IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

TEXT SUMMARIZATION USING DEEP LEARNING

[1]Dr. Jayasudha K, [2]Mr. Thanmay D, [3]Mr. Praveen M Sindagi, [4]Mr. N Pradeep P Pai, [5]Mr. Jeevan Subramanyam

[1]Associate Professor, [2,3,4,5] Students 8th Semester
Department of ISE,
Atria Institute of technology, Bangalore, India

Abstract - *Text summarization could be a process of extracting the context of an oversized document and summarizing it into a smaller paragraph or some sentences. Text summarization plays an important role in saving time in our day to day life. It's also employed in many bigger project implementations of classification of documents or in search engines. This paper presents a technique of achieving text summaries accurately using specific deep learning methods.*

Key Words: Deep Learning, Long Short Term Memory, Natural Language Processing, Text Summarization

I. INTRODUCTION

Text summarization is the process of manufacturing a brief and concise summary by capturing the vital information and also the comprehensive meaning. Text summarization is achieved by linguistic communication processing techniques by using algorithms like page rank algorithms etc. While these algorithms fulfill the target of text summarization, they cannot generate new sentences which don't seem to be within the document like humans. They will even have grammatical errors. This can be avoided using Deep Learning. The utilization of deep learning builds a good and comparatively fast model for text summarization. The employment of deep learning methods helps us generate summaries which might be formed with new phrases and sentences and also which are grammatically correct.

Text Summarization is broadly classified into two types:

1.1 Extractive Text Summarization

The extractive text summarization involves extracting key phrases from the source document and combining them to make a summary. We identify important words or phrases from the text and extract only the most relevant sentences for the summary.

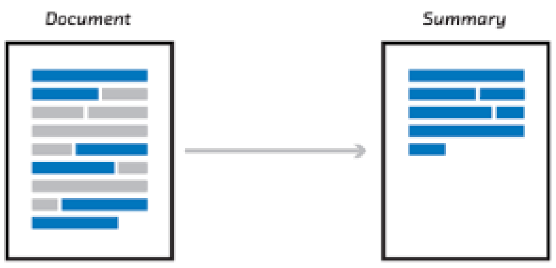


Figure1 Extractive Text Summarization

1.2 Abstractive Text Summarization

The abstractive text summarization can create new phrases and sequences that relay the most useful information from the original source. The sentences generated through this method may or may not be present in the original document.

2 DATA ANALYSIS

Performing basic pre-processing steps is extremely important before we get to the model building part. Using messy and unclean text data could be a potentially disastrous move. So, we are going to drop all the unwanted symbols, characters, etc. from the text that don't affect the target of our problem.

We will perform the below preprocessing tasks for our data:

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

3 METHODS

3.1 CleanData()

It is used to clean the data by using preprocessing steps as mentioned earlier in the paper.

3.2 BUILD DATASET()

It is used to build, train and test the data sets provided.

3.3 BuildDict()

It is used to build a dictionary where keys are words and values are random but unique numbers. It also builds another dictionary with keys as unique numbers and values as words. These are used in tokenization of words so that the input to the model is a set of numbers instead of words so as to make the computation easier using vectors.

3.4 TOKENIZE()

It is used to tokenize the input data and send it to the model. Tokenizing data is important as the networks need numerical data to work on rather than raw data with characters and strings.

4 DATA MODEL

The model used here is sequence to sequence model. Sequence-to-sequence learning could be a training model that may convert sequences of one input domain into the sequences of another output domain. It's generally used when the input and output of a model may be of variable lengths. It's a technique of encoder-decoder based artificial intelligence that maps an input of sequence to an output of sequence with a tag and a spotlight value. The thought is to use two LSTMs which will work along with a special token and take a look at to predict the following state sequence from the previous sequence.

4.1 Encoder-Decoder architecture:

The Sequence to Sequence model uses a technique of encoder- decoder is based artificial intelligence shown in fig.2. Encoder-Decoder architecture is employed in predicting sequences when the length of output and input file may vary. The input sequence is read entirely by the encoder and hard and fast length content is generated. The interior representation captures the complete context of the computer file sequence. The decoder network uses this representation to predict the output words until the top of the sequence token is reached.

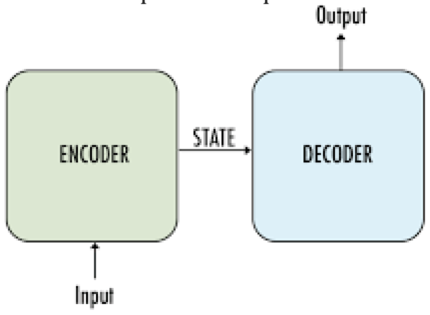


Figure 2 Encoder-Decoder

4.2 Encoder

An encoder is an LSTM network which reads the whole input sequence. At any time step, one word from the input sequence is read by the encoder. It then processes the input at every step and captures the context and therefore the key The decoder is additionally an LSTM network. It reads the complete representation generated by the encoder one word at a time step. It then predicts the identical sequence offset by just one step. The decoder is trained to predict the subsequent word within the output sequence given the previous word supporting the contextual memory stored by the LSTM architecture. Two special tokens are added at the start and at the top of the target sequence before feeding it to the decoder. We start predicting the target sequence by passing one word at a time. The primary word of output of the decoder is usually token. The top of the output sequence is represented by a token.

Information associated with the input sequence. It takes each word of input(x) and generates the hidden state output (h) and therefore the cell state which is an inside state(c). The hidden state (hi) and cell state (ci) of the last time step are the inner representation of the entire input sequence which can be used to initialize the decoder.

4.3 Decoder

The decoder is also an LSTM network. It reads the entire internal representation generated by the encoder one word at a time step. It then predicts the same sequence offset by one time step. The decoder is trained to predict the next word in the output sequence given the previous word based on the contextual memory stored by the LSTM architecture. Two special tokens <start> and <end> are added at the beginning and at the end of the target sequence before feeding it to the decoder. We start predicting the target sequence by passing one word at a time. The first word of output of the decoder is always <start> token. The end of the output sequence is represented by <end> token.

4.4 Attention layer

A Sequence to Sequence model with an attention mechanism consists of encoder, decoder and an attention layer. Attention mechanism is employed to secure individual parts of the input which are more important at that exact time. It is often implemented by taking inputs from when step and giving weightage to time steps. The weightage depends on the contextual importance of that specific time step. It helps listen to the foremost relevant parts of the input file sequence so the decoder can optimally generate the subsequent word within the output sequence.

5 IMPLEMENTATION

We implement the abstractive technique using a deep learning technique called Long Short Term Memory (LSTM) which is a type of Recurrent Neural Network Algorithms. The data used for this project is a manually curated dataset of news articles from prominent news agencies.

5.1 DATA

The data used is our custom dataset. It has two features: article and highlights. The article includes the document that is to be summarized and it's a news article. Highlights are the headlines of the corresponding news which are used as summaries. We will be focusing on the article feature for our project. The model used is the abstractive method which is implemented using deep learning techniques.

5.2 Algorithm

The algorithm used is T5 (Text-To-Text Transfer Transformer). T5 is an encoder-decoder model pre-trained on a multi-task mix of unsupervised and supervised tasks, for which each task is converted into a text-to-text format. The T5 algorithm is shown in figure 3. T5 reframes all NLP tasks into unified text to text format where the input and output are always converted into text streams. It allows to do document summarization in a better way.

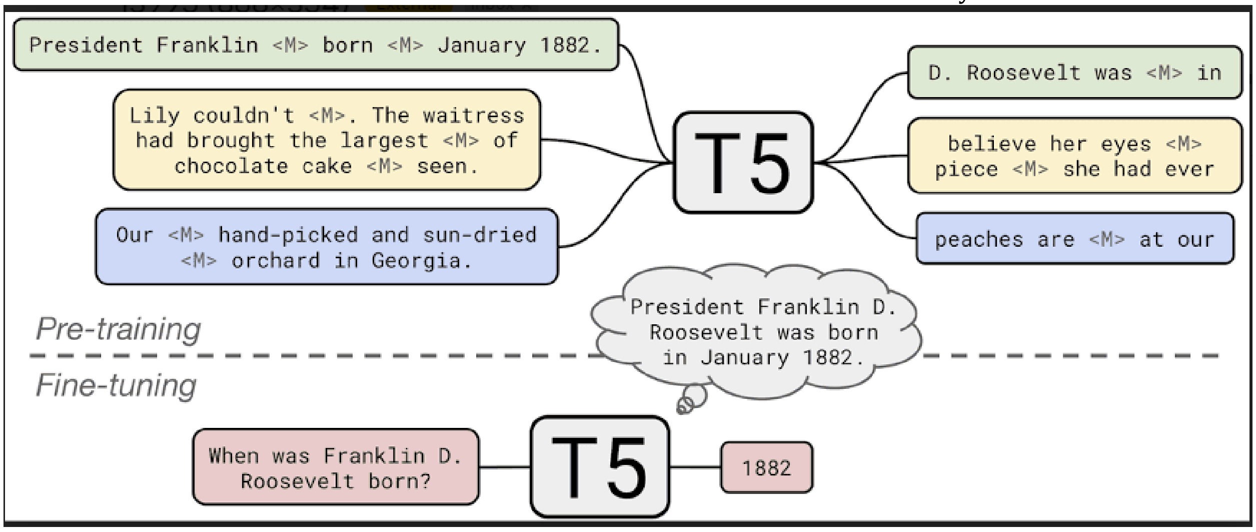

President Franklin <M> born <M> January 1882.
Lily couldn't <M>. The waitress had brought the largest <M> of chocolate cake <M> seen.
Our <M> hand-picked and sun-dried <M> orchard in Georgia.
T5
D. Roosevelt was <M> in
believe her eyes <M> piece <M> she had ever
peaches are <M> at our
Pre-training
Fine-tuning
President Franklin D. Roosevelt was born in January 1882.
When was Franklin D. Roosevelt born?
T5
1882

Figure3 T5 Algorithm

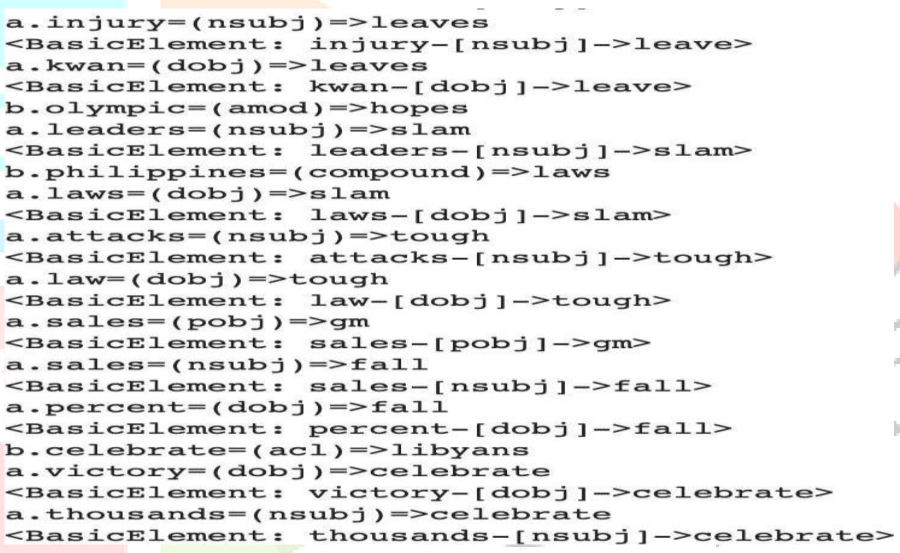
6 RESULTS

In this text summarization model, we use ROUGE (Recall-Oriented Understudy for Gisting Evaluation), a metric for giving the scores for text summarization, by comparing with previous text summaries. It's used for evaluating automatic text summarization and machine translations. It consists of a collection of metrics which are used for evaluation. It compares the result automatically produced using the model by the machine against human produced results. Within the text summarization model, the human produced summaries which are used for reference are reference summaries and also the summary generated using the models are called system summaries. The model is evaluated to support the extent to which the system summaries are almost like the reference summaries. This is often captured by recall. Here, Recall means what proportion the system summary is ready to capture or recover the essence of reference summary. If we are just considering the individual words, it is often computed as:

ROUGE= (Number_of_overlapping_words) ÷ (Total_words_in_reference_summary)

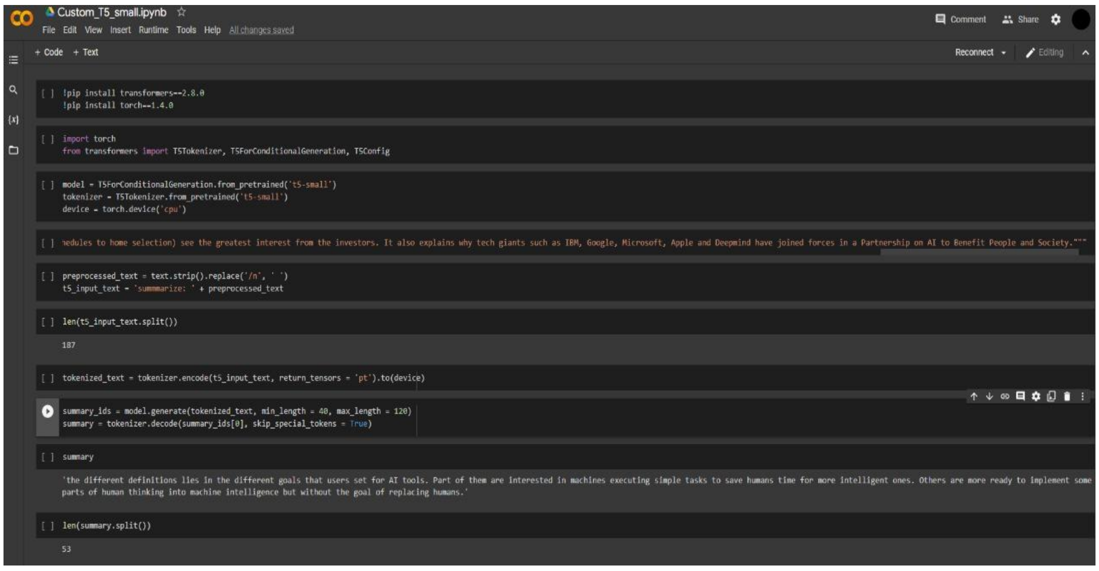
ROUGE-1	: refers to the overlap of unigrams between the system summary and reference summary.
ROUGE-2	: refers to the overlap of bigrams between the system and reference summaries.
ROUGE-N	: measures unigram, bigram, trigram and higher order n-gram overlap.

Figure4 shows the values of statements while running ROUGE scores on unigrams. Figure5 shows the results of text summarization in which 250 word length of data document is provided as input which is summarized and reduced to 100 word document.



```
a.injury=(nsubj)=>leaves
<BasicElement: injury-[nsubj]->leave>
a.kwan=(dobj)=>leaves
<BasicElement: kwan-[dobj]->leave>
b.olympic=(amod)=>hopes
a.leaders=(nsubj)=>slam
<BasicElement: leaders-[nsubj]->slam>
b.philippines=(compound)=>laws
a.laws=(dobj)=>slam
<BasicElement: laws-[dobj]->slam>
a.attacks=(nsubj)=>tough
<BasicElement: attacks-[nsubj]->tough>
a.law=(dobj)=>tough
<BasicElement: law-[dobj]->tough>
a.sales=(pobj)=>gm
<BasicElement: sales-[pobj]->gm>
a.sales=(nsubj)=>fall
<BasicElement: sales-[nsubj]->fall>
a.percent=(dobj)=>fall
<BasicElement: percent-[dobj]->fall>
b.celebrate=(acl)=>libyans
a.victory=(dobj)=>celebrate
<BasicElement: victory-[dobj]->celebrate>
a.thousands=(nsubj)=>celebrate
<BasicElement: thousands-[nsubj]->celebrate>
```

Figure4 ROUGE Unigram Values



```
!pip install transformers==2.8.0
!pip install torch==1.4.0

import torch
from transformers import T5Tokenizer, T5ForConditionalGeneration, T5Config

model = T5ForConditionalGeneration.from_pretrained('t5-small')
tokenizer = T5Tokenizer.from_pretrained('t5-small')
device = torch.device('cpu')

hedules to home selection) see the greatest interest from the investors. It also explains why tech giants such as IBM, Google, Microsoft, Apple and Deepmind have joined forces in a Partnership on AI to Benefit People and Society."""

preprocessed_text = text.strip().replace('/n', ' ')
t5_input_text = 'summarize: ' + preprocessed_text

len(t5_input_text.split())

187

tokenized_text = tokenizer.encode(t5_input_text, return_tensors = 'pt').to(device)

summary_ids = model.generate(tokenized_text, min_length = 40, max_length = 120)
summary = tokenizer.decode(summary_ids[0], skip_special_tokens = True)

summary

'the different definitions lies in the different goals that users set for AI tools. Part of them are interested in machines executing simple tasks to save humans time for more intelligent ones. Others are more ready to implement some
parts of human thinking into machine intelligence but without the goal of replacing humans.'

len(summary.split())

53
```

Figure5 Text summarization Results

7 CONCLUSION

The increasing growth of the net has made an enormous amount of knowledge available. It’s quite difficult for humans to summaries large amounts of text manually. Thus, there's a desire for automatic summarization tools during this age of knowledge overload.

The International Data Corporation (IDC) projects that the overall amount of digital data circulating annually round the world would increase from 4.4 zettabytes in 2013 to almost 180 zettabytes in 2025. That’s a large amount of information circulating within the digital world. There's a dire need of algorithms which may automatically shorten the quantity of information with accurate summaries that capture the essence of the intended messages. Furthermore, applying text summarization reduces reading time and accelerates the method of researching for information playing a serious role within the current era of rapid development and digitalization.

Humans are generally quite good at this task as we've got the capacity to know the meaning of a text document and extract salient features to summarize the documents using our own words. However, automatic methods for text summarization are crucial in today’s world where there's an overabundance of information and lack of manpower in addition as time to interpret the information.

REFERENCES

[1] Gupta, V., & Lehal, G. S. (2009). A survey of text mining techniques and applications. Journal of emerging technologies in web intelligence, 1(1), 60-76.

[2] Tas, O., & Kiyani, F. (2007). A survey automatic text summarization. Press Academia Procedia, 5(1), 205-213.

[3] Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., & Kochut, K. (2017). Text summarization techniques: a brief survey. arXiv preprint arXiv:1707.02268

[4] Nenkova, A., & McKeown, K. (2012). A survey of text summarization techniques. In Mining text data (pp. 43-76). Springer, Boston, MA.

[5] Kanapala, A., Pal, S., & Pamula, R. (2019). Text summarization from legal documents: a survey. Artificial Intelligence Review, 51(3), 371-402.

[6] Wang, L., Yao, J., Tao, Y., Zhong, L., Liu, W., & Du, Q. (2018). A reinforced topic-aware convolutional sequence-to-sequence model for abstractive text summarization. arXiv preprint arXiv:1805.03616.

[7] Kryscinski, W., McCann, B., Xiong, C., & Socher, R. (2020, November). Evaluating the factual consistency of abstractive text summarization. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 9332-9346).

[8] Song, S., Huang, H., & Ruan, T. (2019). Abstractive text summarization using LSTM-CNN based deep learning. Multimedia Tools and Applications, 78(1), 857-875.

[9] Cai, H., Zheng, V. W., & Chang, K. C. C. (2018). A comprehensive survey of graph embedding: Problems, techniques, and applications. IEEE Transactions on Knowledge and Data Engineering, 30(9), 1616-1637.

[10] Kobayashi, V. B., Mol, S. T., Berkers, H. A., Kismihók, G., & Den Hartog, D. N. (2018). Text mining in organizational research. Organizational research methods, 21(3), 733- 765.

[11] Salloum, S. A., Al-Emran, M., Monem, A. A., & Shaalan, K. (2018). Using text mining techniques for extracting information from research articles. In Intelligent natural language processing: trends and applications (pp. 373-397). Springer, Cham.

[12] Qiang, J., Qian, Z., Li, Y., Yuan, Y., & Wu, X. (2020). Short text topic modeling techniques, applications, and performance: a survey. IEEE Transactions on Knowledge and Data Engineering