Integrating Importance, Non-Redundancy and Coherence in Graph-Based Extractive Summarization

BY

N. JAI SAI GOUTHAM

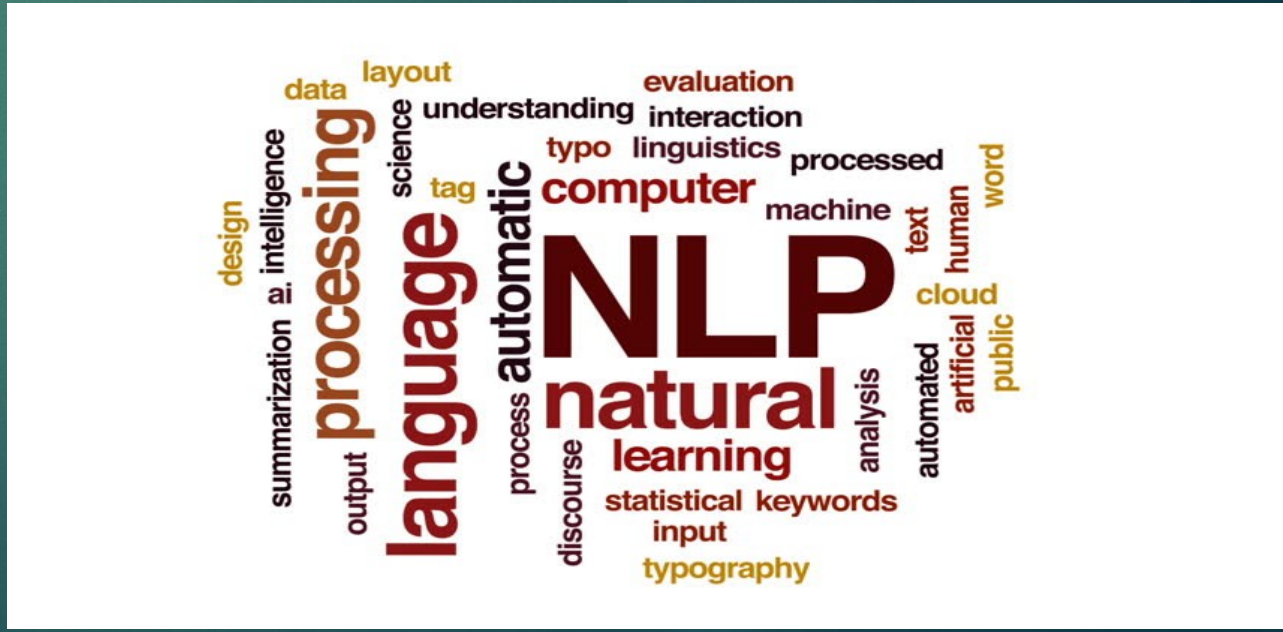
206118015

GUIDED BY

Dr. S.JAYA NIRMALA

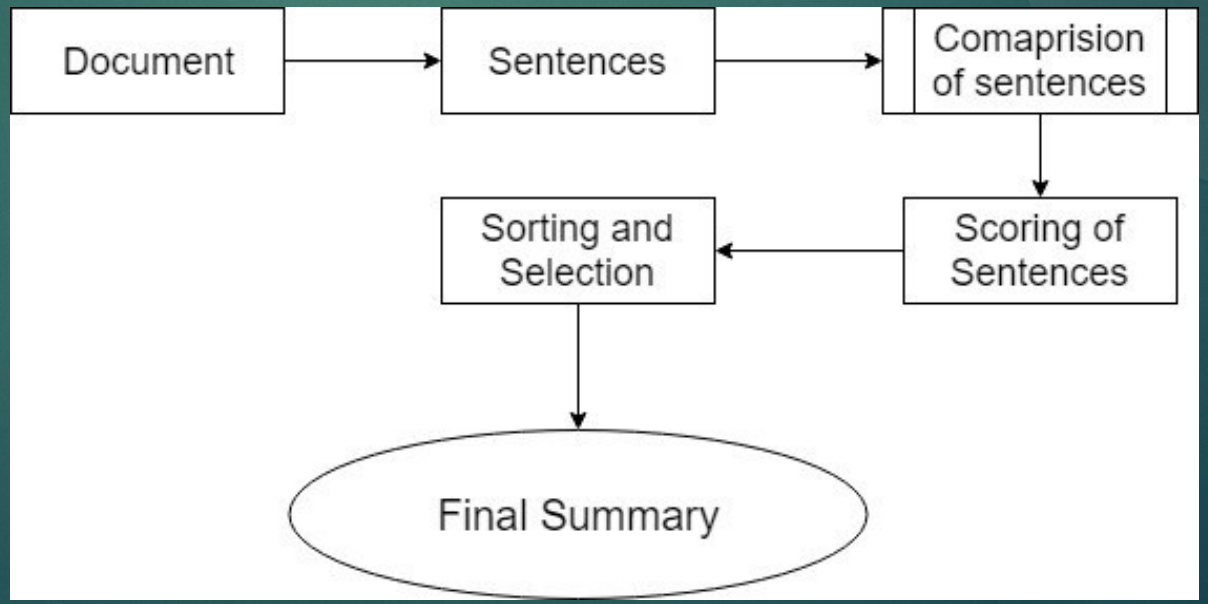
About NLP

- **Natural Language Processing** (**NLP**) is broadly defined as the automatic manipulation of natural language, like speech and text, by software.
- Natural Language Processing frequently involve speech recognition, natural language understanding, and natural language generation.



Extractive Summarization

- These methods rely on extracting several parts, such as phrases and sentences, from a piece of text and stack them together to create a summary.



Problem Statement

Graph-based summarization that attempts to select and organize the sentences in a summary with respect to the input documents.

Literature survey

- Paper 1: **Integrating Importance, Non-Redundancy and Coherence in Graph-Based Extractive Summarization** by Daraksha Parveen and Michael Strube.

- Paper 2: **Topical Coherence for Graph-based Extractive Summarization by** Daraksha Parveen, Hans-Martin Ramsl and Michael Strube.

Continued

- Paper 3,4 : LexRank: Graph-based lexical centrality as salience in text summarization, TextRank: Bringing order into texts.

- In this papers the data is extracted to form a graph merged by relations and rank sentences by algorithms which are used to generate summaries.

Approach

- Our paper approach is to summarize the given dataset by following steps :
  - Find the number of characters in the file.
  - Then By using HITS algorithm to rank the weights that results

    as document-term matrix normalized.
  - By using the results of the HITS algorithm we find vertices and edges and form a graph.

Continued..

- From the graph we will rank every sentence by using Text Rank algorithm.
- Compute the ROGUE scores according to the sentences in the file.
- Finally finding the important sentences and generate the summary of the file.

HITS Algorithm :

- **HITS *(hyperlink-induced topic search)***
- **HITS** algorithm make use of the link structure of the Weight graph in order to decide the relevance of the pages.
- **HITS** ranks the seed nodes according to their authority and hub weights.

Text Rank :

- The basic idea of Text-Rank is to provide a score for each sentence in a text
- Then you can take the top-n sentences and sort them as they appear in the text to build an automatic summary.

ROGUE Scores:

- ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation.

- It works by comparing an automatically produced summary or translation against a set of reference summaries.

Continued..

- ROUGE has different types :
  - ROUGE 1
  - ROUGE 2
  - ROUGE SU4

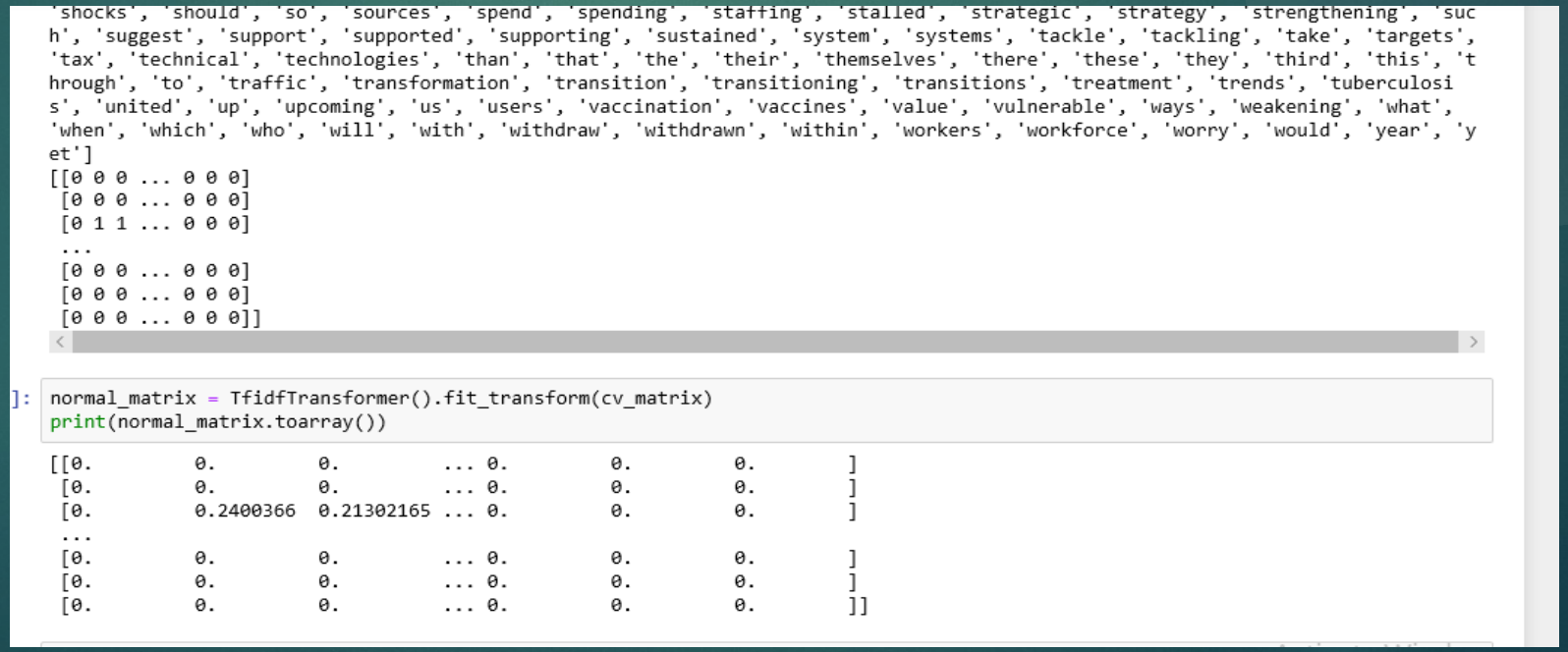
Experiment and Discussion :

- By giving input file from the PLOS data set
- And calculate its length, size of Bytes.

```
In [8]: document = readDoc()
        print('The length of the file is:', end=' ')
        print(len(document))

        Please input a file name: story3.txt
        You have asked for the document story3.txt
        1
        <class 'str'>
        The length of the file is: 4690

In [9]: sentences_list = tokenize(document)
        print('The size of the list in Bytes is: {}'.format(sys.getsizeof(sentences_list)))
        print('The size of the item 0 in Bytes is: {}'.format(sys.getsizeof(sentences_list[0])))

        The size of the list in Bytes is: 344
        The size of the item 0 in Bytes is: 367
```

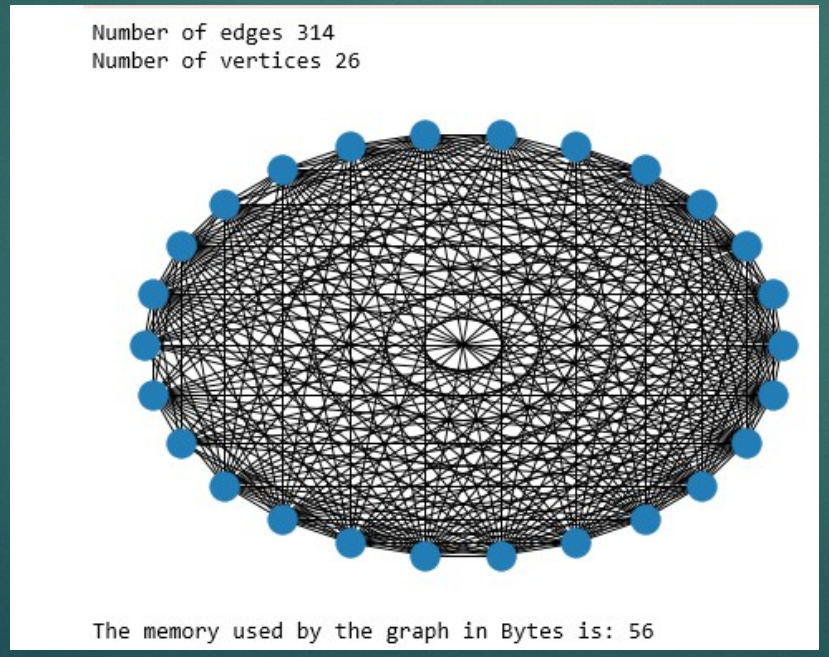
Continued..

- Then HITS Algorithm is intiated and generate the term-document matrix (TD matrix) of the data.



Continued..

- By using the matrix data the graph is generated as shown.



Continued..

- Getting the rank of every sentence using Text-Rank algorithm.

```
<class 'dict'>
The size used by the dictionary in Bytes is: 1184
0 0.04728990582222432
1 0.040466764172989646
2 0.04599944459494297
3 0.039071455797667055
4 0.039326122602634614
5 0.02920471127281726
6 0.04103896190302626
7 0.038565601552342746
8 0.03148377820435344
9 0.03353131330047237
10 0.029124140321561548
11 0.0424915586060884
12 0.04261473878269958
13 0.028523358345298483
14 0.03720796493442315
15 0.02841036442856509
16 0.03878540080844102
17 0.036662377418951446
18 0.04291917904881489
19 0.044770313846590123
20 0.037570582108774474
21 0.03515053658631601
22 0.04135551991984642
23 0.03887380942366619
24 0.04358314978567249
```

Continued..

- By using normalized values in text rank , it finds the important sentences and generate summaries.

Continued..

- Rogue scores are calculated for system summaries and its corresponding reference summaries.

```
Evaluate ROUGE based on sentecnce lists
ROUGE-N(1-2) & SU4 recall only with confidence interval
{'ROUGE-1': 0.24495,
 'ROUGE-1-cf95': (0.21212, 0.27778),
 'ROUGE-2': 0.13832,
 'ROUGE-2-cf95': (0.06452, 0.21212),
 'ROUGE-SU4': 0.13579,
 'ROUGE-SU4-cf95': (0.06325, 0.20833)}
```

Results :

- By HITS Algorithm, Integrating and and Non-Redundancy are shown by implementing graph.

- By text rank we rank the minimum and maximum sentences and generate summary.

Continued..

- Coherence
- By rogue scores , the difference between the summaries are compared.

Conclusion:

- This approach is based on the different algorithms to evaluate the significance of each concept.
- The sentences containing more concepts and highest significance scores are chosen in the summary as long as they are not the same sentences.
- Integrated it closely with determining importance and avoiding redundancy in an unsupervised graph-based method for extractive single-document summarization.

References

- Integrating importance, non-redundancy and coherence in graph-based extractive summarization. In Proceedings of the 24th International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25– 31 July 2015, pages 1298–1304.

- Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In Proceedings of the Text Summarization Branches Out Workshop at ACL ’04, Barcelona, Spain, 25–26 July 2004, pages 74–81, 2004.

Continued..

- Rada Mihalcea and Paul Tarau. TextRank: Bringing order into texts. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, 25–26 July 2004, pages 404–411, 2004.