https://doi.org/10.31871/IJNTR.7.5.9
International Journal of New Technology and Research (IJNTR)
ISSN: 2454-4116, Volume-7, Issue-5, May 2021 Pages 34-38

LexRank Algorithm: Application in Emails and Comparative Analysis

Aviva Munshi, Anoushka Mehra, Ashna Choudhury

***Abstract*— Text summarization can be described as the process that helps to shorten long pieces of text, with the goal of producing succinct and factual content that places specific focus on the basics present in the document. This is a known issue in machine learning and natural language processing, and the amount of attention given to it has only increased over many years, keeping in mind that there are copious quantities of data online. It also has the ability to collect useful information that can be managed fairly easily by humans and could be used for a wide range of purposes, such as text assessment. In this paper, we are attempting to present an automated text summary method that relies on LexRank Algorithm to find the most significant and appropriate statements in the long input text and make them a part of the short summary. In this project, given a set of data for a particular topic, the appropriate summary is produced using the LexRank algorithm. It is also capable of summarizing a single data as well. We are using college circulars as the data for testing the relevance of the produced summaries. We are also testing its relevance by testing the already available data by generating the ROUGE scores where the automatically generated summaries are compared with the manually written summaries of the same.**

***Index Terms*— email summarization, LexRank Algorithm, natural language processing, rogue scores**

I. INTRODUCTION

Text Summarization is a hugely impactful and helpful tool in Natural Language Processing,NLP. In the fast paced world of today, it is unreasonable to go through millions of resources to decide whether any of them are important.. In order to remove this problem, this paper seeks to employ an innovative approach to summarize substantial quantities of emails received by an individual.

We aim to build an email summarizer that helps to automatically summarize email threads, by presenting a paragraph consisting of *the most important sentences*. We employ the LexRank algorithm for this summarization - an extraction based summarization technique. LexRank is based on the concept of eigenvector centrality in graph representation of sentences. In this model, we have a connectivity matrix based on intra-sentence cosine similarity which is used as the adjacency matrix of the graph representation of sentences.

This sentence extraction majorly revolves around the set of sentences with the same intent i.e. a centroid sentence is selected which works as the mean for all other sentences in the document. The sentences are then ranked according to their similarities.

Aviva Munshi, Computer Science Student, Vellore Institute of Technology, Vellore, India
Anoushka Mehra, Computer Science Student, Vellore Institute of Technology, Vellore, India
Ashna Choudhury, Computer Science Engineering, Vellore Institute of Technology, Vellore, India

This algorithm helps us get an optimum solution by maintaining redundancy and improving coherency.

II. METHODOLOGY

The dataset whose summaries need to be generated are saved in the folder named Documents. Under this folder, the topic segregation is further performed. All the articles pertaining to a particular topic are saved under a single folder. This helps the LexRank understand that a collaborative summary needs to be generated. The LexRank algorithm reads the folders one by one and the generated result is saved under the same folder name in which the articles were written. To perform the data fetch, operating system OS walk function is used. Once the data to be summarized is fetched, the first step is the data pre-processing in which the unwanted tags and metadata are removed. Essentially, the data is cleaned and tokenized. This step is known as *email pre-processing.*

Pre-processing includes identifying rhetorical roles, initiating stop words and punctuations. The words are tokenized and given as inputs to the algorithm for the process of sentence scoring. A threshold value is stated for the process of sentence extraction for the summarization process to initiate. The fetched summary is then compared with its parent article to generate an index score.

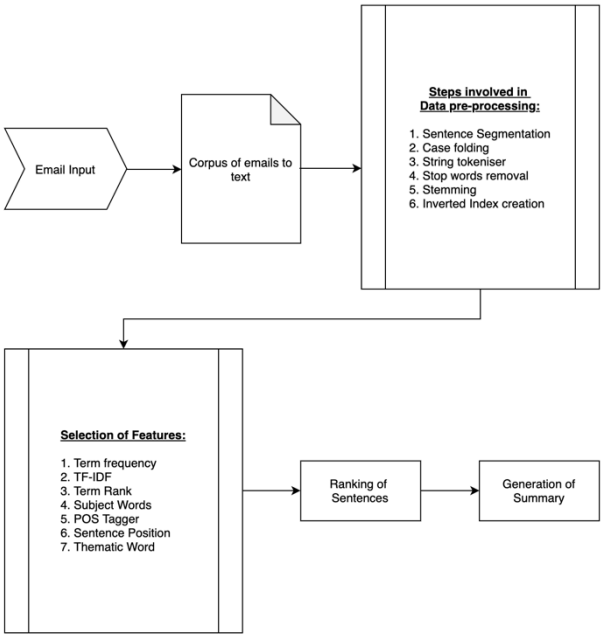


Fig. 1: Architectural Diagram

In the four processing steps, we conduct email cleaning, which consists of non-text filtering, normalization of chapters, normalization of sentences, and normalization of words with an email message as the input.

Consequently, we recognise the current header, signature, quote, and program code in the email in non-text filtering and delete the blocks found. Afterwards, we find extra line breaks in paragraph normalisation and delete them. Our next step is to find out whether a period, a question mark, or an exclamation mark is a true sentence-end in sentence normalisation. If so, we take that as the limit of a sentence. In addition, we also remove non-words, like non-ASCII words, tokens with several special symbols and long tokens. Case restoration is performed on badly cased words in word normalisation. In this process, various python libraries like BeautifulSoup are used to perform the pre-processing task. The terminal steps in the summary generation are the LexRank scores generation and sentence ranking.

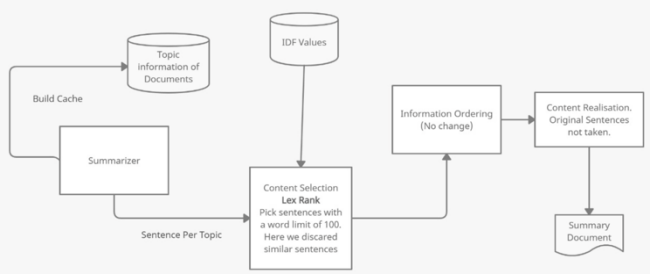


Fig. 2: Flow Diagram

A. *LexRank Algorithm*

i. Matrix Generation

The bag of words model is used to describe N-dimensional vectors in order to define similarity, where N is the number of all possible words in words in a specific language. Its value is increased for each work that occurs in a sentence.

ii. Cosine similarity computation

This is the advanced stage that differentiates the algorithm of LexRank from the algorithm of the original TextRank. The matrix developed in the previous step is here adjusted to the matrix of cosine similarity. The value of the corresponding dimension in the vector representation of the sentence for each term that appears in a sentence is the number of occurrences of the word in the sentence times that of the word inverse document frequency idf.

$$\boldsymbol{idf - cosine}\ (x,y) = \frac{\sum_{w \in x,y} tf_{w,x} tf_{w,y} (idf_w)^2}{\sqrt{\sum_{(x_i \in x)} (tf_{x_i,x} idf_{x_i})^2}\ \sqrt{\sum_{(y_i \in y)} (tf_{y_i,y} idf_{y_i})^2}}$$

where $tf_{w,s}$ is the number of occurrences of the word w in the sentence s and sentence ID dXsY indicates the Yth sentence in the Xth document. The idf is calculated as follows:

$$idf_i = log(N/n_i)$$

where N is the total number of documents in a set, and n_i is the number of documents that contain the word i. This equation calculates the separation between two sentences x and y. The less redundant their relation becomes, the more comparable two sentences are, i.e. they have a certain degree of similarity between them and can be used in the process of summarization.

B. *LexRank Scores*

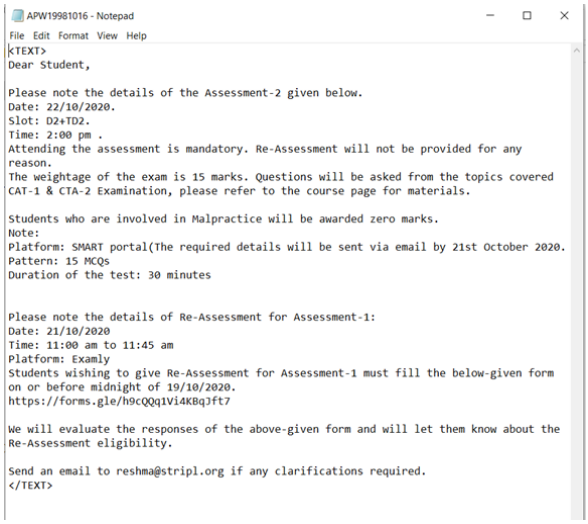
In this step, the LexRank algorithm takes the processed and cleaned data as an input and performs the steps mentioned in the pseudocode to obtain the sentence ranking. Firstly, the matrix is generated. This generated matrix is used to provide the cosine similarity values known as the IDF values to each word is generated which is used to generate the LexRank scores based on the centroid values and the power method. All the sentences are ranked accordingly and depending on the number of sentences required in the summary set by the user, the top sentences are retrieved to form the summary. If the values are greater than the value of a threshold idf, the element is replaced by value 1. If it is lower than the appropriate threshold, the value is replaced by 0. In these steps, we thus generate a standard term frequency-inverse document frequency tf-idf table or matrix. Each value of the cosine matrix is split by each nodes degree. The corresponding degree of each node is the degree of centrality here. To choose similar word matrices, LexRank collects sentences with the highest inverse document frequencyidf values to form the most accurate description of any corpus. The summaries are then stored in the Lexrank_result folder using the document articles folder name as the file name.

C. *Power iteration method*

Basically, the Power Iteration or Power method is used to measure the value of a matrix 's largest eigen vector. The algorithm will generate a number lambda, which is the largest (in absolute value) eigenvalue of A, for a diagonalizable matrix A, and a nonzero vector v, which is a corresponding lambda eigenvector.

III. DATASET ANALYSIS

We have made use of the emails that we have received from college for the generation of the email summaries. For the generation of the email summaries, a wide range of emails were taken from the mails received such as information about the academic changes, test information, cultural activities and events and many more. All these mails were provided as the input for the email summarizer developed using the LexRank algorithm.


APW19981016 - Notepad

File Edit Format View Help

<TEXT>
Dear Student,

Please note the details of the Assessment-2 given below.
Date: 22/10/2020.
Slot: D2+TD2.
Time: 2:00 pm .
Attending the assessment is mandatory. Re-Assessment will not be provided for any reason.
The weightage of the exam is 15 marks. Questions will be asked from the topics covered CAT-1 & CTA-2 Examination, please refer to the course page for materials.

Students who are involved in Malpractice will be awarded zero marks.
Note:
Platform: SMART portal(The required details will be sent via email by 21st October 2020.
Pattern: 15 MCQs
Duration of the test: 30 minutes

Please note the details of Re-Assessment for Assessment-1:
Date: 21/10/2020
Time: 11:00 am to 11:45 am
Platform: Examly
Students wishing to give Re-Assessment for Assessment-1 must fill the below-given form on or before midnight of 19/10/2020.
https://forms.gle/h9cQQq1Vi4KBqJft7

We will evaluate the responses of the above-given form and will let them know about the Re-Assessment eligibility.

Send an email to reshma@stripl.org if any clarifications required.
</TEXT>

Fig. 3: Sample email data used for summarisation

For this purpose, the entire mail including the signatures, and salutations are considered which will be ignored by the LexRank summarizer. We have also considered emails which have trailing mails with the same subject. In such cases, a collaborated summary is provided to the user. This process not only reduces the time take to read the emails one by one but it also reduces the hassle of reading all the summaries of the emails one by one.

IV. RESULTS

A. LexRank Results

The email summarization was performed on various college emails that we have received. For the summarized document, we have set the summary length to 4. This helps us get all the important information with an adequate length. Making this value too small or too large might lead to losing sight of important data or getting unwanted information respectively. For the sample data, we have considered the recent mails received by the dean Academics regarding the modification of the FALL 20-21 calendar. We have received two mails and both of them have been saved under the same folder. The two mails saved are as follows:

Fig. 4: Email to be summarized

The generated output is saved in the Lexrank_result. The generated summary screenshot is as follows:

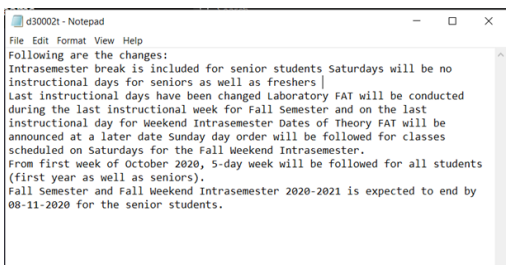


Fig. 5: Generated email summary as output

From the generated summary, we can observe that only the necessary and the most important information is generated covering all the aspects in just a very short paragraph. This is extremely helpful as we receive various emails and it is very difficult to read them all. We have also compared our LexRank algorithm with the maximum marginal relevance MMR algorithm which can also generate a summary based on multiple documents or articles.

B. MMR Algorithm Results

The summary task is modelled on the basis of MMRmethods in such a way that the contents of the summary generated should consist of the relevant query information and minimal similarity between the contents. This approach combines the coverage and significance in the description with question variables. For non-redundancy, the balance weight for relevancy is taken as 0.7and 0.3.
The general drawback of these strategies is that their success does not ensure that all coverage and non-redundancy elements are included in the overview. Therefore, our proposed method that is using LexRank Algorithm for text summarization is a better approach. The summaries for both were generated using the Document Understanding Conferences DUC database and the results were evaluated using the PyRouge scores in comparison with the human summaries. Let us consider one of the news articles topics provided by the DUC to observe the results.
The article whose output is provided below is the summary of the multiple articles provided on the Hurricane whose sample can be observed in Fig. 4. For the summary generation, the summary length is set to 6 as the news articles are comparatively longer and keeping the length 4 might lead to losing out on some important information.

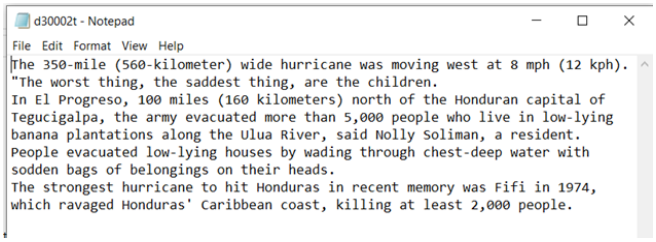


Fig. 6: MMR Summary

For the same input the generated LexRank summary can be observed below. We can see that the summary provided by LexRank is more detailed even though the summary length remains the same. The general drawback of the MMR strategies is that their success does not ensure that all coverage and non-redundancy elements are included in the overview. Therefore, our proposed method that is using LexRank Algorithm for text summarization is a better approach. This can also be observed by the ROUGE scores obtained.

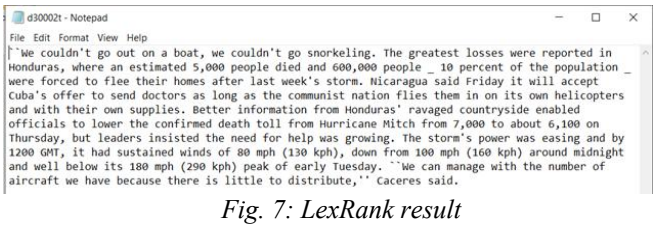


Fig. 7: LexRank result

C. PyRouge Scores

PyRouge is a Python wrapper for the evaluation package for ROUGE summarization. By automatically converting your summaries into a format ROUGE understands, and

automatically generating the ROUGE configuration file, PyRouge is designed to make it easier to get ROUGE scores. It is basically a set of metrics for measuring both automated text summarization and machine translation. It operates by comparing a summary or translation produced automatically against a collection of (typically human-produced) reference summaries. The three metrics on which the algorithms can be compared are recall, precision and the F Measure scores. The recall value is calculated by obtaining the ratio of the no. of words that are exactly matching with the human generated summary to the total number of words present in the human summary. This is a sentence level comparison.

$$\frac{\textit{number of overlapping words}}{\textit{total words in reference summary}}$$

The precision scores are calculated by obtaining the ratio of the number of words that are exactly matching with the human generated summary to the total number of words present in the system generated summary. This is also a sentence level calculation and its average is obtained finally for the comparison. F-measure score is nothing but the total average based on Recall and Precision value. For the effective comparison of the algorithms in our project, we will use the Recall values of the ROUGE measures generated as it is highly important to match the words and generate apt summaries for email irrespective of the generated sentence length. Our focus should be the similarity with the human summaries which means, the number of coinciding words and the average should be higher which is obtained by the Recall value. For a better comparison, we have considered three ROUGE score generation types namely, ROUGE-1
(this refers to overlap of unigrams between the system summary and reference summary), ROUGE-2 (this refers to the overlap of bigrams between the system and reference summaries) and ROUGE-SU* (this can also be called skip-gram co-occurrence). For example, skip-bigram measures the overlap of word pairs that can have a maximum of two gaps in between words.
The scores obtained for all the three ROUGE types are shown as below:

Table I: Scores obtained by ROUGE

System		LexRank (%)	MMR (%)
ROUGE-1	R%	38.672	33.837
	P%	23.042	31.504
ROUGE-2	R%	7.114	5.625
	P%	4.080	5.243
ROUGE-SU*	R%	14.860	10.402
	P%	5.174	9.035

From the above table, we can clearly see that the Recall values of LexRank for all the ROUGE types are higher indicating that it is a better algorithm to use for the generation of the email summaries. The screenshots of the outputs obtained are as follows:

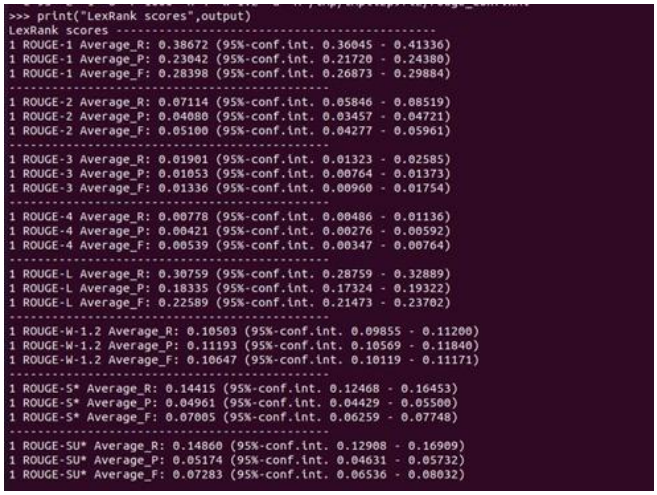


Fig.8a: LexRank Scores

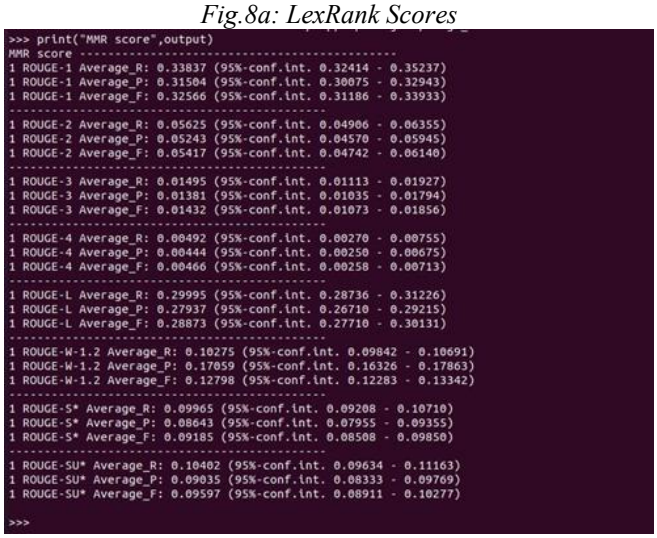


Fig. 8b: LexRank Scores

D. *Jaccard Similarity Index*

We have also generated the Jaccard Scores for the comparison of the two algorithms to check for the similarity between the generated summaries by the two approaches. For two sets, the Jaccard similarity index (sometimes referred to as the Jaccard similarity coefficient) compares members to see which members are shared and which are distinct. For the two data sets, it is a measure of similarity, with a scale from 0 percent to 100 percent. The greater the percentage, the more comparable the two populations are.

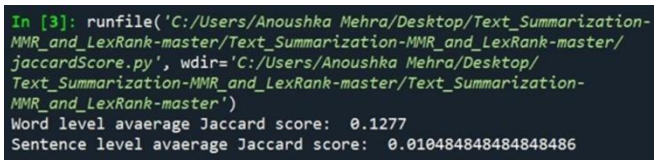


Fig. 9: Jaccard Scores

V. CONCLUSION

In this project, the LexRank algorithm for email summarization was performed on various emails. The summary length was fixed so as to make sure that no important point is missed out in the summarized text. Further, we have compared our approach, LexRank algorithm with MMR algorithm to show that our proposed method is a better alternative for text summarization. This is

done by observing the recall values of all three types of ROUGE in both algorithms.

VI. FUTURE WORK

Despite many years of research, there is still room for improvement in this field. Till date, the most efficient and flexible methods used in automated summarization are based on extractive methods. The collection of sentences from various documents contributes to redundancy in multidocument summarization, which in turn must be removed. Researchers are currently working on a fair representation of the text content in order to solve the duplication problem and, more interestingly, are now attempting to provide summaries customized to individual user needs.

The analysis of automatic summarization is still a challenging issue. A real issue is the lack of agreement between humans when analysing summaries. The development of more concentrated summaries can lead to a more consistent assessment and better convergence between human and automated methods of evaluation.

REFERENCES

[1] Giuseppe Carenini; Raymond Ng; Gabriel Murray, "Methods for mining and summarizing text conversations," in *Methods for Mining and Summarizing Text Conversations* , 1, Morgan & Claypool, 2011, pp.1-130.

[2] T. Ayodele, R. Khusainov and D. Ndzi, "Email classification and summarization: A machine learning approach," *2007 IET Conference on Wireless, Mobile and Sensor Networks*(CCWMSN07), Shanghai, 2007, pp. 805-808.

[3] A. El-Kilany and I. Saleh, "Unsupervised document summarization using clusters of dependency graph nodes," *2012 12th International Conference on Intelligent Systems Designand Applications (ISDA),* Kochi, 2012, pp. 557-561.

[4] G. Carenini, R. T. Ng, X. Zhou, "Summarizing email conversations with clue words", *WWW'07*, pp. 91- 100, 2007.

[5] Muresan S., Tzoukermann E. and Klavans J. L., "Combining linguistic and machine learning techniques for email summarization" in *ConLL '01: Proceedings of the 2001 workshop on Computational Natural Language Learning,Association for Computational Linguistics*, USA, pp.1-8, 2001.

[6] G.M.R.N. Jan Ulrich Giuseppe Carenini, "Regression based summarization of email conversations," in *3rd Int'l AAAI Conference on Weblogs and Social Media* (ICWSM-09). SanJose, CA: AAAI, 2009.

[7] David M. Zajica, Bonnie J. Dorra and Jimmy Linb, "Single-document and multi-document summarization techniques for email threads using sentence compression", *Information Processing and Management, Vol. 44*, No. 4, pp. 1600-1610, July 2008.

[8] Chuang, Wesley T., and Jihoon Yang. "Text summarization by sentence segment extraction using machine learning algorithms." in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 454-457. Springer, Berlin, Heidelberg, 2000.

[9] Takamura, Hiroya, and Manabu Okumura. "Text summarization model based on maximum coverage problem and its variant." in *Proceedings of the 12th Conference of the European Chapter of the ACL* (EACL 2009), pp. 781-789. 2009.

[10] Mihalcea, Rada. "Graph-based ranking algorithms for sentence extraction, applied to text summarization." in *Proceedings of the ACL interactive poster and demonstration sessions*, pp. 170-173. 2004.

Aviva Munshi
https://www.linkedin.com/in/aviva-munshi/
Anoushka Mehra
https://www.linkedin.com/in/anoushka-mehra-7388111ba/
Ashna Choudhury
https://www.linkedin.com/in/ashna-choudhury-9870b31b7/