Text Summarization Using Deep Learning

Riddhesh Mahajan[1], Adhishree Vast[2], Sharayu Mhaske[3], Prof. Sachin Barahate[4]

[1-3]Students, Department of Computer Engineering, Padmabhushan Vasantdada Patil Prathisthan's College Of Engineering, Mumbai, Maharashtra, India
[4]Professor, Department of Computer Engineering, Padmabhushan Vasantdada Patil Prathisthan's College Of Engineering, Mumbai, Maharashtra, India

---***---

Abstract – *The aim of the project is to generate a text summary using the Encoder-Decoder model with the attention layer. The main aim of the model is to generate an abstractive summary with proper grammar and no repeated words. The interface of model and API is built using Django rest framework and Django Framework which are the web development frameworks of python. The main use case of the project is to generate the abstractive summary of the news articles.*

***Key Words*: Abstractive, Encoder-decoder, Django, Summary**

1. INTRODUCTION

Text Summarization is the technique of extracting notable data from the first content archive. In this method, the separated data is produced as a consolidated report and introduced as a clearly expressed new doc. Text Summarization can be classified into two categories: Extractive Summarization and Abstractive Summarization. In extractive summary, the important sentences are picked from the article and no changes are performed in the sentences, while the abstractive summarization model tries to build the sentences by itself and generate the summary using it. we present a model based on encoder Decoder architecture with an attention layer built using a gated recurrent unit that is capable of generating an abstractive summary.

2. OBJECTIVE

1. Develop an Encoder-Decoder model using Gated recurrent units.
2. Train the model to generate an abstractive summary.
3. Allow individuals to generate an abstractive summary of the article.

3. PROPOSED SYSTEM

3.1. Dataset

The dataset consists of 98000 news articles and summaries. The dataset contains news articles with different lengths.

3.2. Train validate split

90%-10% split is used. 90% data is for training the model and the 10% data is used for validation. The model accuracy is measured using this 10% data.

3.3. Data preprocessing

The dataset has null values with variable article sizes. Thus, it requires preprocessing before using it in model training.

Following pre-processing is done on the dataset: -

1. Removed rows containing null values
2. Prepended <start> and appended <end> tag to each article and summary to allow the article to detect the start and end of the article.
3. Padded each article and summary to create constant size input and output.
4. Added space between word and its punctuation.

3.4. Model architecture

Model architecture defines the logical connections of various functions used in the model creation and training. We used the batch size of 100 to train the model because we don't have the memory to store more than a hundred samples at a time to train the model. The model architecture is mainly consisting of an encoder network, Decoder network, and attention layer.

Encoder - network consists of 128 gated recurrent units it accepts the vectorized input. The output of the encoder is given to the attention layer.

Attention layer - attention layer is responsible for eliminating repeated words in the summary and generating proper grammar in the abstractive summary. It has 64 units. The output of the attention layer is provided to the Decoder.

Decoder - network consists of 128 gated recurrent units, it generates the numerical output, this output is mapped using the tokenized array which is built using Python at the start of the training of the model.

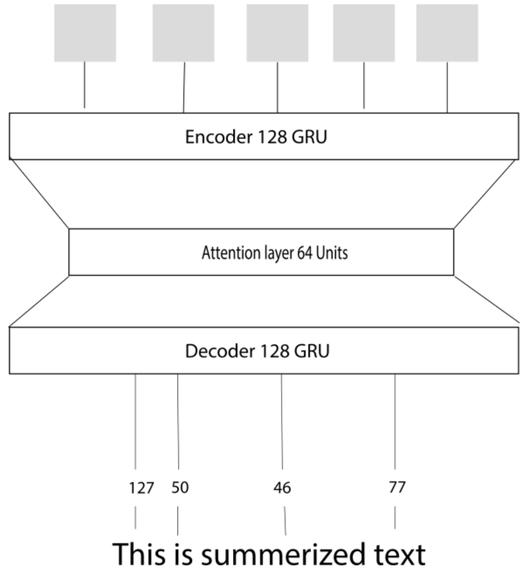

Fig -1: Layers in network

The above diagram shows the layers That are used in our project.

Using an extractive summarizer before an abstractive summarizer leads to better results. We used a pre-trained BERT model to generate an extractive summary and passed it as input to the encoder.

4. Literature Review

Anish Jadhav, Rajat Jain, Steve Fernandes, Sana Shaik (2019) -Text Summarization using Neural Network Encoder-Decoder model with attention layer performs better than only ender decoder model. Required more data to improve the accuracy of the model.

Rahimi Shohreh, Rad Ali Toofanzadeh, Mozhdehi and Mohamad Abdolahi (2017) -An overview on extractive text summarization Compared different summarization techniques

C.S. Lee, Z.W. Jian and L.K. Huang (2005)- A fuzzy ontology and its application to news summarization Application of fuzzy ontology to news summarization. Ontology performs best only for the Chinese language.

Christopher Ifeanyi Eke, Azah Norman, Liyana Shuib, Faith B. Fatokun, Isaiah Omame(2020)-The Significance of Global Vectors Representation in Sarcasm Analysis The globally accepted word embedding feature engineering techniques, GloVe for sarcasm classification has been studied. New words are not supported. The related vector may generate the wrong summary.

Paula Lauren,Guangzhi Qu, Guang-Bin Huang, Paul Watta, Amaury Lendasse (2017)- A low-dimensional vector representation for words using an extreme learning machine In this study, a low-dimensional vector representation of words has been demonstrated. Larger embeddings require more time to train.

5. HARDWARE SOFTWARE REQUIREMENT

5.1 Hardware

- Quad-Core Processor – Intel Core i5 or AMD Ryzen 5 or greater
- RAM – 16Gb
- Hard Disk – 40Gb Free Space
- GPU – AMD RX 480 (8GB) or Nvidia GTX 1060 (6GB) or greater

5.2. Software

5.2.1. Python

It is a programming language used for developing various software. It is also a leading language to build deep networks.

5.2.2. TensorFlow

TensorFlow is an open-source AI framework for machine learning and high-performance numerical computation. We have used the Keras API in Tensorflow to build the model. Tensorflow tokenize function is used to tokenize each word in the article.

5.2.3. Web Speech API

It is a built-in API in browsers to convert speech to text. Javascript language can be used to access the API on the HTML page.

5.2.4. Django

Django is a high-level Python Web framework used to develop web apps using python language.

5.2.5 Django REST framework

Django REST framework can be used with the Django app to create the API.

5.2.6 Pandas

Pandas is a fast and reliable data analysis library. We have used the Pandas library to remove null values from the dataset and other preprocessing techniques on the dataset.

6. RESULT AND DISCUSSIONS

We have compared 3 models to generate an abstractive summary. The Encoder-decoder model with the attention layer performs best.

RNN	Encoder-decoder model	Encoder-decoder model with Attention layer
Accuracy: 10%	Accuracy: 40%	Accuracy: 60% to 70%
Wrong grammar	Wrong grammar	Correct grammar
Most of the words are repeated	Few words are repeated	For most of the samples, words are not repeated. Unknown words may lead to some repeated words in summary.

Table -1: Comparison of three models

Training loss of Encoder-decoder model with attention layer for 200 epochs.

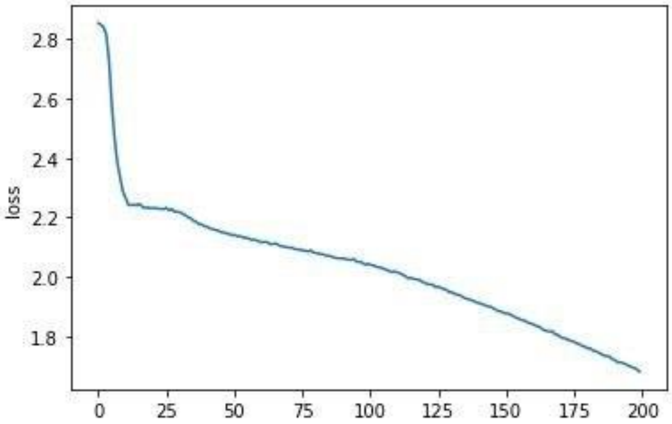

Chart -1: Training Loss Graph

Embedding sizes also performs the important role in abstractive summarization

50*50	100*100	200*200
The model is not able to generate any summary	The model generates summary but some summaries are meaningless	Model is generating proper summaries for 60% samples.

Table -2: Different embedding sizes

7. CONCLUSION

The Encoder-decoder model with the attention layer is generating better results than a simple RNN network. It is eliminating repeated words and using proper grammar to generate the summary. Providing an extractive summary as the input to generate an abstractive summary, shows better results than processing a complete article. If this model is used in the actual world, then it requires continuous training to generate the summary of the latest news.

8. REFERENCES

[1] Anish Jadhav, Rajat Jain, Steve Fernandes, Sana Shaikh (2019) Text Summarization using Neural Networks

[2] Rahimi Shohreh, Rad Ali Toofanzadeh Mozhdehi and Mohamad Abdolahi (2017) An overview on extractive text summarization"

[3] C. S. Lee, Z. W. Jian and L. K. Huang (2005) A fuzzy ontology and its application to news summarization

[4] Andhale Narendra and L. A. Bewoor (2016) An overview of text summarization techniques

[5] Barrios Federico et al., (2016.) Variations of the similarity function of textrank for automated summarization

[6] Nallapati Ramesh, Bowen Zhou and Mingbo Ma (2016) Classify or select: Neural architectures for extractive document summarization

[7] Jain Aditya, Divij Bhatia and Manish K. Thakur (2017) Extractive Text Summarization Using Word Vector Embedding

[9] Christopher Ifeanyi Eke ; Azah Norman ; Liyana Shuib ; Faith B. Fatokun ; Isaiah Omame (2020) The Significance of Global Vectors Representation in Sarcasm Analysis

[10] Paula Lauren ; Guangzhi Qu ; Guang-Bin Huang ; Paul Watta ; Amaury Lendasse (2017) A lowdimensional vector representation for words using an extreme learning machine Neungsoo Park ; Hyung Keun Ahn (2019) Multi-Layer RNN-based Short-term Photovoltaic Power Forecasting using IoT Data.