OPINION SUMMARIZATION

Report submitted in fulfillment of the requirements

for the Exploratory Project of

Second Year B. Tech

by

Pacha Venkata Sri Harsha

Under the guidance of

Dr. Ravindranath Chowdary

Department of Computer Science and Engineering
INDIAN INSTITUTE OF TECHNOLOGY (BHU) – VARANASI
Varanasi 221005, Uttar Pradesh, India
May 2022

Declaration

I declare that,

- The work contained in this project is original and has been done by myself and under the general supervision of my supervisor.

- The work has not been submitted for any project.

- Whenever I have used materials (data, theoretical analysis, results) from other
  sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.

- Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

Place: IIT (BHU) Varanasi

Date: 02-05-2022

Pacha Venkata Sri Harsha,

B. Tech,

Department of Computer Science and Engineering,

Indian Institute of Technology (BHU) Varanasi,

VARANASI, India – 220005.

Certificate

This is to certify that the work contained in this report entitled "**OPINION SUMMARISATION**" is being submitted by **Pacha Venkata Sri Harsha (20075063)** carried out in the Department of Computer Science and Engineering, Indian Institute of Technology (BHU) Varanasi, is Bonafede of our supervision.

Place: IIT (BHU) Varanasi
Date: 02-05-2022

Dr. Ravindranath Chowdary,
Department of Computer Science and Engineering,
Indian Institute of Technology (BHU) Varanasi,
Varanasi, INDIA 221005.

Acknowledgments

It is a great pleasure for us to express respect and a deep sense of gratitude to my supervisor Dr. Ravindranath Chowdary C, Assistant Prof., Department of Computer Science and Engineering, Indian Institute of Technology (BHU) Varanasi, for his wisdom, vision, expertise, guidance, enthusiastic involvement and persistent encouragement during the planning and development of this work.

I also gratefully acknowledge his painstaking efforts in thoroughly going through and improving the manuscripts without which this work could not have been completed. We are also highly obliged to Prof. Pramod Kumar Jain, Director, Indian Institute of Technology (BHU) Varanasi, and Prof. Sanjay Kumar Singh, Head of Department, CSE for providing all the facilities, help and encouragement for carrying out this exploratory project work. We are also obliged to our parents for their moral support, love, encouragement and blessings to complete this task. Finally, we are indebted and grateful to the Almighty for helping us in this endeavour.

Place: IIT (BHU) Varanasi

Date: 3 May 2022

Pacha Venkata Sri Harsha

Contents

Introduction

The opinion summarization, in which the reviews given by several users, describe a product in a paragraph, was interpreted by the machine to understand how the customer is reacting, like if he was actually loving it or disliking it, and then return a short summary about the product in one line. This program was adopted by many websites so that users who want to see the reviews can look according to their view, whether they want positive or negative reviews.

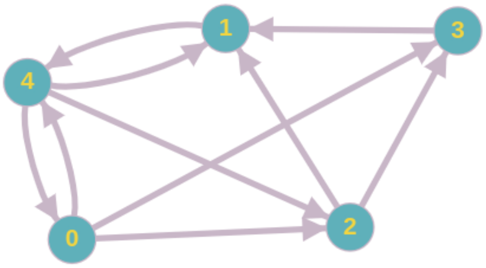
There are many methods to implement this program. Basically, its methods are classified into Extractive and Abstractive methods. Usually, extractive models are unsupervised where we don't need user input. It doesn't need machine learning algorithms. It uses a ranking algorithm, which had a need for graphs. For Example, HITS, LexRank and PageRank. Another one is the abstractive type method. It requires machine learning and it is basically supervised (user input is required for training the model). Here, neural networks are used. In the case of extractive methods, dangling anaphora is not a big issue, but in the case of abstractive methods, dangling anaphora is considered.

In this project, we are using the **Text Rank** method.

Working Algorithm - Text Rank

In the text rank method, we are going to rank sentences and sort them in ascending order to print the first 5 sentences as the summary. First of all the program takes the input and then will concatenate all the sentences. Further text will be split into individual sentences. In the next step, we will find the vector representation for each and every sentence. Then similarities between the sentence vectors and then will be stored in a square matrix. Then the similarity matrix graph is converted into a graph with sentences as vertices and the value of the similarity score as edges, for sentence rank calculation.

As we are told that the sentences or pages are ranked on the basis of final scores which were computed between the sentences, let us say some link between two pages and each sentence having is a score. Let us take an example:



The graph can also be written in the form of how much input it has got, i.e in the linear form.

The **Markov-Chain** theorem states that the sentence which is to be selected next is dependent on the probability distribution, in that it will be updated in each step.

The linear representation of the graph is as follows:

$$\left\{ \begin{array}{c} r_0 = \frac{r_4}{3} \\ r_1 = \frac{r_2}{2} + \frac{r_4}{3} + r_3 \\ r_2 = \frac{r_0}{3} + \frac{r_4}{3} \\ r_3 = \frac{r_0}{3} + \frac{r_2}{2} \\ r_4 = \frac{r_0}{3} + r_1 \end{array} \right\}$$

The scores r_j are given by the formula:

$$r_{j} = \sum_{i \to j} \frac{r_i}{d_i}$$

The same graph when written in the form of matrix:

$$\mathrm{P} = \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{1}{3} \\ 0 & 0 & \frac{1}{2} & 1 & \frac{1}{3} \\ \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} \\ \frac{1}{3} & 0 & \frac{1}{2} & 0 & 0 \\ \frac{1}{3} & 1 & 0 & 0 & 0 \end{bmatrix}$$

Let π be the initial probability distribution which is equal to,

$$\pi = \left(\frac{1}{n}, \frac{1}{n}, \frac{1}{n} \dots \dots, \frac{1}{n}\right)$$

To calculate the ranking of whole sentences, it should jump from one sentence to another sentence by seeing its probability to the next sentence. Hence for that, the probability distribution is calculated as

$$\pi^{(t+1)} = P\pi^{t},$$

Where P is the matrix representation of the Graph.

But we had got a flaw from this theorem. What if we get a **stationary distribution**, i.e, the probability distribution is not changed anytime ($\pi = P\pi$)? We can see that π is an eigenvector of the matrix P having the eigenvalue 1. Instead of calculating the eigenvectors of P, we will choose which one have the eigenvalue, and then we use **Frobenius-Perron theorem**.

Frobenius-Perron theorem states that, if a matrix **A** is a square matrix and all the values in it are positive, then it has a positive eigenvalue **r**, such that eigenvalue A is less than **r**. In this case, our matrix P is positive and square, we can freely use this theorem. We can deduce that the dominant eigenvector of P with dominant eigenvalue 1. To calculate the value of π, we are going to use the power method through iteration, with dominant eigenvalue b was initialised randomly, until the distribution gets converged.

The pseudocode for Power Method:

```
powerMethod():
    while not converge do:
        b = A*b
        norm = compute_norm(b)
        b /= norm
    end while
```

But we should get to know about another worst situation, where a web page 1 is linked to web page 2 and web page 2 refers web page 1. This type of situation is called **Spider Trap Problem**. We can also find where a page is been referred but it doesn't refer any other page. This type of situation is called **Dead End**.

To overcome this problem, we had introduced the method of **teleportation**.

Teleportation consists a graph connecting each node of the graph to all other nodes. With a certain probability β, we will be jumping from one sentence to another sentence according to our transitionmatrix P, we will have a probability to do our action of (1-β)/n. Now we will get a new transistion matrix R,

$$R = \beta P + (1-\beta)ve^T,$$

Where v is a vector of ones and e is a vector of 1/n.

$$e^T = \left(\frac{1}{n}, \frac{1}{n}, \frac{1}{n} \dots \dots, \frac{1}{n}\right) \quad \text{and} \quad \mathrm{v} = (1, 1, \dots 1)^{\mathrm{T}}.$$

Here β is referred as the damping factor. Usually, its value is set to 0.85. The new transistion matrix will be as

$$\mathrm{R} = \begin{pmatrix} \frac{1-\beta}{5} & \frac{1-\beta}{5} & \frac{1-\beta}{5} & \frac{1-\beta}{5} & \frac{1}{3}\beta + \frac{1-\beta}{5} \\ \frac{1-\beta}{5} & \frac{1-\beta}{5} & \frac{1}{2}\beta + \frac{1-\beta}{5} & \beta + \frac{1-\beta}{5} & \frac{1}{3}\beta + \frac{1-\beta}{5} \\ \frac{1}{3}\beta + \frac{1-\beta}{5} & \frac{1-\beta}{5} & \frac{1-\beta}{5} & \frac{1-\beta}{5} & \frac{1}{3}\beta + \frac{1-\beta}{5} \\ \frac{1}{3}\beta + \frac{1-\beta}{5} & \frac{1-\beta}{5} & \frac{1}{2}\beta + \frac{1-\beta}{5} & \frac{1-\beta}{5} & \frac{1-\beta}{5} \\ \frac{1}{3}\beta + \frac{1-\beta}{5} & \beta + \frac{1-\beta}{5} & \frac{1-\beta}{5} & \frac{1-\beta}{5} & \frac{1-\beta}{5} \end{pmatrix}$$

Hence, we had a perfect working algorithm, and now it is ready to find the cosine similarity between two sentences to deduce the final scores. The cosine similarity value increases when the angle subtended between two sentences is low (which means they are close).

Conclusion

- As we told earlier, there are two types of summarizations, namely Abstractive and Extractive.
- This method is light when compared to the abstractive one.
- Its even time saving when compared to machine learning algorithm.

This algorithm actually does not generate the summary, it simply ranks the sentences on the basis of its score at the end and then the summary is printed out. In a straight word, this algorithm is not as accurate as abstractive one. In the final summary, each sentence may not be linked with other one. But it has its own pros like ranking the documents and selecting the most informative document. One thing we can observe is that we get the same result when you input from several documents and when you merge all the document in one. Each method has its own pros and cons.

References

- B. Balcerzak, W. Jaworski and A. Wierzbicki, "Application of TextRank Algorithm for Credibility Assessment," 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2014, pp. 451-454, doi: 10.1109/WI-IAT.2014.70.
- Solanki, Surabhi & Verma, Seema & Chahar, Kishore. (2022). A Comprehensive Study of Page-Rank Algorithm. 10.1007/978-981-16-6616-2_1.
- Parimoo, Rohit & Sharma, Rohit & Gaur, Naleen & Jain, Nimish & Bansal, Sweeta. (2022). Applying Text Rank to Build an Automatic Text Summarization Web Application. International Journal for Research in Applied Science and Engineering Technology. 10. 865-867. 10.22214/ijraset.2022.40766.
- **An Introduction to Text Summarization using the TextRank Algorithm**
- **Understanding Automatic Text Summarization-1: Extractive Methods**
- **Text summarization using TextRank in NLP**
- **Guide to NLP's Textrank Algorithm**