International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 04 | Apr 2019 www.irjet.net p-ISSN: 2395-0072

Automatic Text Summarization using Text Rank

Tanwi[1], Satanik Ghosh[1], Viplav Kumar[1], Yashika S Jain[1], Mr. Avinash B[2]

[1]Eighth Semester, Dept. of ISE, The National Institute of Engineering, Mysore
[2]Asst.Professor. of ISE, The National Institute of Engineering, Mysore

---

Abstract - *In today's world the amount of data present online it's very difficult to summarize the entire content and refine it into suitable form of information. The abundance of unstructured information increases the need for automatic systems that can "condense" information from various documents into a shorter- length, readable summary. Such summaries may be further required to cover a specific information needed (e.g., summarizing web search results, medical records, question answering and more).In our system, we will take data from various sources for a particular topic and summarize it for the convenience of the people, so that they don't have to go through so multiple sites for relevant data.*

***Key Words*: Summarize, Online Information, Unstructured Information, Condensed Information, Shorter Length**

1. INTRODUCTION

In order to solve the problem of visiting N sites and then getting relevant information we have a solution that is Automatic text summarization, which helps us summarize the entire context and gives you a brief summary based on what is relevant. Various methods have been proposed for this summarization task. These methods can be categorized based on two main dimensions i.e. extractive versus abstractive and supervised versus unsupervised. What we are going to use is the Extractive and Unsupervised learning technique. Extractive methods generate a summary using only text fragments extracted from the document(s) and Unsupervised learning is the training of machine using information that is neither classified nor labeled and allowing the algorithm to act on that information without guidance. Here the task of the machine is to group unsorted information according to similarities, patterns and differences without any prior training of data. Automatic text summarization is a machine learning technique ,that shortens the text document to create summary for our convenience. Now basically what we are using here is NLP i.e. Natural Language Processing. Summarization technologies typically include the following functionalities: identification of important information in source documents, identification and removal of duplicate information, summary information coverage optimization, and content ordering and rewriting to enhance readability.

2. LITERATURE REVIEW

In paper Performing literature review using text mining, Part III: Summarizing articles using Text Rank, they presented A text-mining-based algorithm, Text Rank, is used to perform automatic summary of academic articles, for effective and efficient literature review. The algorithm utilizes a weighted undirected graph to represent the sentences in an article, and uses Google Page Rank to order the sentences.

In the paper Stop Words in Review Summarization Using Text Rank, This paper presents a comparison of automatic review summarization with and without stop words. An extractive, unsupervised graph-based ranking model Text Rank is employed to highlight the differences between both approaches.

In paper Text Rank algorithm by exploiting Wikipedia for short text keywords extraction they proposed a graph-based algorithm to extract keywords for short text application by using Wikipedia as knowledge base. This algorithm uses Wikipedia as the knowledge base to improve the traditional Text Rank algorithm.

In the paper an overview of extractive text summarization proposes that text summarization is one of the most exciting research, this tells relationship between text mining and text summarization

3. PROPOSED SYSTEM

Our project focuses on providing a system design which could prepare a bullet-point summary for the user by scanning through multiple articles. What we have in existing system is that they don't provide any summary mechanism that will help in getting concise and reliable summaries of technological topics such as ML, AI etc. If a reader wants to get an overview, he has to go through multiple sources such as blogs, articles, question forums etc which is very rigorous and time consuming. So we are improvising this existing system to come up with summary for the user. We will apply the Text Rank algorithm on a dataset of scraped articles with the aim of creating a nice and concise summary. Also, that this would be essentially a single domain multiple documents summarization task, i.e., we will take multiple articles as input and generate a single bullet point summary.

The first step would be to concatenate all the text contained in the articles. Then split the text into individual sentences. Then do pre-processing on them. In the next step, we will

find vector representation (word embeddings) for each and every sentence. Similarities between sentence vectors are then calculated and stored in a matrix. The similarity matrix is then converted into a graph, with sentences as vertices and similarity scores as edges, for sentence rank calculation.

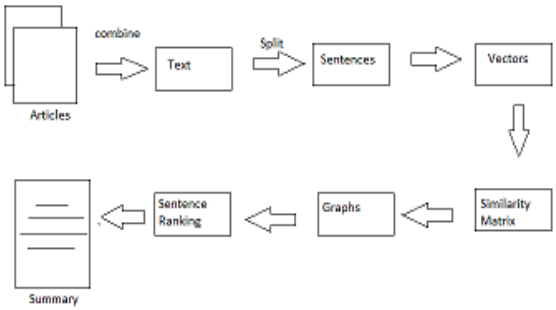

Fig.1 Flow Diagram of the project

3.1 READING FROM CSV FILE

We read the data from our CSV file which has our datasets and then concatenate them and form a text, now we split the text to individual sentences and do the further step on top of that. The columns are article_id, article_text, source, topic and when we do other tasks all of them is going to be applied on to article_text.

3.2 PREPROCESSING

This is cleaning of data to begin with on our datasets that contains information and some content collected from the websites. So we need to do the data preprocessing on this information. For achieving better results from the applied model in Machine Learning projects the format of the data has to be in a proper manner. One of the major forms of pre-processing is to filter out useless data. In natural language processing, useless words, are referred to as stop words. We apply preprocessing on the article_text which enables us to remove all the extra data from it which are of no use e.g.: "the", "a", "an", "in". etc

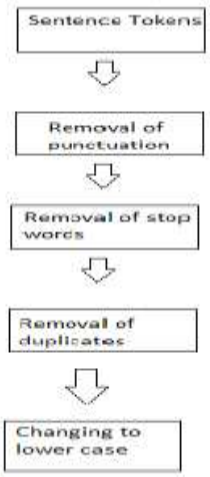

Fig.2 Preprocessing Flow Diagram

3.3 WORD EMBEDDING, SIMILARITY MATRIX AND GRAPHS

Word embeddings are vector representation of words. These word embeddings will be used to create vectors for our sentences. This is language modeling techniques used in NLP. Where words from vocabulary is mapped to vectors or numbers. Word embeddings are in fact a class of techniques where individual words are represented as real-valued vectors in a predefined vector space. The algorithm we use for word embedding is GloVe, The Global Vectors for Word Representation, or GloVe, algorithm is an extension to the word2vec method for efficiently learning word vectors, developed by Pennington, et al. at Stanford. The model is an unsupervised learning algorithm for obtaining vector representations for words. This is achieved by mapping words into a meaningful space where the distance between words is related to semantic similarity. We use this basically to find similarity between sentence vectors. Here now we create vectors for our sentences which are nothing but the article_text to which this is going to be implemented.

3.4 SIMILARITY MATRIX AND GRAPH

We will first fetch vectors (each of size 100 elements) for the constituent words in a sentence and then take mean/average of those vectors to arrive at a consolidated vector for the sentence. Based on vector of sentences we prepare a similarity matrix, i.e. find similarities between the sentences, and we will use the cosine similarity approach for this challenge. Let's create an empty similarity matrix for this task and populate it with cosine similarities of the sentences. Let's first define a zero matrix of dimensions (n * n). We will initialize this matrix with cosine similarity scores of the sentences. We will use Cosine Similarity to compute the similarity between a pair of sentences and initialize the matrix with cosine similarity scores. Before proceeding further, let's convert the similarity matrix into a graph. By definition graph is a combination of nodes(vertices) and identified pair of nodes(edges, links).The nodes of this graph will represent the sentences and the edges will represent the similarity scores between the sentences. Assuming that your matrix is an numpy array, you can use the method Graph=networkx.from_numpy_matrix('numpy_adj_matrix.npy') to draw the graph. On this graph, we will apply the Page Rank algorithm to arrive at the sentence rankings.

3.5 PAGE RANKING AND TEXT RANK

Before getting started with the Text Rank algorithm, there's another algorithm which we should become familiar with – the Page Rank algorithm. In fact, this actually inspired Text Rank**.** Page Rank is used primarily for ranking web pages in online search results. In order to rank these webpage, we would have to compute a score called the Page Rank score. This score is the probability of a user visiting that page.

To capture the probabilities of users navigating from one page to another, we will create a square matrix M, having n rows and n columns, where n is the number of web pages. Each element of this matrix denotes the probability of a user transitioning from one web page to another. For example, the highlighted cell below contains the probability of transition from w1 to w2.This way we find the probability and use page ranking.

The similarity between page and text rank is that in place of web pages, we use sentences. Similarity between any two sentences is used as an equivalent to the web page transition probability. The similarity scores are stored in a square matrix, similar to the matrix M used for Page Rank.

Finally, it's time to extract the top N sentences based on their rankings for summary generation. For our, based on similarity we take out top 2 or 3 sentences and summarize them. 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.

First, the words are assigned parts of speech, so that only nouns and adjectives are considered. Then a graph of words is created. The words are the nodes/vertices. Each word is connected to other words that are close to it in the text. In the graph, this is represented by the connections on the graph.

The algorithm is then run on the graph. Each node is given a weight of 1. Then the algorithm goes through the list of nodes and collects the influence of each of its inbound connections. The influence is usually just the value of the connected vertex (initially 1, but it varies) and then summed up to determine the new score for the node. Then these scores are normalized, the highest score becomes 1, and the rest are scaled from 0 to 1 based on that value. Each time through the algorithm gets closer to the actual "value" for each node, and it repeats until the values stop changing.

In post-processing, the algorithm takes the top scored words that have been identified as important and outputs them as key/important words. They can also be combined if they are used together often.

4. CONCLUSION

The entire project is focused on creating a system that gets concise summaries of technological topics. The implications of this would mean that knowledge gathering would be easier and time saving. This project can have additional features such as domain-unspecific so that it could be used for any applications, like travelling blogs, product review summarization and many more.

REFERENCES

[1] Dazhi Yang_ and Allan N. Zhang Singapore Institute of Manufacturing Technology "Title of the paper Performing literature review using text mining, Part III: Summarizing articles using Text Rank".

[2] Ali Toofanzadeh Mozhdehi, Mohamad Abdolahi and Shohreh Rad Rahimi title " Overview of extractive text summarization" .

[3] Wengen Li and Jiabao Zhao School of management and engineering title "Text Rank algorithm by exploiting Wikipedia for short text keywords extraction."

[4] Sonya Rapinta Manalu, Willy School of Computer Science title "Stop Words in Review Summarization Using Text Rank ".

[5]Blog Vidhya Analytics https://www.analyticsvidhya.com/blog/2018/11/introduction-text-summarization-textrank-python/

BIOGRAPHIES

Tanwi is currently pursuing her B.E degree in the Department of Information Science & Engineering, Mysore. Her B.E major project area is Machine Learning. This paper is survey paper of his B.E project.

Satanik Ghosh is currently pursuing his B.E degree in the Department of Information Science & Engineering, Mysore. His B.E major project area is Machine Learning. This paper is survey paper of his B.E project.

Viplav Kumar is currently pursuing his B.E degree in the Department of Information Science & Engineering, Mysore. His B.E major project area is Machine Learning. This paper is survey paper of his B.E project.

Yashika S Jain is currently pursuing her B.E degree in the Department of Information Science & Engineering, Mysore. Her B.E major project area is Machine Learning. This paper is survey paper of his B.E project.

Mr. Avinash B is Asst.Professor (FTC) in the Department of Information Science & Engineering,Mysore. He has received his M.Tech from VTU.