Fall 2022 ADVANCED MACHINE LEARNING (MIS-64061-003)

Final Project

Text Summarization using different models

Submitted by: Riba Khan

M.S. in Business Analytics, Kent State University

Abstract

The main objective of the project is to perform **text summarization** on news articles using five different models as mentioned in the report further. I further used **ROUGE metrics** for evaluating the generated summary and analyzing which models perform the best result. After careful analysis, we found out that Luhn's Heuristic Method and Text rank scored very similarly and could have performed differently on different input documents. In terms of readability, the Transformer model produced the best summary as it closely resembles the human-made summary which makes sense because nowadays transformer models are the most frequently used for text summarization. Therefore, further, in my project, I applied the transformer model to a real-world news dataset using the TensorFlow API. The generated summary is mentioned in the report. In conclusion, I can state that considering the short training period and the small amount of data, the results seem to be much better than anticipated.

Keywords: Text Summarization, Transformers, ROUGE Metrics, TensorFlow.

Introduction: Summarization is the ability to explain a larger piece of literature in short and cover most of the meaning the context addresses.

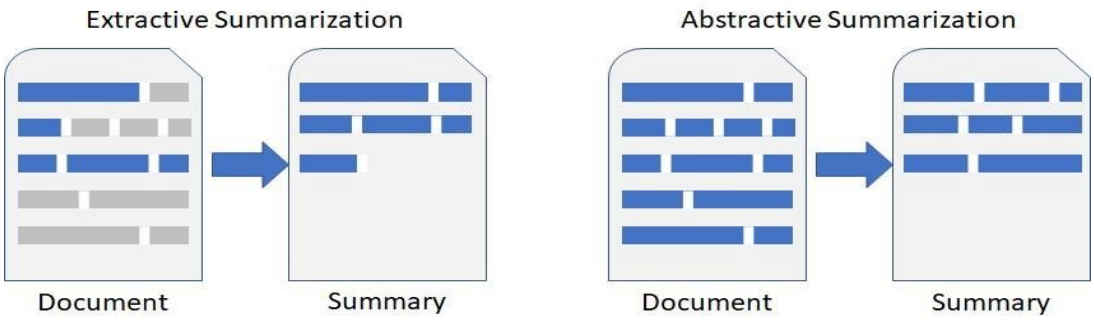
The way we communicate is changing as technology advances quickly throughout the world. Because we may now communicate via text, audio, or video chat, the volume of content produced each day is rapidly increasing. Having to read a lot of texts to get the gist of what they need to know might make it difficult for people to keep up with what they need to know, which is the problem with all this knowledge.

The average person spends roughly two hours each week reading stuff online, according to a Google survey. (https://www.businessnewsdaily.com/4718-weekly-online-social-media-time.html). Making ensuring you spend time on the appropriate things is essential with so much information available. One of those solutions is text summarization. It enables you to read something significant without having to devote a lot of time to it. You'll do this to save time and energy and to lower your stress levels. With machine learning making a lot of progress now text summarization is becoming much easier and every day new tools are introduced to make the process convenient.

There can be two types of text summarization:

1) ***Extractive Summarization***: It means selecting a certain element (often sentences) based on specified weights assigned to the keywords, with the choice of text being determined by the weights of the words inside it.
2) ***Abstractive Summarization*****:** In contrast, Abstractive Summarization uses heuristic techniques to teach the system to try and comprehend the entire context before producing a summary based on that understanding.

In this project, we will be dealing with abstractive summarization.



Abstractive summarization models typically require more processing resources. This enables them to create grammatically and contextually sound comments that are correct in the field being discussed. The model must first completely understand the original content in order to be able to summarize it accurately and meaningfully. As a result, extractive summarization techniques are frequently used nowadays. Finding the key sentences that must be included in the summary is the only objective that extractive summarization models need to focus on. Before they can generate an adequate summary, models for abstractive summarization need to take much more information into account.

The goal of the project

- Perform abstractive text summarization using five different summarization models and then evaluate the results.
- Then using the best technique to apply to a larger dataset. (Inshorts Dataset)

Now let us talk about different models that I will be using for comparing my text summaries. ***The list of text summarization models is as follows***:

- **Luhn's Heuristic Method**
- **Text Rank**
- **Latent Semantic Analysis (LSA)**

- **Kullback-Leibler Sum (KL-Sum)**
- **Transformer Model**

Let us understand each of these models briefly:

- ***Luhn's Heuristic Method :***

Luhn's Heuristic Method for Text Summarization is one of the first Text Summarization algorithms, being published in 1958. It is based on TF-IDF (Term Frequency-Inverse Document Frequency), and selects words of high importance based on their frequency of occurrence. Also, higher weightage is given to the words that occur at the beginning of the document.

- ***TextRank:***

It is used to find the most relevant sentences (as well as keywords) in a piece of text. Here, sentences that contain highly frequent words are considered important. Thus, the algorithm assigns scores to each sentence in the source material. The sentences are then ranked in descending order of their scores, and the top-scoring sentences are included in the summary.

- ***Latent Semantic Analysis:***

It is an unsupervised Natural Language Processing (NLP) technique. The goal is to identify the most important topics from the source material and then choose the sentences with the greatest combined weights across the topics.

- ***Kullback-Leibler Sum (KL-Sum):***

K-L sum algorithm (**Kullback-Lieber (KL) Sum algorithm**) for **text summarization** which focuses on minimization of summary vocabulary by checking the divergence from the input vocabulary. It is a **sentence selection algorithm** where a target length for the summary is fixed (L

words). The objective of KL sum algorithm is to find a set of sentences whose length is less than L words and the *unigram* distribution is as similar to the source document

- **Transformers :**

Transformers are a type of neural network architecture and were developed by a group of researchers at Google (and UoT) in 2017. They avoid using the principle of recurrence and work entirely on an attention mechanism to draw global dependencies between the input and the output.

I used the following article to perform text summarization using the above models :

"Flames and a large plume of smoke could be seen in the Kent area Friday as crews worked to extinguish a massive re in the Historic Mill District. There broke out at Star of the West Milling Co. building at 162 N Water Street just before 9 a.m. Kent Fire Chief Bill Myers said that 3 minutes after firefighters arrived at the scene, there was an explosion. It's not clear what caused it. The structure of the building was compromised after the explosion, so crews couldn't go inside it. They worked on extinguishing the re from the outside, the re chief said. Chief Myers said they are also having some water supply issues. According to the chief, getting the re was taxing the water supply, so they were trying to reduce their use. The chief said re-operations could stretch into Saturday. No one was hurt in the re. It's unclear how the building was currently being used. The chief said no evacuations were issued. Kent State University issued an alert saying there is no threat to the campus but to avoid the downtown area. The cause of the re is unknown. Named after the Williams Brothers in 1879, the Mill District of Kent, Ohio says the mill once produced hundreds of millions of pounds of ours. The iconic grain elevators are a well-known fixture in the city's skyline. "

Now the question arrives, as to how we will evaluate the generated summaries.

For evaluating our text summaries we will be using ROUGE Metrics.

ROUGE, which stands for Recall-Oriented Understudy for Gisting Evaluation, is a metric that is used to obtain the degree of similarity between a candidate summary (automatically generated summary) and the target summary (hand-written reference summary.

ROUGE scores are further subdivided into ROUGE-1, ROUGE-2, and ROUGE-L scores.

ROUGE 1 : The similarity of uni-grams between reference and candidate summaries is compared using the ROUGE-1 Precision and Recall test. Simply put, we mean by "uni-grams" that each token of comparison is a single word.

ROUGE-2: In ROUGE-2 Precision and Recall, bi-gram similarity between reference and candidate summaries is compared. Bi-grams are two words from the reference and candidate summaries that are used as comparison tokens.

After successfully implementing all the models these were the score that we obtained.

	EVALUATION METRIC	Luhn's Heuristic Method	Text Rank	Latent Semantic Analysis	Kullback-Leibler Sum	T5 Transformer Model
ROUGE 1	Precision	1.0	1.0	1.0	1.0	0.844
	Recall	0.58	0.59	0.53	0.42	0.34
	F-Score	0.74	0.74	0.69	0.59	0.48
ROUGE 2	Precision	0.98	1.0	0.98	0.98	0.67
	Recall	0.50	0.51	0.44	0.37	0.23
	F-Score	0.66	0.68	0.61	0.54	0.34
ROUGE L	Precision	1.0	1.0	1.0	1.0	0.82
	Recall	0.58	0.59	0.53	0.42	0.33
	F-Score	0.74	0.74	0.69	0.59	0.47

What we can interpret from the table is :

Findings and Conclusions:

1) Luhn's Heuristic Method and Text rank scored very similarly and could have performed differently on different input documents.
2) In terms of readability, the Transformer model produced the best summary as it closely resembles the human-made summary. On the contrast, it is shocking to see that the F1 score of the transformers model is really low but the summary generated is conceptually great and covers most part of the original text.
3) The summary produced by Luhn's Heuristic Method in terms of readability is worse because it is one of the earliest Text summarization models to come into the picture.

Since, according to my understanding transformer model performs the best. I will again implement it to see how the predicted summary turns out to be. I will be using an inshort data set which is a news dataset to perform text summarization using transformers.

Output generated by Transformer

```
print(summary)
```

```
the fire broke out at the Star of the West Milling Co. building at 162 N Water Street just before 9 a.m. it's unclear how the building was currently being used. no one was hurt in
```

Output: the fire broke out at the Star of the West Milling Co. building at 162 N Water Street just before 9 a.m. it's unclear how the building was currently being used. no one was hurt in the blaze, and no evacuations have been issued for the historic mill district of Kent, ohio, where the mill once produced hundreds of millions of pounds of flour. the iconic grain elevators are well-known fixture in city's skyline.

Text summarization using Transformers using a large dataset this time.

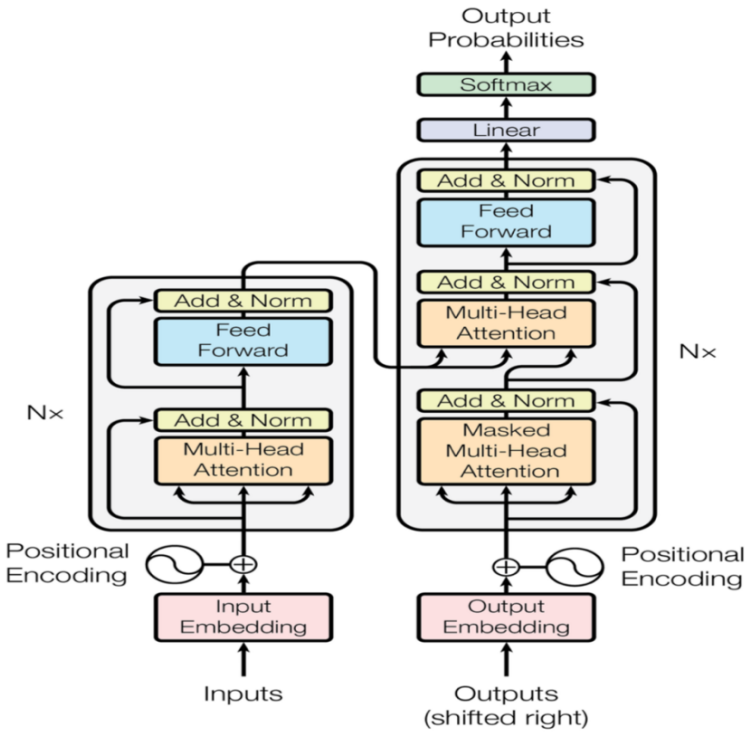


Figure 1: The Transformer - model architecture.

What are Transformers?

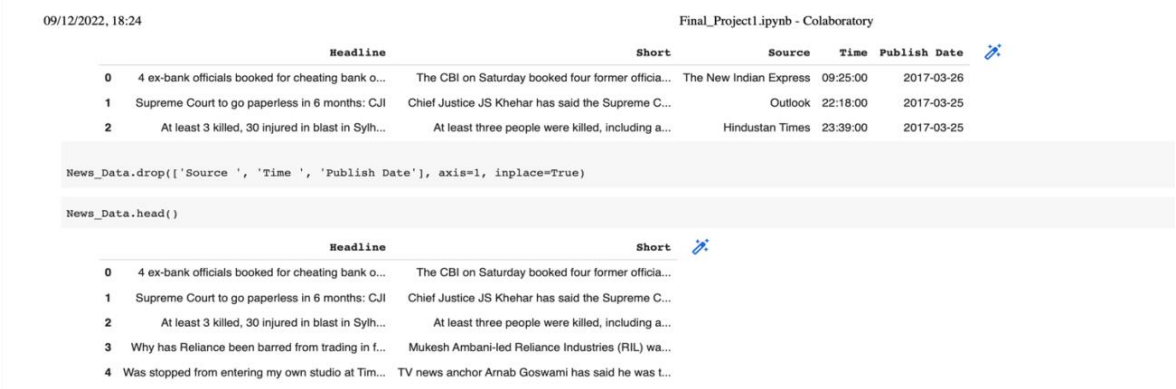
Transformers are a subset of artificial neural networks that are employed in deep learning applications to address the issue of transduction, or the translation of input sequences into output sequences. It is used primarily in the fields of natural language processing (NLP) and computer vision (CV). A novel architecture called NLP's Transformer aims to solve problems sequentially while resolving long-distance dependencies with ease. Without the use of convolutions or sequence aligned RNNs, the input and output representations are computed exclusively utilizing self-attention. The above picture accurately depicts the steps I have followed in developing my model.

Techniques Used:

I used the following techniques that were taught in class for this project. Steps followed:

1. ***Loading the dataset***

2. ***Removing unnecessary columns***



3. ***Pre-Processing***

Here I performed the basic preprocessing that is required to train any NLP model. For recognizing the start and end of target sequences, we pad them with start (“<SOS>”) and end (“<EOS>”) tokens.

I applied a tokenizer to the sequences that essentially removes punctuation, changes the text's case to lower case, keeps track of a vocabulary dictionary that is arranged according to a word's frequency of occurrence and maps each word to its token equivalent, and finally transforms the textual data into tokens that can be directly entered into the model. used the Tokenization API for TensorFlow. It essentially takes care of all aspects of data preparation and cleaning. To provide the model a more universal input, there is still one more step that involves padding or truncating the sequences to a predetermined length.

Finally, I batch and shuffle e the data to make it simple to obtain it for model training. I accelerate the calculation of these operations using the TensorFlow Dataset API.

Batch Size = 64

Buffer size = batch size * 8

4. Postal Encodings

These procedures are in charge of getting the input sequences' positional encodings. Positional Encodings essentially create an idea of ordering among the input words because the Transformer's self-attention mechanism disregards it.

5. Masking

These processes oversee obtaining the positional encodings of the input sequences. The Transformer's self-attention mechanism ignores the input words' order; therefore, Positional Encodings basically construct that order.

6. The model

Now we arrive at the construction of the model

Scaled Dot-Product

This is one of the most important steps in building the Transformer as this is the base for attention computation in the model.

Multi Head Attention

This is a typical two-layered, after practically every sub-layer, the feed-forward network that is utilized in the same way. As a TensorFlow Custom Layer, Multi-Head Attention is what we refer to. Here, we divide the inputs into various heads, calculate the attention weights using scaled dot-product attention, and then combine the output from all the heads.

7. Enocder and Decoder Blocks

These serve as the basic building blocks of the encoder and decoder, respectively. While fine-tuning the model, these could be expanded into Nx encoder/decoder layers.

It is the responsibility of this layer to interact directly with inputs and outputs. In this instance, the inclusion of the input embeddings improves the positional encodings.

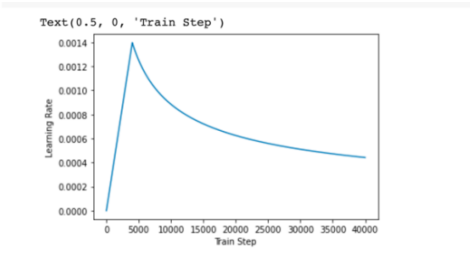
8. Adding Layers to the "Model"

In a Custom Model class that is descended from the tf. keras.Model class, finally stack all the intermediate layers.

The output layer with vocab size units is added to the Decoder. When training, the output of this class will be used for either loss computation or inference.

9. Custom Learning Rate :

Custom learning rate scheduler helps faster convergence. Using adam optimizer.



10. Training the model

Following are the values of the hyperparameters that I will be using while training the summarizer.

- ***Training Parameters***

Number of layers = 3

Epochs = 15

Dropout rate = 0.2

Batch size = 64

Buffer size = batch size * 8

Encoder maxlen = 100

Decoder maxlen= 20

Encoder vocab size = 76362

Decoder vocab size = 29661

Here is a brief excerpt from the verbose of a few trainings' last epochs.

```
Epoch 15 Batch 0 Loss 3.9455 Accuracy 0.3311
Epoch 15 Batch 100 Loss 3.3397 Accuracy 0.3322
Epoch 15 Batch 200 Loss 3.2523 Accuracy 0.3334
Epoch 15 Batch 300 Loss 3.1952 Accuracy 0.3347
Epoch 15 Batch 400 Loss 3.1637 Accuracy 0.3360
Epoch 15 Batch 500 Loss 3.1144 Accuracy 0.3375
Epoch 15 Batch 600 Loss 3.0732 Accuracy 0.3391
Epoch 15 Batch 700 Loss 3.0342 Accuracy 0.3407
Epoch 15 Batch 800 Loss 3.0000 Accuracy 0.3423
Saving checkpoint for epoch 15 at checkpoints/ckpt-3
Epoch 15 Loss 2.9811 Accuracy 0.3433
Time taken for 1 epoch: 67.68230056762695 secs
```

The Result

After training and implementing the model, we got the following results attached as a screenshot.

```
article[5]

'<SOS> A new trailer for the upcoming superhero film  Justice League  was released on Saturday. Based on the DC Comi
cs superhero team, the film stars Ben Affleck as  Batman , Gal Gadot as  Wonder Woman , Ezra Miller as  The Flash  a
nd Jason Momoa as  Aquaman . Directed by Zack Snyder, the film is scheduled to release on November 17, 2017. <EOS>'

print("Real Headline : ", summary[5][5:-5],"\n Predicted Summary : ", summarize(article[5]))

Real Headline :   New trailer of  Justice League  released
 Predicted Summary :  new trailer for justice league released

article[43]

'<SOS> Actor Ranbir Kapoor has penned a special letter to Shah Rukh Khan s wife Gauri Khan to thank her for designin
g his home. The letter has been shared by Gauri on Instagram. Gauri, who is an interior designer, has designed Ranbi
r Kapoor s house in Bandra, Mumbai. Ranbir will reportedly move into his new home in the month of October.  <EOS>'

print("Real Headline : ", summary[43][5:-5],"\nPredicted Summary : ", summarize(article[43]))

Real Headline :   Ranbir Kapoor pens special letter to SRK s wife Gauri Khan
Predicted Summary :  ranbir kapoor pens letter to srk s wife

article[12]

'<SOS> Thousands of people on Saturday took to the streets in London to protest against the UK s decision to leave t
he European Union. Demanding continuation of benefits of remaining in the EU, the protesters said that they were the
48% who voted to remain in the EU during the 2016 referendum. The Brexit process is to be initiated on March 29. <EO
S>'

print("Real Headline : ", summary[12][5:-5],"\nPredicted Summary : ", summarize(article[12]))

Real Headline :   Thousands march in London to protest against Brexit
Predicted Summary :  london protests against uk on brexit issue
```

Conclusion:

Transformers models are way more accurate than any other model. The predicted summary matched closely with the actual summary. One more advantage of transformers is that they can process and train on more data in lesser time.

I can conclude by saying that my *contributions* include trying some different text summarization models and learning how to evaluate ROUGE metrics. However, I would like to expand my knowledge further about ROUGE metrics and study more about it because the Rouge scores and summary generated both were very different. I try to experiment a new technique – Rouge metrics as professor encouraged in class to try something different. Later, applied transformers using TensorFlow to put into experiment the techniques such as padding, masking and so on as studied in class.

References

- Source of Dataset :

1) Inshort Dataset : Taken from Kaggle

   About the data: Inshort is a news service that provides a brief summary of news around the web. This dataset contains headlines and a summary of news items along with its source.

   https://www.kaggle.com/datasets/shashichander009/inshorts-news-data

2) News Summary

   Source : Fox 8 News

   Heading : Flames, explosions in Kent mill fire

   By: Cris Belle, Dave Nethers

   https://fox8.com/email-alerts/breaking-news-email-alerts/15re-breaks-out-at-in-kent-mill-district/

- Figure 1: The Transformer and Model Architecture

Source: https://miro.medium.com/max/700/1*BHzGVskWGS_3jEcYYi6miQ.png

Sharma, G. (2021, July 16). *Decide Best Learning Rate with LearningRateScheduler in Tensorflow*. Retrieved from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2021/06/decide-best-learning-rate-with-learningratescheduler-in-tensorflow/

Jagtap, R. (2020, JULY 20). *Abstractive Text Summarization Using Transformers* . Retrieved from https://medium.com/swlh/abstractive-text-summarization-using-transformers-3e774cc42453

Briggs, J. (2021, MARCH 4). *The Ultimate Performance Metric in NLP*. Retrieved from Towards Data Science: https://towardsdatascience.com/the-ultimate-performance-metric-in-nlp-111df6c64460

Chiusano, F. (n.d.). *Two minutes NLP — Learn the ROUGE metric by examples*. Retrieved from https://medium.com/nlplanet/two-minutes-nlp-learn-the-rouge-metric-by-examples-f179cc285499