sidorov, G., Gelbukh, A., Gómez-Adorno, H., Pinto, D.: Soft Similarity and Soft Cosine Measure: Similarity of Features in Vector Space Model. Computación y Sistemas, 18(3) (2014). doi:10.13053/cys-18-3-2043
21. Oldham, S., Fulcher, B., Parkes, L., Arnatkevičiūtė, A., Suo, C., Fornito, A.: Consistency and differences between centrality measures across distinct classes of networks. PLOS ONE, 14(7), e0220061 (2019). doi:10.1371/journal.pone.0220061
22. Li, B., Han, L.: Distance Weighted Cosine Similarity Measure for Text Classification. Lecture Notes in Computer Science, 611–618 (2013). doi:10.1007/978-3-642-41278-3_74
23. Shivakumar, K., Soumya, R.: Text summarization using clustering technique and SVM technique. International Journal of Applied Engineering Research, 10(12), 28873-81 (2015).
24. Boguraev, B., Kennedy, C.: Salience-based Content Characterisation of Text Documents (2002).
25. Steinberger, J., Jezek, K., Karel: Using latent semantic analysis in text summarization and summary evaluation. Proceedings of ISIM'04, pp. 93-100 (2004).

List of Publications

1. 4th International Conference on Electronics, Communication and Aerospace Technology [ICECA 2020]
   5-7, November 2020
   The Hotel Arcadia, 4, Avinashi Road, Goldwins, Coimbatore - 641 014.

2. 3rd International Conference on Computational Intelligence, Security & Internet of Things (ICCISIoT), 2020
   29-30 December 2020
   National Institute of Technology Agartala, Tripura, India.