International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

IJIEMR Transactions, online available on 26th Apr 2022. Link :http://www.ijiemr.org/downloads.php?vol=Volume-11&issue= Spl Issue 05

DOI: 10.48047/IJIEMR/V11/SPL ISSUE 05/06

Title **TEXT SUMMARIZATION USING DEEP LEARNING**

Volume 11, SPL ISSUE 05, Pages: 35-39

Paper Authors
Mr. O. Shiv Bhagwan, Chunduri. Medha Lakshman Kumar, Gadi. Giri Venkata Pavan Kumar, Mogadala. Lakshmi Swetha, Kanchi. Saket

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

TEXT SUMMARIZATION USING DEEP LEARNING

Mr. O. Shiv Bhagwan[1], Chunduri. Medha Lakshman Kumar[2], Gadi. Giri Venkata Pavan Kumar[3],Mogadala. Lakshmi Swetha[4], Kanchi. Saket[5]
[1]Associate Professor, Dept of CSE, [2]18ME1A0522, [3]18ME1A0531, [4]18ME1A0558, [5]18ME1A0550
[1,2,3,4,5]Ramachandra College of Engineering, Eluru, A.P., India

cseplacements@rcee.ac.in , lakshma906@gmail.com, girivenkatapavankumar@gmail.com, swetha.mogadala@gmail.com, Kanchisaket155@gmail.com

Abstract
There has been an upsurge in intelligent text classification in numerous applications since deep learning techniques were introduced. The use of deep learning techniques to improve automatic text summarization is currently a hot topic of research. Traditional extractive text summarizing approaches relied significantly on human-engineered features in the past. It is, however, a time-consuming and exhausting process. To produce extractive summaries using deep learning, this article adopted a data-driven strategy. The suggested method employs paraphrasing techniques to determine whether or not a sentence is a contender for inclusion in the summary.

Keywords: Summarization; convolution neural networks; Long Short Term Memory(LSTM).

1. Introduction

1.1 Motivation

Abstractive and extractive text summarizing techniques can be divided into two groups. Additional categorization categories, such as single vs. multi-document sorting and mono-lingual vs. multi-lingual summarization, have been defined in the past based on a variety of other factors.

1.2 Problem Definition

Extractive text summarizing is a more straightforward and reliable method of generating summaries that involve choosing key sentences from a text and presenting them to the user. Each sentence is given a score, and the sentences with the highest scores are chosen to be included in the extract. In contrast to abstractive summaries, which require producing catchphrases and words and structuring them into intelligible decrees while still conveying an interpreted list of the material, this is a lot easier. It would require a significant amount of natural language processing, making it a ample more complex task. The purpose of this research is to generate extractive summaries utilizing a data- driven methodology and deep learning approaches to meet the goal of text summarization. This entails sorting through the pile of text and creating a list of phrases that are likely to be the most beneficial and include the text's main points.

1.3 Objective of the Project

Abstractive text summarizing techniques try to generate summaries that summarize the substance of the text in the same way as humans do after analysis any text. It employs

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

propagative methodologies that can produce meaningful phrases while maintaining the semantics of the unique text. This is seen as a challenging problematic to tackle, and several new ways have been offered in response to Deep Learning's boost.

1.4 Limitations of Project

Although most people don't think of summaries as lines taken directly from a book, they do aim to describe it in a way that takes the similar idea as the original. Though extractive précises are not obvious, they help the function of providing the greatest significant portion of the material, which can offer a good impression of what the book is about, as well as specific lines that can be quoted or referred to for other purposes. As a result, utilizing paraphrase detection, this work attempts short summaries to capture the essence. Additional purpose for this effort is to test this extract generating technique for Indian languages utilizing data-driven approaches.

2. System Analysis

2.1 Requisities Accumulating And Analysis

As ours is a scholarly leave, we followed IEEE Journals and amassed so many IEEE Relegated papers that we finally separated a Paper assigned "Singular web revisitation by setting and substance significance input, and for the investigation stage, we took refs from the paper and did writing review of certain papers and amassed all of the venture's requirements in this stage.

2.2 Existing System

Simple classifiers such as naive-Bayes classifiers, decision trees, clustering, and hidden Markov models were also trained using feature vectors that were hand-engineered founded on approximately of the characteristics given overhead. Sure studies looked at the usage of Genetic algorithms, which are algorithms that represent optimization issues and solve them by means of natural selection approaches such as mutation, cloning, and cross-overs.

2.3 Proposed system

In this research, deep learning was utilized to construct extractive summaries utilizing a data- driven strategy. The suggested method employs paraphrase techniques to determine if a sentence is acandidate for inclusion in the summary or not.

2.4 Software Requirements

The functional requirements or overall description documents cover the product's perspective and features, as well as the operating system and operating environment, graphical requirements, design limitations, and user documentation.

The application of requirements and implementation restrictions provides an overall picture of the project in terms of its strengths and weaknesses, as well as how to address them.

• Python idle 3.7 version (or) Anaconda 3.7 version(or) Jupiter (or) Google collab

2.5 Hardware Requirements

Minimum hardware requirements vary greatly depending on the program that an Enthought Python / Canopy / VS Code user is developing. Applications that need to keep big arrays/objects in memory will need more RAM, whereas those that need to do several calculations or activities quickly would needa faster CPU.

• Processor: minimum intel i3 • Operating system:Windows, Linux

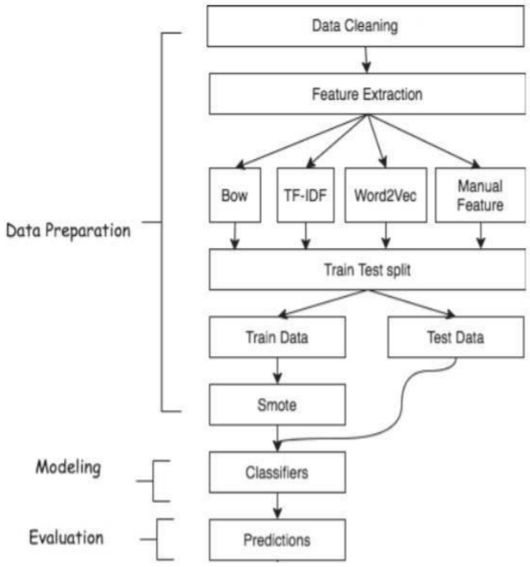
• RAM: at least 4 GB • Hard disc: at least 250 GB

2.6 System Design

ML model will learn and improve. To determine a model's efficacy, the data must first be divided into training and test sets. So, before training our models, we divided the

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

data into two sets: the training set, which comprised 70% of the total dataset, and the Test set, which comprised the remaining 30%. It was therefore necessary to apply a variety of performance indicators to our model's predictions.



3 Implementation

3.1 Data Gathering

The dataset was used to analyze human voice emotion. Also, the dataset's makeup. comprehend the link between several aspects The basic features and the complete dataset are shown. The dataset is then divided into two parts: training and testing algorithms. Furthermore, each class in the whole dataset is represented in about the correct proportionin both the training and testing datasets to create a representative sample. The percentage of the training and testing datasets that were used in the study.

3.2 Data Preparation

The information gathered might include missing values, resulting in inconsistencies. To get better results, data must be preprocessed to boost the algorithm's performance. Outliers must be deleted, and variable conversion must be performed. We utilize the map function to solve these problems.

3.3 Model Picking

Machine learning is the process of anticipating and detecting patterns to create appropriate outcomes after fully comprehending them. Data patterns are studied and learned using machine learning techniques. With each try, an

3.4 Algorithm

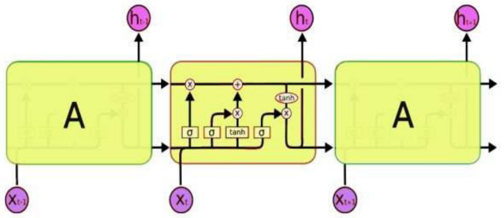
The long short-term memory (LSTM) architecture is a deep learning architecture based on a recurrent neural network (RNN). Unlike traditional feedforward neural networks, LSTM has feedback connections. It can handle both single data points (such as photographs) and entire data streams (such as speech or video).For example, LSTM can be used for unsegmented, linked handwriting recognition, speech recognition,[3][4] and anomaly detection in network traffic or IDSs (intrusion detection systems).

A typical LSTM unit consists of a cell, an input gate, an output gate, and a forget gate. The three gates regulate the flow of information into and out of the cell, and the cell retains values throughout time. LSTM networks are ideal for categorizing data.

Training: An RNN with LSTM units can be trained and supervised on a usual of training sequences, and can change each weight of the LSTM network in proportion to the derivative of the error (at the output layer of the LSTM network) concerning corresponding weight, using an optimization algorithm such as gradient descent combined with backpropagation through time to compute the gradients needed during the optimization process.

The use of gradient descent for conventional RNNs has the drawback that error gradients drop exponentially with the length of time between key occurrences. When error values are back-propagatedfrom the output layer to LSTM units, however, the error stays in the LSTM unit's cell. This "error carousel" keeps feeding back errors. The mistake is still in the cell of the LSTM unit. This "error carousel" feeds errors to each LSTM unit's gates until they learn to cut off the value.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org



3.5 Summary Generator

During the summary process, all information in the sentences in a document that has been determined assalient is included, regardless of its relevance. This is still a problem when summarizing a significant quantity of material [30] because the final summaries will include superfluous information.

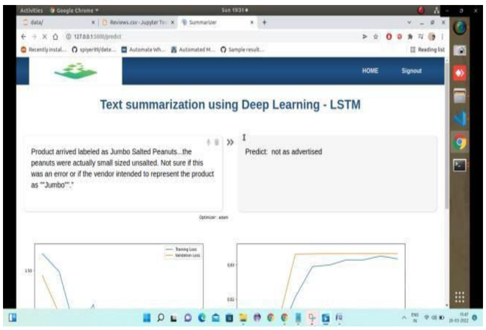
Two ways [30] can be used to overcome this issue:

Compression of Sentences: It "compresses" the sentences in the original text by removing extraneous details or irrelevant information from the selected key critical phrases to provide a compact summary using the Sentence Compression technique.

Sentence Fusion (or Information Fusion) has been utilized in Multi-Document

Summarization and Question and Answer Systems, where the common sentences or phrases throughout the texts are identified. without losing meaning or redundancy, recognized and merged into a single text or sentence.

Sentence Fusion must create new



sentences with an emphasis on improving coherence and redundancy.

4 Result

5 Conclusion

This study provides an overview of automatic text summarization [8], including methodologies, approaches, and assessment metrics.

An autonomous summarization system's major goal is to provide a summary with the least amount of duplication and significant information in the shortest amount of time possible. Future research will focus on the newest computational approaches for single and multi-document extractive summarization tasks.

References

Textbooks:
Programming Python, Mark Lutz Head First Python, Paul Barry
Core Python Programming, R. Nageswara Rao Learning with Python, Allen B. Downey
PAPERS
Giuseppe Carenini, Raymond Ng and Gabriel Murray, "Methods for Mining and Summarizing Text Conversations," 2011.
Dragomir R. Radev, Hongyan Jing and Malgorzata Budzikowska, "Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation," In ANLP/NAACL Workshop on Summarization, April 2000.
Lin, C. Y., and Hovy, E., "Automated Multi- document Summarization in NeATS," In Proceedings of the Human Language Technology (HLT) Conference, 2001.
Harabagiu, S. and Lacatusu, F, "Generating Single and MultiDocument Summaries with GISTEXTER," In Workshop on Text Summarization (In Conjunction with the ACL 2002 and including the DARPA/NIST sponsored

DUC 2002 Meeting on Text Summarization), 2002.
Radev, D, Weiguo, F and Zhang, Z, "Webinessence: A personalized web-based multi- document summarization and recommendation system," In NAACL Workshop on Automatic Summarization, 2001.
Svore, K., Vanderwende, L and Burges, C., "Enhancing SingleDocument Summarization by Combining RankNet and Third-Party Sources," In Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), p. 448–457, 2007.

Websites:

https://www.w3schools.com/python/
https://www.tutorialspoint.com/python/index.htm
https://www.javatpoint.com/python-tutorial
https://www.learnpython.org/