Abstractive Text Summarisation

CS4099D Project
End Semester Report

Submitted by

Bhukya Vasanth Kumar	**(B180441CS)**
Billa Amulya	**(B180404CS)**
Sevakula Jyothi	**(B180359CS)**

Under the Guidance of

Dr. Raju Hazari
Assistant Professor

Department of Computer Science and Engineering
National Institute of Technology Calicut
Calicut, Kerala, India - 673 601

May, 2022

NATIONAL INSTITUTE OF TECHNOLOGY CALICUT
KERALA, INDIA - 673 601

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that this is a bonafide report of the project work titled

ABSTRACTIVE TEXT SUMMARISATION

done by

Bhukya Vasanth Kumar
Billa Amulya
Sevakula Jyothi

of Eighth Semester B. Tech, during the Winter Semester 2021-'22, in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering of the National Institute of Technology, Calicut.

Hazari

(Dr. Raju Hazari)
(Assistant Professor)
Project Guide

05-05-2022
Date

DECLARATION

I hereby declare that the project titled, **Abstractive text summarisation**, is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or any other institute of higher learning, except where due acknowledgement and reference has been made in the text.

Place : NIT Calicut
Date : 05-05-2022

Name : Bhukya Vasanth Kumar
Roll. No. : B180441CS
[illegible]

Name : Billa Amulya
Roll. No. : B180404CS
B. Amulya

Name : Sevakula Jyothi
Roll. No. : B180359CS
S. Jyothi

Abstract

Now a days, Text summarization has become important as the amount of text data available online grows at an exponential rate. Most of the text classification systems require going through a huge amount of data. In general, Producing exact and meaningful summaries of big texts is a time-consuming endeavour. Hence generating abstract summaries which retain the key information of the data and using it to train machine learning models will make these models space and time-efficient. Abstractive text summarization has been successful in moving from linear models to nonlinear neural network models using sparse models [1]. This success comes from the application of deep learning models on natural language processing tasks where these models are capable of modeling the interrelating patterns in data without handcrafted features. The Text to Text Transfer Transformer(T5) approach was used to investigate the text summarization problem, and the results showed that the Transfer Learning based model performed significantly better for abstractive text summarization than the Sequence to Sequence Recurrent Model.

ACKNOWLEDGEMENT

We would like to express our sincere and immense appreciation to our guide and coordinator Dr. Raju Hazari and Dr. Arun Raj Kumar P, who have guided us throughout the course of the final year project.We would also like to thank the numerous authors who have contributed to research in NLP and their publications which have guided us to get an in-depth understanding through literature survey. We would like to thank our parents and the faculty members for motivating us and being supportive throughout our work. We also take this opportunity to thank Department of CSE, our friends who have cooperated with us throughout the course of the project.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The advancement of natural language processing (NLP) over the years has created more possibilities in the way we can manipulate data. With the phenomenal growth in the internet and technology, everything now in front of us is data that is being used for various purposes.There is a vast amount of information and papers available on the internet that can be used for a variety of purposes. As the number of papers available grows, so does the requirement for in-depth research in the field of automatic text summarization.

The text summarisation should produce a precise summary from the given text without losing the key data and comprehensive meaning and can be viewed as data compression and information understanding both are related to key information retrieval. Since the last few years, various approaches and models have been designed for abstractive summarization and the summarisation problem is one of the demanding and crucial problems in NLP tasks, yet it is one of the least solved problems and research has been continuing in this area to tackle the problem in an efficient way. Text summarisation is widely used in areas like news article summary, tracking patient's health history, search engines, etc. Apps like In-shorts use this method to summarise news and provide headlines. Search engines like Google Chrome use them to generate snippets of products, and to facilitate headlines for news across the globe. The word ‘abstractive’ denotes a summary of sentences that are not directly extracted from the source, but a compressed paraphrasing of the main key contents of the document, potentially using/adding vocabulary and the grammar that is unseen in the source document [2].

A human can produce a summary of a text in various ways based on how they understand it and the keywords they use. Doing the same with machines is challenging and hence, text summarisation is considered to be limited. Since, the machines have a deficiency of human knowledge, text summarisation turns out to be a non-trivial task [23] and the results vary.

1.1 Definitions

1.1.1 Summarisation

[21] defines summary as ”a text created from one or more texts that provides crucial information from the original text(s) and is no more than half the length of the original text(s), and frequently much less..”

1.1.2 Transformer

The transformer is an NLP-based architecture that uses self-attention to calculate representations of its input and output, allowing it to handle sequence-to-sequence problems without the use of RNNs.

1.1.3 Natural Language Processing

NLP (Natural Language Processing) is a fundamental component of Artificial Intelligence that explains a computer program's capacity to interpret human language as it is spoken or written.

Chapter 2

Problem Statement

To create a brief, precise summary of a lengthy text while keeping the content's key information Using natural language processing (NLP) techniques.

2.1 Overview of the problem

2.1.1 Input - Output

Input:

1. Data(Sentence/Paragraph)
2. Original summary of data (Used for calculating accuracy)

Output:

Based on abstractive text summarization, a concise summary is generated.

2.1.2 Diagramatic Representation

The Diagramatic representation of the Input and Output Design is as shown in the Figure 2.1

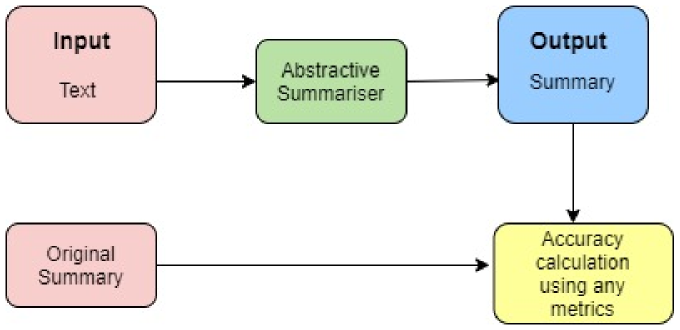


Figure 2.1: Input and Output Diagram

2.1.3 Data Set

The experiment was performed using Kaggle's News Summary dataset..The dataset consists of 4515 examples and contains Author_name, Headlines, Url of Article, Short text, Complete Article. The dataset only scraped news stories from Hindu, Indian Times, and Guardian, and only summarised news from Inshorts.

Chapter 3

Literature Survey

3.1 History of Text Summarisation

The number of summarisation models introduced every year has been increasing rapidly. Advancements in neural network architectures, and the Large-scale data made it possible to move away from expert knowledge and heuristics and toward data-driven techniques based on end-to-end deep neural models..

3.2 Challenges

A summary can be defined as a text that represents the main ideas or key information in an original text in less space. Suppose, if all the sentences in the original text are important, then the process of summarization would be less effective as the size of the summary would effect its informativeness. The main challenge in the process of summarization is identifying the informative segments and finally generating it as a concise text [18].

3.3 Quality of Summary

One of the most challenging problems is determining the quality of a summary because there is no such thing as a "ideal" summary. Human summarises only meet up to 60% of the time when measuring sentence content overlap in several circumstances, like as news articles[13]. In general, summarization can be characterised as the process of identifying, interpreting, and producing information about a given topic.

The major purpose of this portion of the identification process is to keep the most relevant, central issues. The major goal of the interpretation section is to accomplish compression by re-interpreting and fusing the picked out subjects into more concise ones.This is quite important because abstracts are usually much shorter than their equivalent extracts. The generation phase, on the other hand, is quite difficult since reformulating the extracted and fused content into a text with new phrases is quite difficult.

3.4 Current Approaches

Current approaches to text summarization uses copying mechanisms multi-task and multi-reward training techniques, graph-based methods that basically involve arranging the input text in a graph and then using ranking or graph traversal algorithms in order to construct the summary, reinforcement learning strategies , and hybrid extractive-abstractive models [**?**] few among them is text rank algorithm.

Initially, statistical methods were employed to provide a score to each sentence, and then the sentences with the highest scores were chosen. Various techniques were employed to calculate this score, such as TF-IDF [19], Bayesian models [14], etc. While these techniques were able to compute a sound summary by key phrase extraction, all of them were mostly extractive approaches and were simply trimming the original text into some summary.

Then the main focus is on Machine Learning [15] for summarization, such as Bayesian Learning Models as was done in the paper [14]. These machine learning techniques proved to be successful for pattern recognition in texts and establishing a correlation between different words.Because the order of words is crucial for natural language understanding and production, every text or sentence can be conceived of as sequential data. In order to process sequential data, the architecture needs to retain and memorize information with the help of some memory.

3.5 Concept of Self Attention

Reading comprehension, abstractive summarization, textual entailment, and learning task-independent phrase representations have all been successfully utilised with self-attention [6, 17, 8, 9]. End-to-end memory networks use a recurrent attention mechanism rather than sequence aligned recurrence to perform well on simple-language question answering and language modelling tasks [10]. To our knowledge, the Transformer is the first transduction model to calculate representations of its input and output using only self-attention rather than sequence aligned RNNs or convolution.

3.6 Transformers

One of the main advantages of the transformers is parallelisation purely depending on self attention with normalisation. The basic idea behind the self attention is that they allow inputs to interact with inputs i.e calculation of all other inputs with respect to one input.

Chapter 4

Proposed Work

4.1 Transformer

The transformer is an NLP-based architecture that uses self-attention to calculate representations of its input and output, allowing it to handle sequence-to-sequence problems without the use of RNNs. Figure 4.1 represents the Transformer Model With all components.

4.1.1 Encoder

The encoder is a 6 layered stack.Each layer is identical, including a feed-forward neural network and a multi-head attention mechanism. In the encoder, a residual connection is used around every two sub-layers, followed by layer normalisation.
Thus, the output of every layer is *LayerNormalisation(x+Sublayer(x))* as shown in Figure **??**.

1. x - Input to the Sublayer

2. Sublayer(x) - Function implemented by the Sublayer

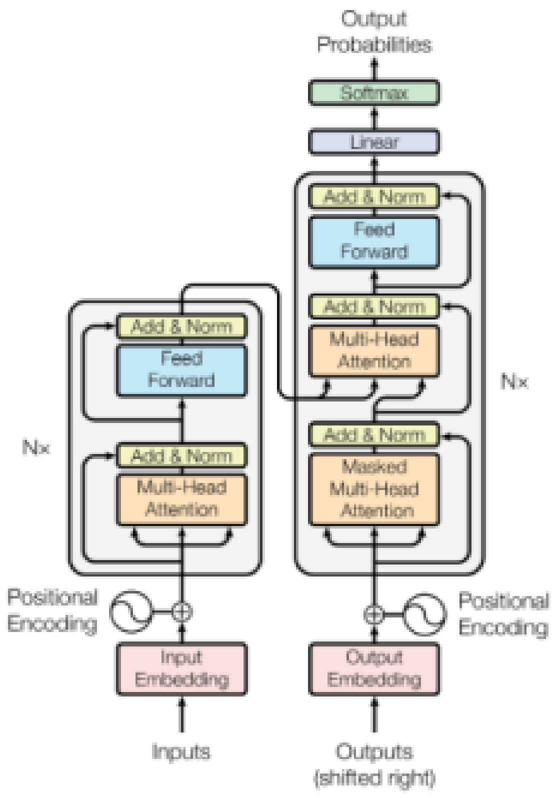


Figure 4.1: Transformer Model

4.1.2 Decoder

The decoder also consists of 6 layers like an encoder as shown in Figure 4.2. We also have a Masked multi-headed attention sublayer in addition to the multi-head attention mechanism and the feed-forward neural network in which the scope of the attention is restricted to the words that occurred before the given word. This is done in order to make the model learn.

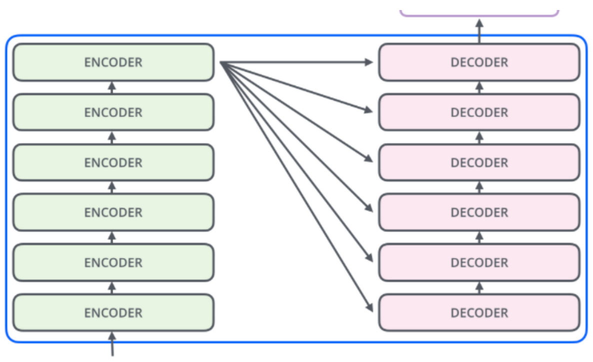


Figure 4.2: Encoder-Decoder Stacks

4.2 Text-to-Text Transformer

4.2.1 Introduction

T5 is the abbreviation for ”Text-to-Text Transfer Transformer” [4]. The main idea behind the T5 model is transfer learning [20]. The model as shown in Figure 4.3 was initially trained on a task containing large text in Transfer Learning before it was finely tuned on a downstream task so that the model picks up general-purpose skills and knowledge that can be applied to tasks like summarization [4] T5 uses a sequence-to-sequence creation strategy that feeds the encoded input to the decoder via cross-attention layers and provides autoregressive decoder output.

We have fine-tuned a T5 model [4], where the encoder takes an input a series of tokens which are mapped to a sequence of embeddings In this paper, we have used Glove embeddings dataset. In the encoder block, there is a self attention layer and a feed forward network, which are two subcomponents. The decoder and encoder are similar in structure, except that there's a generalized attention mechanism after every self attention layer. This enables the model to just work with the previous outputs. The final decoder block produces an output which is fed into another layer. The final layer is a thick layer with a softmax activation function. The weights from this layer's output are supplied into the embedding matrix's input.

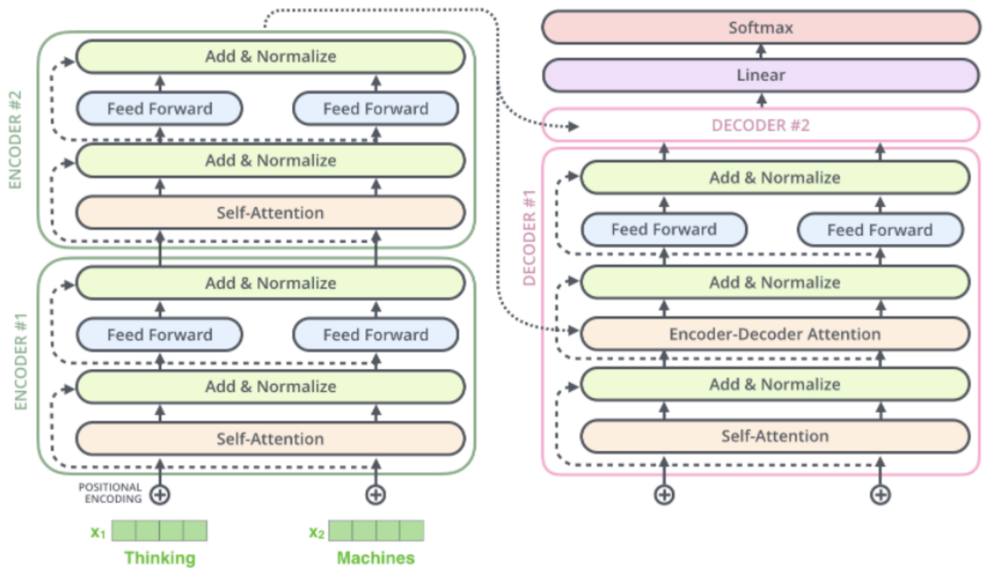


Figure 4.3: Text to text Transfer Transformer (T5) Model

4.2.2 Input-Output Scenario

Here, both the input and output are in text format. The main ideology behind T5 is transfer learning where the model is firstly trained on a large data set before being fine-tuned on a downstream task.T5 uses a sequence-to-sequence method that passes the encoded outputs to the decoders. The encoders take the input of a series of tokens which are further embedded by using predefined embedding methods such as Glove. Encoder and decoder are a 6-layered stack. Encoder and decoder have quite similar layers namely multi-head attention and feed-forward neural network except that decoder have an additional layer called masked multi-head attention. The softmax is an activation function, and the output of the final decoder block will be sent to another layer. This layer's output will be fed into the input embedding matrix.. T5 works on a variety of tasks such as Summarisation and Translation.

4.2.3 Framework

Many of the NLP tasks are formulated using " text to text " transformer.Tthe main idea behind this is the model is fed with some text context or conditioning and is then asked to output some text this similar idea can be seen in some of the tasks such as "WMT1" task which is translating English to german, "CoLA" is a classification system that determines whether or not a statement is grammatically correct., "STSB" [25] task is to Classify the sentiment of a sentence on a scale from 1 to 5 (21 Sentiment Classes) "Summarisation" i.e summarising text into shorter representation, "SQuAD" task which is Answering the question in the given context and many more.The framework of the t5 transformer is shown in Figure 4.4

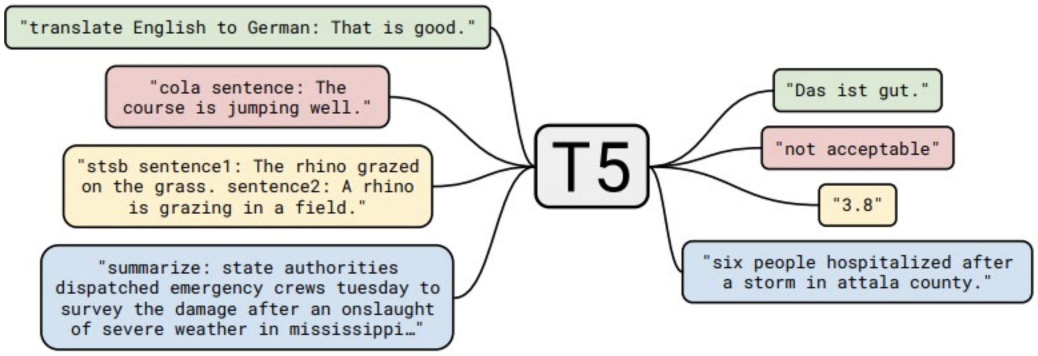

"translate English to German: That is good."
"cola sentence: The course is jumping well."
"stsb sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."
"summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi…"
T5
"Das ist gut."
"not acceptable"
"3.8"
"six people hospitalized after a storm in attala county."

Figure 4.4: Text to text Transfer Transformer (T5) Framework

4.2.4 T5 In-Detail

On a high-level overview as shown in Figure 4.5, T5 consists of an encoder block and a decoder block. The encoder block is a stack of 6 encoders and the decoder block is a stack of 6 decoders.

The word embeddings is the first encoder's input (vector representation of word). The output of the prior encoders will be the inputs of the subsequent encoders. Every decoder block will get the output from the final encoder. The output from the decoder will be given to linear and softmax blocks and mapped to the embeddings to produce the text output.

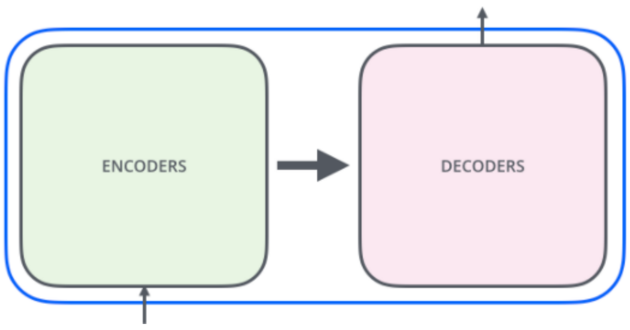


Figure 4.5: High Level Overview

4.2.5 Embedding

The very first step in most of the NLP applications is converting the words into vectors by using any embedding algorithms.After embedding, each word will be transferred to its matching vector. Each embedding is 512 bytes in size. As a result, the first encoder receives word embeddings of the size, or sentence length. After embedding the words in our input sequence, each one passes through each of the encoders.The graphical representation of word embeddings is shown in Figure 4.6

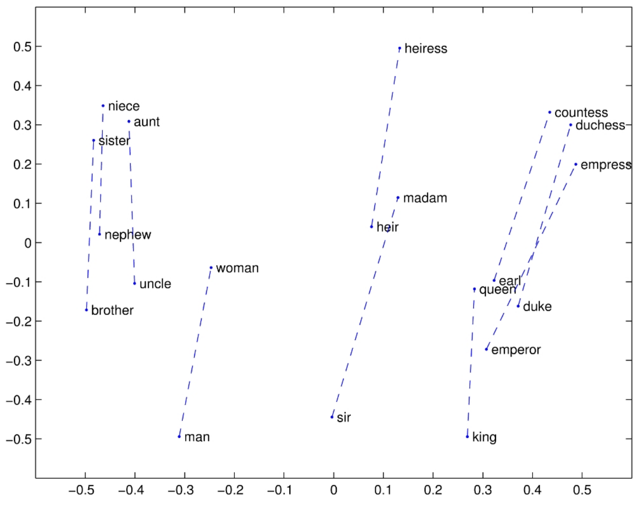


Figure 4.6: Glove Embeddings Graph

4.2.6 Positional Embedding

Positional embeddings are really important in transformer architecture. Positional embedding has information about both the meaning and position of the word. The normal embedding will be added to positional word embedding before sending it to the encoder block as represented in Figure 4.7.

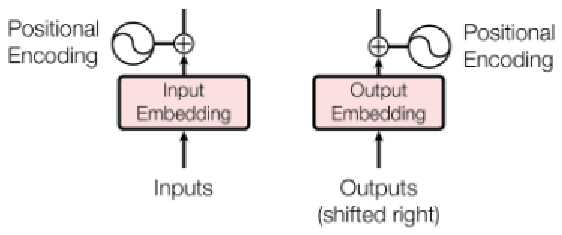


Figure 4.7: Positional Encoding

4.2.7 Self Attention Procedure

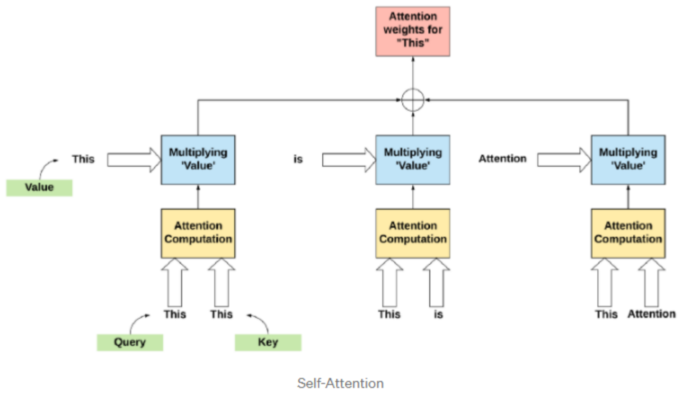


Figure 4.8: Self Attention

Overview of self attention is shown in Figure 4.8

- **Creating 3 vectors Q, K, V:** We make a Query vector (Q), a Key vector (K), and a Value vector (V) for each word (V). To obtain these vectors, the embedding is multiplied by three weight matrices Wq, Wk, and Wv.

- **Calculating Score:** This given word must be scored against each word in the input sentence. This score indicates how important one word is to the other. A dot product of the query vector and the key vector of the word being scored yields the score.. The dot product of q1 and k1 would be the first score of the first word. The dot product of q1 and k2 would be the second score of the first word as shown in Figure 4.9.

- **Dividing the scores by 8:** This is done in order to get more stable gradients

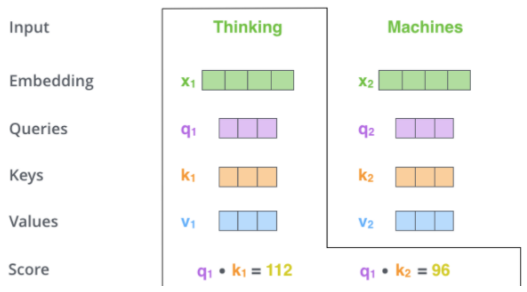


Figure 4.9: Score

- **Softmax:** The The values are reduced to a [0, 1] probability distribution using the softmax algorithm as shown in Figure 4.11.
- **Multiplying with value vector:** Multiply each value vector of the word by the softmax score
- **Summing up :** Add all the weighted value vectors together.For the first word, this would result in the output of the self-attention layer.This is demonstrated in Figure 4.10

appending word embeddings as rows of the matrix, the sentence is compressed into a two-dimensional space.

- **Calculating Q, K, V matrices:** We can get the Q, K, V matrices by multiplying the sentence matrix with the weighted matrices(WQ, WK, WV).
- **Calculating Z:** Using the formula shown in Figure 4.11, we can compute z.

Multi-headed(encoder side)

As shown in Fig 16, In multi-headed attention, we have not one, but several sets of Query/Key/Value weight matrices. The Transformer has eight weight

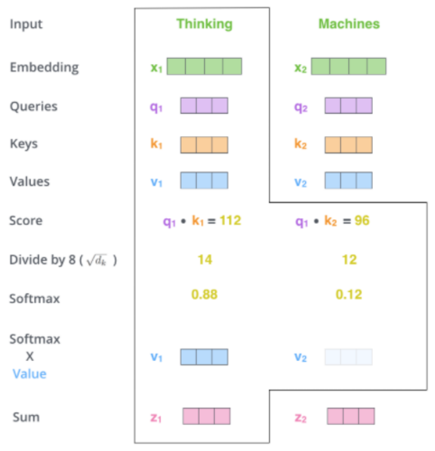


Figure 4.10: Whole Attention

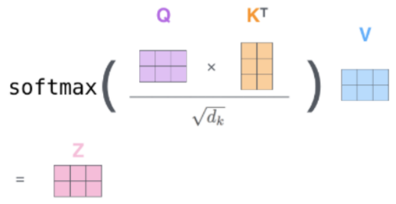


Figure 4.11: Softmax Formula

sets. each of them is randomly initialized. after doing the self-attention calculation, we get 8 Z matrices for each word.

The output is 8 matrices. but the feedforward block only accepts one matrix. So these 8 matrices will be concatenated to one matrix and will be multiplied with weight wo to give the final z as shown in Figure 4.13.

Masked multi-headed(decoder side)

Because the Transformer employs self-attention when constructing target sequences at the decoder, it prefers to include all of the words from the

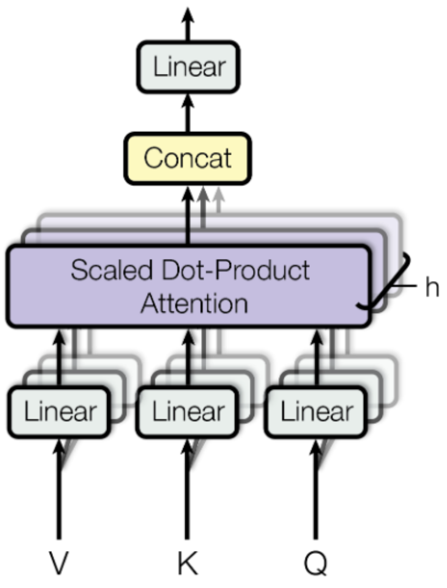


Figure 4.12: Multi Headed Attention

decoder inputs. However, this is inaccurate in practice. Only the words that came before the current word can aid in the creation of the following term. This is ensured through masked multi-head attention as shown in Figure 4.12. This is done by masking(padding to Zero) the later words in the matrix.

4.2.8 Feed Forward Encoder-Decoder

The feed-forward neural networks convert the attention vectors into a form that is acceptable by encoder/decoder block.

Encoder

- as shown in Figure 4.14, Every word will be passed into pre-trained embedding algorithms which give 512 sized vectors as output. This vector will be added to the positional encoding of the same before

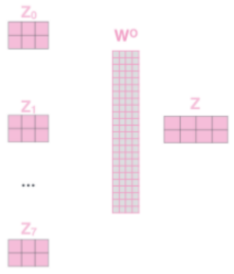


Figure 4.13: Concate

passing it to the encoder block.

- The vector will pass through each encoder's multi-head attention and feed-forward block.

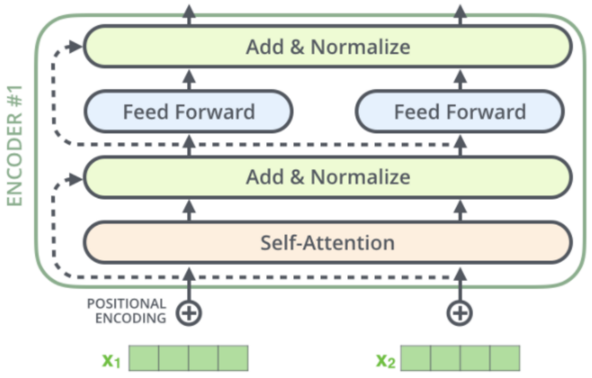


Figure 4.14: Encoder

- The Residuals: In Figure 4.15, Each encoder has a residual connection around each sub-layer (attention/feedforward), which is followed by a layer-normalization step.

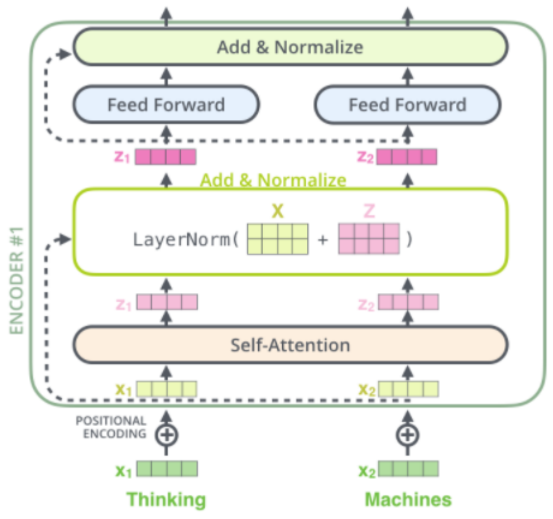


Figure 4.15: Residual

Decoder

- As shown in Figure 4.16 The top encoder's output is then translated into a set of K and V attention vectors. Each decoder will make use of these.
- We embed and applied positional encoding to those decoder inputs to determine the position of each word.
- In the next time step, the output of each step is supplied to the bottom decoder, and in the same way that the encoders did, the decoders bubble up their decoding findings.
- The decoder's self-attention layers work in a somewhat different way from the encoder's i.e, Only earlier points in the output sequence are allowed to be prioritised by the decoder's self-attention layer. This is accomplished by masking future positions (putting them to -inf) before the self-attention calculation's Soft-max stage.

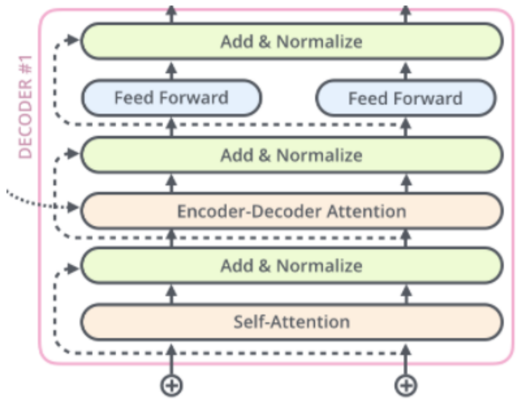


Figure 4.16: Decoder

- In the same way as Multi-headed self-attention gets its Queries matrix from the layer below it, the "Encoder-Decoder Attention" layer gets its Keys and Values matrix from the encoder stack's output.

4.2.9 Linear and Softmax Layer

decoder block's output would be a set of float vectors. This layer will be in charge of converting the vectors into words. The Linear layer is a simple fully connected neural network that converts the decoder's block's vector into a much larger vector known as a logits vector. The cells of unique words will be mapped to the logits vector.

The softmax layer then converts these scores to probabilities. For this time step, the cell with the highest probability is chosen as the output, and the word associated with it is generated.

Chapter 5

Experimental Results

The given sample input - output text is a single unit data from the music dataset. This is a random review of a guitar product from Amazon. The dataset contains thousands of such reviews.

The 'model summary' predicts the short abstractive summary for the text . Here, since the cost is mentioned in the text/ review with almost positive qualities, the summary model responds with affordable price by comparing both the cost and quality of the product.The 'model summary' generates the summary by considering the key words and phrases it's own short sentences.

Figure 5.1 shows output summary given by our model for a guitar review.

Figure 5.2 shows the output for a news summary article.

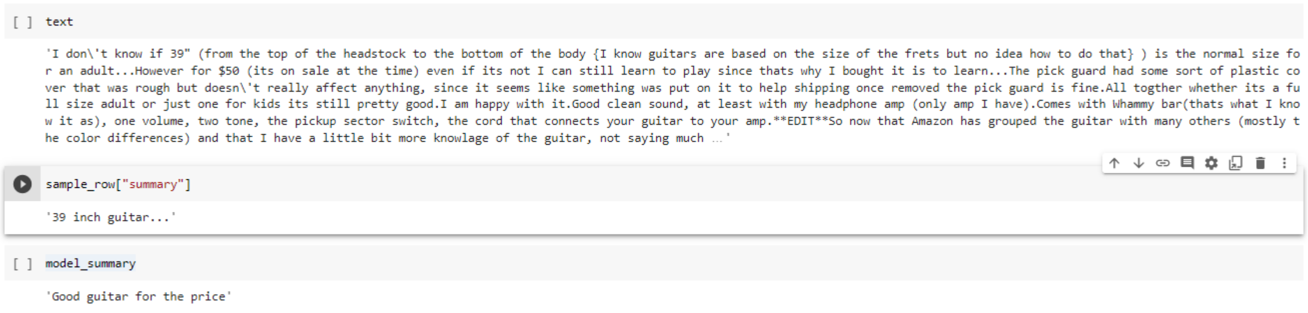
The 'model summary' predicts the short abstractive summary for the text. The model emphasises the importance of the speech and includes the important sentences as it is. Since both the date of public holiday and the reason for the same are equally important, it focuses on them and summarises them by phrasing into simpler sentences.

for Figure 5.3 The given sample input - output text is a single unit data from the reviews dataset. This text is a random review from products in Amazon.

This sample input is about a earphones product from Amazon. The text here explains how good the product is, by highlighting the battery life and quality.

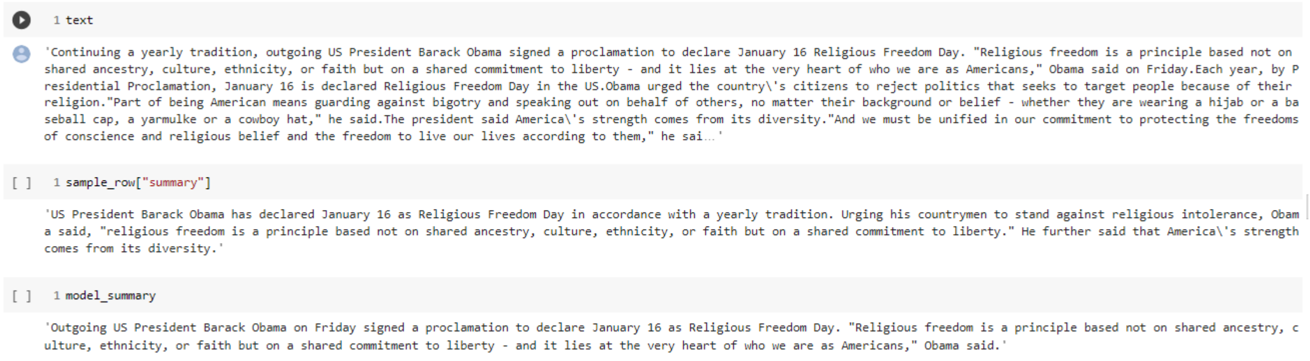
The ‘model summary’ predicts the short abstractive summary for the text. Since, the text highlights that the product is good in terms of quality, and battery life, the model summarises it as a good product. Determining it as good / bad by the model is quiet essential as users have a normal tendency to look after the better or not review and then get into further details.

5.1 Diagrams



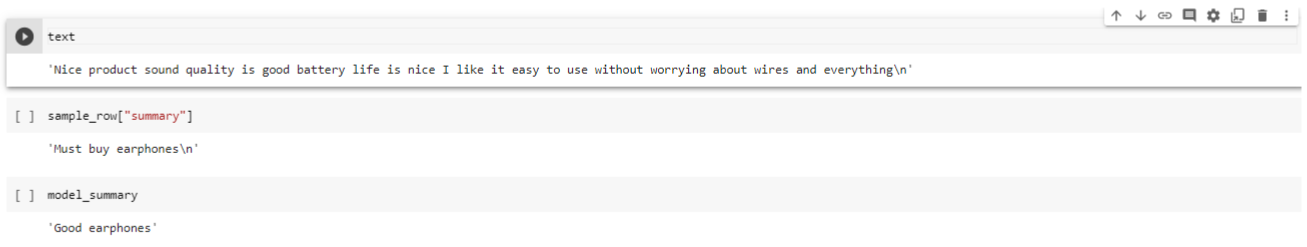
```
[ ] text

'I don\'t know if 39" (from the top of the headstock to the bottom of the body {I know guitars are based on the size of the frets but no idea how to do that} ) is the normal size fo
r an adult...However for $50 (its on sale at the time) even if its not I can still learn to play since thats why I bought it is to learn...The pick guard had some sort of plastic co
ver that was rough but doesn\'t really affect anything, since it seems like something was put on it to help shipping once removed the pick guard is fine.All togther whether its a fu
ll size adult or just one for kids its still pretty good.I am happy with it.Good clean sound, at least with my headphone amp (only amp I have).Comes with Whammy bar(thats what I kno
w it as), one volume, two tone, the pickup sector switch, the cord that connects your guitar to your amp.**EDIT**So now that Amazon has grouped the guitar with many others (mostly t
he color differences) and that I have a little bit more knowlage of the guitar, not saying much ...'

sample_row["summary"]

'39 inch guitar...'

[ ] model_summary

'Good guitar for the price'
```

Figure 5.1: Results for music dataset



```
1 text

'Continuing a yearly tradition, outgoing US President Barack Obama signed a proclamation to declare January 16 Religious Freedom Day. "Religious freedom is a principle based not on
shared ancestry, culture, ethnicity, or faith but on a shared commitment to liberty - and it lies at the very heart of who we are as Americans," Obama said on Friday.Each year, by P
residential Proclamation, January 16 is declared Religious Freedom Day in the US.Obama urged the country\'s citizens to reject politics that seeks to target people because of their
religion."Part of being American means guarding against bigotry and speaking out on behalf of others, no matter their background or belief - whether they are wearing a hijab or a ba
seball cap, a yarmulke or a cowboy hat," he said.The president said America\'s strength comes from its diversity."And we must be unified in our commitment to protecting the freedoms
of conscience and religious belief and the freedom to live our lives according to them," he sai...'

[ ] 1 sample_row["summary"]

'US President Barack Obama has declared January 16 as Religious Freedom Day in accordance with a yearly tradition. Urging his countrymen to stand against religious intolerance, Obam
a said, "religious freedom is a principle based not on shared ancestry, culture, ethnicity, or faith but on a shared commitment to liberty." He further said that America\'s strength
comes from its diversity.'

[ ] 1 model_summary

'Outgoing US President Barack Obama on Friday signed a proclamation to declare January 16 as Religious Freedom Day. "Religious freedom is a principle based not on shared ancestry, c
ulture, ethnicity, or faith but on a shared commitment to liberty - and it lies at the very heart of who we are as Americans," Obama said.'
```

Figure 5.2: Results for news summary dataset



```
text
'Nice product sound quality is good battery life is nice I like it easy to use without worrying about wires and everything\n'

[ ] sample_row["summary"]
'Must buy earphones\n'

[ ] model_summary
'Good earphones'
```

Figure 5.3: Results for earphone review dataset

	ROUGE-1	ROUGE-2	ROUGE-3
F1	0.473	0.265	0.361
Precision	0.467	0.261	0.338
Recall	0.480	0.269	0.389

Table 5.1: Results on news summary using T5 Model

5.2 Accuracy Tables

The following tables displays the accuracy rates for the datasets considering the scores of F1, Recall and Precision using the T5 Framework model that has been implemented

The table 5.1 displays the accuracy rates for the datasets considering the scores of F1, Recall and Precision using the Seq2Seq model. Rouge 1, Rouge 2 and Rouge 3 values have been taken into consideration. The accuracy here ranges from 20% to 38%.

The table 5.3 displays the accuracy rates for the datasets considering the scores of F1, Recall and Precision using the Bert model. Rouge 1, Rouge 2 and Rouge 3 values have been taken into consideration. The accuracy here ranges from 60% to 70%.

	ROUGE-1	ROUGE-2	ROUGE-3
F1	0.312	0.194	o.264
Precision	0.388	0.274	0.288
Recall	0.324	0.1329	0.199

Table 5.2: Results on news summary using Seq2Seq Model

	ROUGE-1	ROUGE-2	ROUGE-3
F-Bert	0.672	0.665	o.662
P-Bert	0.668	0.660	0.638
R-Bert	0.680	0.668	0.62

Table 5.3: Results on news summary using bert Model

Models	ROUGE-1	ROUGE-2	ROUGE-3
Pipeline-BART	0.37	0.28	o.38
BART modified	0.40	0.274	0.42
T5	0.46	0.33	0.43
Pegasus	0.42	0.28	0.40

Table 5.4: evaluation and comparision of ROUGE values of different transformer models

Chapter 6

Conclusion

Our work deals with the implemenation of a model for abstractive text summarisation. The work uses deep learning in order to increase the efficiency. Decreased train loss is also a major contribution in this the sequence to sequence model.

In this Report, we discussed how exactly abstractive text summarisation is a challenging task, yet much needed in present times.We clearly specified the problem definition with a brief input-output statement along with a pictorial representation. Also, we've mentioned most closely related works that has been done by researches before, that were based on our problem. It follows by section which provides a brief idea on the basics of Neural Networks. It further explains the advancement in Neural Networks. Feed forward neural networks and RNN are explained later on. LSTM is finally explained with the internal mechanism. Later on, motivation behind the NLP and LSTM is given. We went on further explaining the design and the flow graph. The entire preprocessing we did in implementation is explained which includes cleaning and encoding of data. Results along with tabular representation are provided.

Considering our current work, we came to a conclusion that using the current T5 Framework would give the summarisation with around 70-80% of accuracy. The work has been tested with around 7 datsets which includes news reports, music rewiews, product reviews, medical reports, etc.

References

[1] Ekaterina Zolotareva, Tsegaye Misikir Tashu and Toma´s Horv ˇ ath ELTE- ´ Eotv ¨ os Lor ¨ and University, Faculty of Informatics, Department of Data ´ Science and Engineering, Telekom Innovation Laboratories Pazm´ any ´ Peter s ´ et´ any 1/C, 1117, Budapest, Hungary (dnbo45, tomas.horvath , ´ misikir)@inf.elte.hu

[2] Ramesh Nallapati et al. "Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond". In: Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 280–290

[3] Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. ¨ Neural computation, 9(8):1735–1780, 1997.

[4] Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, Zhou Y, Li W, Liu PJ (2019) Exploring the Limits of Transfer Learning with a Unified Textto-Text Transformer. CoRR abs/1910.10683:

[5] Zhang J, Zhao Y, Saleh M, Liu PJ (2019) PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. CoRR abs/1912.08777

[6] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memorynetworks for machine reading. arXiv preprint arXiv:1601.06733, 2016

[7] Junyoung Chung, C¸ aglar Gulc¸ehre, Kyunghyun Cho, and Yoshua Ben- ¨ gio. Empirical evaluation of gated recurrent neural networks on sequence modeling. CoRR, abs/1412.3555, 2014

[8] Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. arXiv preprint arXiv:1705.04304, 2017

[9] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. arXiv preprint arXiv:1703.03130, 2017

[10]] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, Advances in Neural Information Processing Systems 28, pages 2440–2448. Curran Associates, Inc., 2015.

[11] Devlin J, Chang M-W, Lee K, Toutanova K (2018) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. CoRR abs/1810.04805:

[12] Radford A (2018) Improving Language Understanding by Generative PreTraining

[13] Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. Exploring the limits of language modeling. arXiv preprint arXiv:1602.02410, 2016.

[14] Nomoto T (2005) Bayesian Learning in Text Summarization

[15] Babar S, Tech-Cse M, Rit (2013) Text Summarization:An Overview

[16] Lewis M, Liu Y, Goyal N, Ghazvininejad M, Mohamed A, Levy O, Stoyanov V, Zettlemoyer L (2019) BART: Denoising Sequence-toSequence

Pre-training for Natural Language Generation, Translation, and Comprehension. CoRR abs/1910.13461:

[17] Ankur Parikh, Oscar Tackstr ¨ om, Dipanjan Das, and Jakob Uszkoreit. ¨ A decomposable attention model. In Empirical Methods in Natural Language Processing, 2016.

[18] Dragomir R Radev, Eduard Hovy, and Kathleen McKeown. 2002." Introduction to the special issue on summarization,"Computational linguistics 28, 4 (2002)

[19] Christian H, Agus M, Suhartono D (2016) Single Document Automatic Text Summarization using Term Frequency-Inverse Document Frequency (TFIDF). ComTech: Computer, Mathematics and Engineering Applications 7:285 . https://doi.org/10.21512/comtech.v7i4.3746

[20] Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. Unifying question answering and text classification via span extraction. arXiv preprint arXiv:1904.09286, 2019b.

[21] Dragomir R Radev, Eduard Hovy, and Kathleen McKeown. 2002." Introduction to the special issue on summarization,"Computational linguistics 28, 4 (2002)

[22] Natural Language Processing Market — 2022 - 27 — Industry Share, Size, Growth - Mordor Intelligence

[23] llahyari, Seyedamin Pouriyeh and Mehdi Assef "Text Summarization Techniques: A Brief Survey".

[24] Attention Is All You Need Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin

[25] https://towardsdatascience.com/hands-on-googles-text-to-text-transfer-transformer-t5-with-spark-nlp-6f7db75cecff?gi=ae6f13491c99

[26] https://arxiv.org/pdf/1609.04747.pdf

[27] GloVe: Global Vectors for Word Representation Jeffrey Pennington, Richard Socher, Christopher D. Manning Computer Science Department, Stanford University, Stanford, CA 94305 jpennin@stanford.edu, richard@socher.org, manning@stanford.edu

[28] http://jalammar.github.io/illustrated-transformer/