D Y PATIL INTERNATIONAL UNIVERSITY
AKURDI PUNE

Sector 29, Pradhikaran, Akurdi, Pune - Maharashtra, INDIA 411044
(Establishment by Maharashtra Act No. LXIII of 2017)

School of Computer Science, Engg. & Applications

TEXT SUMMARIZATION

Krish J Shetty

PRN: 20190802129
Email Id: 20190802129@dypiu.ac.in
Phone Number: 8828489577

Vedant R Landge

PRN: 20190802005
Email Id: 20190802005@dypiu.ac.in
Phone Number: 8237789479

ABSTRACT:

"We don't need a full report. Just give me a summary of what you did." This is something that most people find themselves getting thrown at – both in college as well as in their professional life. They slog for nights and prepare a comprehensive report only for the teachers/supervisors to read nothing but the summary. Writing the summary manually, however, is an extremely tedious, labor-intensive, and time-consuming task. This is where the awesome concept of Text Summarization using Deep Learning comes into the picture.

Project reports are one of the fragments of the problems that people face, we can also agree that the amount of textual data produced daily is increasing rapidly. The International Data Corporation (IDC) projects that the total amount of digital data circulating annually around the world used to be around 4.4 zettabytes in 2013 but will definitely hit 180 zettabytes by 2025. Now that is a lot of data! However, it is possible to create models that shorten extremely long pieces of text and provide us with a summary, thanks to Deep Learning, and save time as well as understand the key points effectively

KEYWORDS:

1. Deep Learning
2. Natural Language Processing (NLP)
3. the International Data Corporation (IDC)

LITERATURE SURVEY:

i. *Vishal Gupta and Gurpreet Singh:*

"A Survey of Text Summarization Extractive techniques". In this paper, the author describes the extractive summarization methods which comprise two parts Pre-Processing and Processing. In this paper, the pre-processing step is further divided into other subprocesses which are sentence segmentation, stop word removal, and stemming. In the processing step, the weights are given to the features used for extraction of summary from the large document respectively.

ii. *Saranyamol C S and Sindhu L:*

"A Survey on Automatic Text Summarization." In this paper, the author describes the various techniques used in automatic text summarization which are extractive text summarization and abstractive text summarization.

iii. *Rafael Ferreira, Luciano de Souza:*

Cabrala, Rafael Dueire Lins, Gabriel Pereira Silva, Fred Freitas, George D.C. Cavalcanti, Rinaldo Lima a, Steven J. Simske, Luciano Favaro, "Assessing sentence scoring techniques for extractive text summarization." This paper gives a brief description of various features used to perform extractive summarization and it also describes the methods for summary evaluation.

iv. *K. Vimal Kumar, Divakar Yadav:*

"An Improvised Extractive Approach for Hindi Text Summarization." This paper mainly laid emphasis on the Hindi text summarization. It also describes various features used for Hindi summarization using the extractive approach of text summarization. The author proposed a system that can generate the summary with 85 % accuracy.

v. *Vishal Gupta:*

"Hybrid Algorithm for Multilingual Summarization of Hindi and Punjabi Documents." The author of this paper has proposed a hybrid algorithm for Hindi and Punjabi text summarization. The algorithm proposed by the author is the first algorithm that can summarize both Hindi as well as Punjabi text.

vi. *Ani Nenkova:*

"Summarization Evaluation for Text and Speech: Issues and Approaches." This paper suggests methods for summary evaluation after the process of text summarization. Also, it describes some human models for summary evaluation.

INTRODUCTION (AIM & OBJECTIVES):

Text summarization is a key natural language processing (NLP) task that automatically converts a text, or a collection of texts within the same topic, into a concise summary that contains key semantic information. Basically, it is the problem of creating a short, accurate, and fluent summary of a longer text document. In order to address the ever-growing amount of text data available online, automatic text summarization methods are proving to be of immense need to help discover relevant information more efficiently as well as consume relevant information much faster. Machine learning models are usually trained for understanding the documents, gaining useful information, and giving out the required summarized texts as the output.

In this project, we will dive deep into the problem of text summarization in natural language processing.

METHODOLOGY:

1. LIBRARIES:

Before we start with the project, we need to first import all the libraries and dependencies. There are five libraries that we have thought of using as of now.

i. *Pandas:* the most commonly used tools for Data Science and Machine learning, which are used for data cleaning and analysis.
ii. *NumPy:* helps you implement best practices for data automation, model tracking, performance monitoring, and model retraining
iii. *Time:* provides many ways of representing time in code, such as objects, numbers, and strings
iv. *Re:* lets you check if a particular string matches a given regular expression (or if a given regular expression matches a particular string, which comes down to the same thing).
v. *Pickle: a* process of converting a Python object into a byte stream to store it in a file/database, maintain program state across sessions, or transport data over the network.

2. DATA:

After we get our libraries straight, we straightaway head to load the data into our project. We will be using the Pandas data frame for this purpose. Once the data is loaded into our project, we must then understand the data. Now, the dataset contains data collected from Inshorts and consists of five columns. Out of them, we will only be needing two of them, namely, Headline and Short. The rest three are of no use to us for the purpose of this project. Hence, we drop them.

3. Pre-processing:

Following are the pre-processing tasks we will be performing:

- Convert everything to lowercase
- Remove (‘s)
- Remove any text inside the parenthesis () x
- Eliminate punctuation and special characters
- Remove stop words
- Remove short words

4. METHOD:

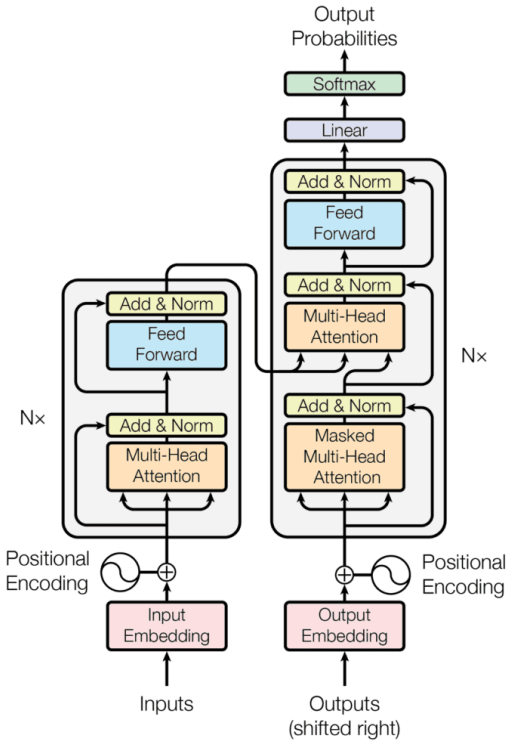
The method, that we will be using here, would be the ‘Abstractive Method’. This method will be implemented using Deep Learning

5. *ARCHITECTURE:*

There is a research paper "Attention Is All You Need". This paper introduces a novel architecture, namely '*Transformer*'. This new model uses the attention mechanism. Similar to LSTM, we use it to transform one sequence to another by following the encoder-decoder structure.

The transformer model does not need to rely on recurrence or convolution for generating outputs.

As mentioned earlier, the transformer model works on two concepts: Encoder and Decoder.



ALGORITHM:

The Transformer model runs as follows:

1. Each word of the input sequence is transformed into a d_{model}-dimensional embedding vector.
2. Each embedding vector representing an input word is augmented by summing it (element-wise) to a positional encoding vector of the same d_{model} length, hence introducing positional information into the input.
3. The augmented embedding vectors are fed into the encoder block consisting of the two sublayers explained above. Since the encoder attends to all words in the input sequence, irrespective if they precede or succeed the word under consideration, then the Transformer encoder is bidirectional.
4. The decoder receives as input its own predicted output word at time-step.
5. The input to the decoder is also augmented by positional encoding in the same manner done on the encoder side.
6. The augmented decoder input is fed into the three sublayers comprising the decoder block explained above. Masking is applied in the first sublayer in order to stop the decoder from attending to the succeeding words. At the second sublayer, the decoder also receives the output of the encoder, which now allows the decoder to attend to all the words in the input sequence.

7. The output of the decoder finally passes through a fully connected layer, followed by a SoftMax layer, to generate a prediction for the next word of the output sequence.

RESULTS & DISCUSSION:

In languages, what really matters is the order of the words apart from their positions in a particular sentence. Change the order, and the entire meaning of the sentence will change. That is why we used the Transformer architecture.

The input sequence first got converted into embeddings (with position encoding). Then, its fed to the encoder layer where it is processed to produce an encoded representation. Then the Decoder is fed with an empty sequence with only a start-of-sentence token as well as position encoding. The Decoder processes this along with the output from the Encoder layer and produces this encoded representation of the target sequence. The Output layer then converts it into word probabilities and produces an output sequence. The last word of the output sequence is taken as the predicted word. That word then gets filled into the second position of our Decoder input sequence, which now contains a start-of-sentence token and the first word. And then this is repeated until it predicts an end-of-sentence token.

The key reason why Transformers worked so efficiently is the Attention Layers. While processing a particular word, the Attention makes sure that other words, that are closely related to the word in consideration, are focused on as well. Hence, self-attention gives the model more information about the meaning of the word in question so that it can associate that word with the correct word.

CONCLUSION:

Therefore, we were able to successfully build a model that can summarize long texts into one sentence.

Text summarization can help us dive into a good number of problems and try to develop a solution for them. Like, for instance, elderly people can get help with a talk-back system that summarizes the news, which they ideally see in 3 hours, within minutes. We can even use this as a tool to help enhance the experience of meetings. Most of the time, when people are attending a meeting, like in an office per se, people tend to make messy keywords when their superiors ask them to jot down the key takeaways. Text summarization can help people with understanding the key takeaways and never missing on them.

Another field of application for text summarization is to help people understand what a person is trying to say in a completely different language. This can even help us bridge the language barrier.

REFERENCE:

1. https://medium.com/inside-machine-learning/what-is-a-transformer-d07dd1fbec04
2. https://www.dominodatalab.com/blog/transformers-self-attention-to-the-rescue
3. https://machinelearningmastery.com/gentle-introduction-text-summarization/#:~:text=Summaries%20reduce%20reading%20time.,less%20biased%20than%20human%20summarizers.
4. https://www.analyticsvidhya.com/blog/2019/06/comprehensive-guide-text-summarization-using-deep-learning-python/
5. https://towardsdatascience.com/a-quick-introduction-to-text-summarization-in-machine-learning-3d27ccf18a9f
6. https://towardsdatascience.com/transformers-explained-visually-part-1-overview-of-functionality-95a6dd460452
7. https://towardsdatascience.com/transformers-explained-visually-part-2-how-it-works-step-by-step-b49fa4a64f34
8. https://towardsdatascience.com/transformers-explained-visually-part-3-multi-head-attention-deep-dive-1c1ff1024853
9. https://towardsdatascience.com/transformers-explained-visually-not-just-how-but-why-they-work-so-well-d840bd61a9d3
10. https://towardsdatascience.com/transformers-141e32e69591