University of New Haven

Tagliatela college of Engineering

Master's in data science

Document Summarization

Natural Language Processing

DSCI-6004-01

Date: 12/11/2022

By,

Naga Yashwanth Bodduluri

Thriloknadh kandimalla

Upender Boddu

Supervisor,

Vahid Behzadan

Motivation:

We have selected this project because it can take a person day, or even weeks, to sift through a 50-page technical document, filter out irrelevant material, and write a complete summary of the text without compromising on correctness. When you consider sensitive legal and financial documents, there is no room for error including leaving an important detail out. This is where data science, AI, machine learning, and natural language processing (NLP) text summarization comes in handy.

NLP text summarization is the process of breaking down lengthy text into digestible paragraphs or sentences. This method extracts vital information while also preserving the meaning of the text. This reduces the time required for grasping lengthy pieces such as articles without losing vital information.

Introduction:

"I don't want a full report, just give me a summary of the results". We often found ourselves in this situation – both in college as well as in professional life. We prepare a comprehensive report, and the teacher/supervisor only has time to read the summary.

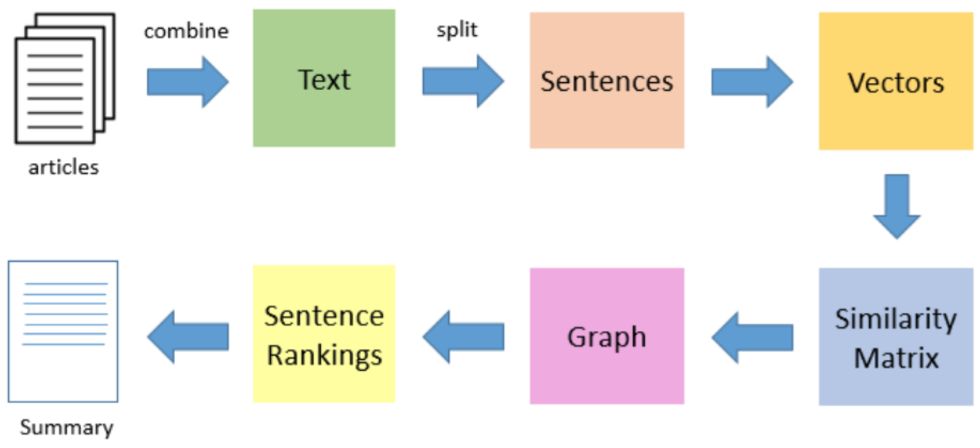
The process of creating a summary from a reasonably long text is known as text summarization. Only a fraction of the actual text's sentences would be used in this summary, but it would still convey the primary idea. Its uses include business analysis, document summarizing, sentiment analysis, and search engine queries. The area of abstract text summarization has seen a lot of research throughout time, particularly with the appearance of pre-trained models suggested by researchers. Automatic text summarization is the task of producing a concise and fluent summary while preserving key information content and overall meaning.

According to Microsoft's Mark McNicholas, co-founder of Netscape, "Automatic text summarization is the job of providing a compact and fluent summary of vital information while keeping critical information substance and overall meaning."

Text summarization :

Text summarization is a very useful and important part of Natural Language Processing (NLP). First let us talk about what text summarization is. Suppose we have too many lines of text data in any form, such as from articles or magazines or on social media. We have time scarcity, so we want only a nutshell report of that text. We can summarize our text in a few lines by removing unimportant text and converting the same text into smaller semantic text form.

In this approach we build algorithms or programs which will reduce the text size and create a summary of our text data. This is called automatic text summarization in machine learning. Text summarization is the process of creating shorter text without removing the semantic structure of text.



Types of Text Summarization:

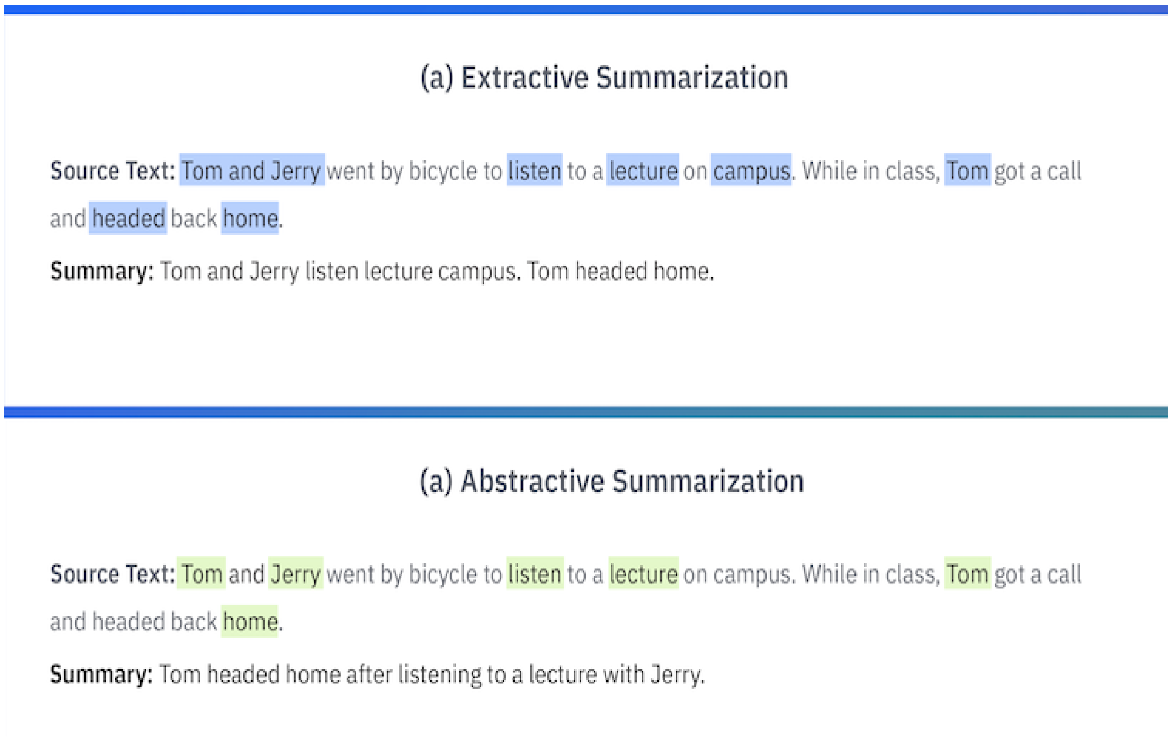
Text summarization methods can be grouped into two main categories: **Extractive and Abstractive methods**

<u>Extractive Text Summarization:</u>

It is the traditional method developed first. The main objective is to identify the significant sentences of the text and add them to the summary. You need to note that the summary obtained contains exact sentences from the original text.

<u>Abstractive Text Summarization:</u>

It is a more advanced method, many advancements keep coming out frequently (I will cover some of the best here). The approach is to identify the important sections, interpret the context and reproduce in a new way. This ensures that the core information is conveyed through shortest text possible. Note that here, the sentences in summary are generated, not just extracted from original text.



(a) Extractive Summarization

Source Text: Tom and Jerry went by bicycle to listen to a lecture on campus. While in class, Tom got a call and headed back home.

Summary: Tom and Jerry listen lecture campus. Tom headed home.

(a) Abstractive Summarization

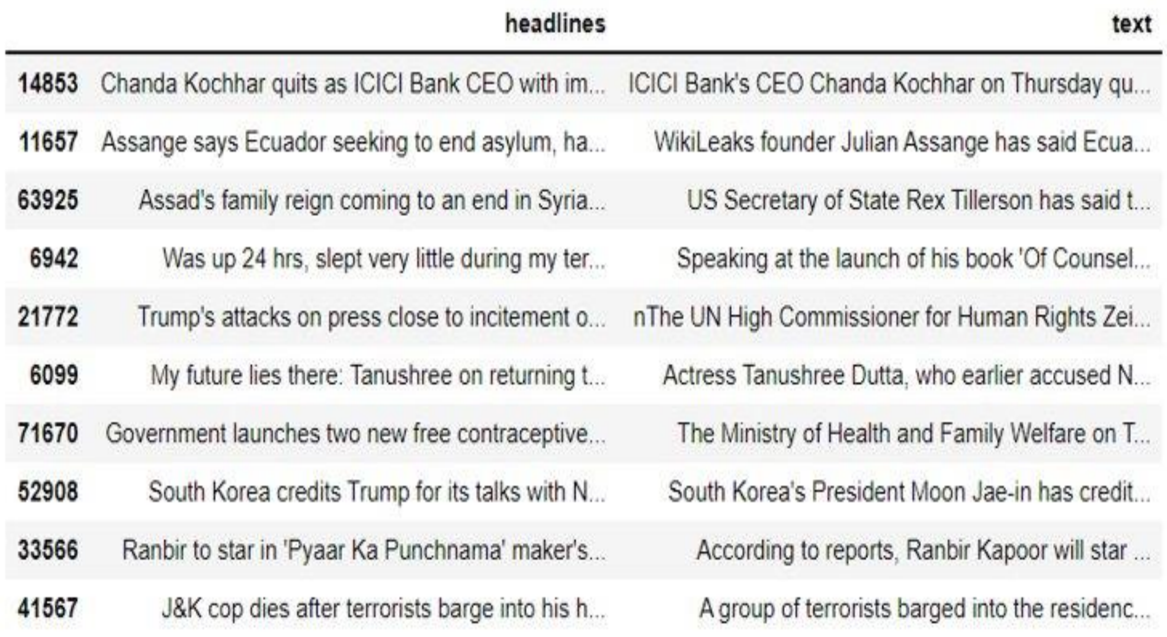
Source Text: Tom and Jerry went by bicycle to listen to a lecture on campus. While in class, Tom got a call and headed back home.

Summary: Tom headed home after listening to a lecture with Jerry.

Describing Data Set:

We have used the dataset new summary articles dataset for training and CNN Daily mail test dataset, which is of size 11490 samples for testing the model and calculating the ROUGE score. These data are human generated abstractive summary bullets were generated from news stories in CNN and Daily Mail websites as questions (with one of the entities hidden), and stories as the corresponding passages from which the system is expected to answer the fill-in the-blank question.

- We create custom datasets from university websites and the data is collected using web scraping
-
- https://drive.google.com/file/d/14Dut_GZrLbLeVAWS3QL861HJ0Zdb5Of2/view



	headlines	text
14853	Chanda Kochhar quits as ICICI Bank CEO with im...	ICICI Bank's CEO Chanda Kochhar on Thursday qu...
11657	Assange says Ecuador seeking to end asylum, ha...	WikiLeaks founder Julian Assange has said Ecua...
63925	Assad's family reign coming to an end in Syria...	US Secretary of State Rex Tillerson has said t...
6942	Was up 24 hrs, slept very little during my ter...	Speaking at the launch of his book 'Of Counsel...
21772	Trump's attacks on press close to incitement o...	nThe UN High Commissioner for Human Rights Zei...
6099	My future lies there: Tanushree on returning t...	Actress Tanushree Dutta, who earlier accused N...
71670	Government launches two new free contraceptive...	The Ministry of Health and Family Welfare on T...
52908	South Korea credits Trump for its talks with N...	South Korea's President Moon Jae-in has credit...
33566	Ranbir to star in 'Pyaar Ka Punchnama' maker's...	According to reports, Ranbir Kapoor will star ...
41567	J&K cop dies after terrorists barge into his h...	A group of terrorists barged into the residenc...

Method:

The goal of this project is to text summarization models based on summaries for one or many documents at a time. In this project, we have implemented the ideas needed to develop a Text Summarizer using Deep Learning and work through the process step-by-step.
We will use a model that already has been pre-trained to generate summaries. This is possible with a modern approach in ML called Transfer Learning. This is another useful step because we basically take a model off the shelf and train it on our dataset. Transfer Learning allows us to create another baseline which will be useful to see what happens when we train the model on our dataset. After that, it is time to

use a pre-trained model and train it on our own dataset. This is also called fine-tuning. It will enable the model to learn from the patterns and idiosyncrasies of our data and slowly adapt to it. Once we have trained the model, we will use it to create summaries.

The model we use for training or fine-tuning is Transformers BART or T-5, of better performance. BART is a sequence-to-sequence model trained as a denoising autoencoder. This means that a fine-tuned BART model can take a text sequence as input and produce a different text sequence at the output.

Google has released the pre-trained T5 text-to-text framework models which are trained on the unlabelled large text corpus called C4 (Colossal Clean Crawled Corpus) using deep learning. C4 is the web extract text of 800Gb cleaned data. The cleaning process involves deduplication, discarding incomplete sentences, and removing offensive or noisy content. T5 (Text-to-Text Transfer Transformer) uses several pretraining objectives, including unsupervised fill-in-the-blank as well as supervised translation, summarization, classification, and reading comprehension tasks where each task is represented as a language generation task (Raffel et al., 2019). T5 closely follows the originally-proposed Transformer architecture (Vaswani et al., 2017) except using relative

positional embeddings rather than sinusoidal encoding. In this work, we used T5-Base, which includes 12 transformer layers.
You can get these T5 pre-trained models from the HuggingFace website:

T5-small with 60 million parameters.
T5-base with 220 million parameters.
T5-large with 770 million parameters.
T5-3B with 3 billion parameters.
T5-11B with 11 billion parameters.

T5 (Text-To-Text Transfer Transformer) is a transformer model that is trained in an end-to-end manner with text as input and modified text as output, in contrast to BERT-style models that can only output either a class label or a span of the input. This text-to-text formatting makes the T5 model fit for multiple NLP tasks like Summarization, Question-Answering, Machine Translation, and Classification problems. Both T5 and BERT are trained with MLM (Masked Language Model) approach. In this project, we work on abstractive summarization using T5-base.

Tools:

For extracting the text summary, we use the following libraries,

- NLTK
- Python3
- Gensim
- Google Colab
- Transformers

- T-5

Training:

Finally, our dataset is ready, we divided the dataset has 80% for training and 20% for testing and we can start training! First, we load the t5-base pretrained model from Hugging face’s repository. Then we can fine-tune it using the transformers.Trainer API, which requires you to set all the hyperparameters in a transformers. TrainingArguments object instance.

we decided to train a T5 small model. I used a batch size of 8 for both train and Val and could train this model. The model was trained for 25 epochs with val_epochs:1, max_source_text_length:512, max_target_text_length:50 and learning_rate:le-4 and showed good improvement in Rouge score and Val loss as the model trained.

Evaluation Methodology:

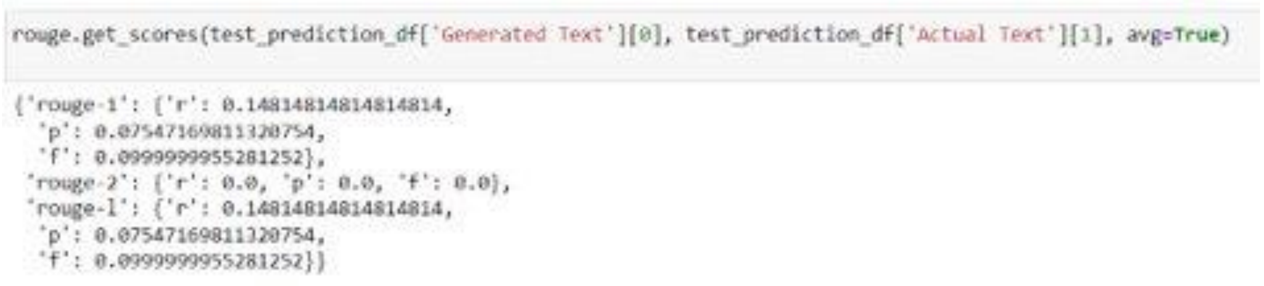
The metric that is used most often in text summarization to measure the quality of a model is the ROUGE score.

ROUGE-N measures the number of matchings ‘n-grams’ between our model-generated text and a ‘reference’.

Evaluation Methodology

The metric that is used most often in text summarization to measure the quality of a model is the ROUGE score.

ROUGE-N measures the number of matchings ‘n-grams’ between our model-generated text and a ‘reference’.



```
rouge.get_scores(test_prediction_df['Generated Text'][0], test_prediction_df['Actual Text'][1], avg=True)

{'rouge-1': {'r': 0.14814814814814814,
  'p': 0.07547169811320754,
  'f': 0.0999999955281252},
 'rouge-2': {'r': 0.0, 'p': 0.0, 'f': 0.0},
 'rouge-l': {'r': 0.14814814814814814,
  'p': 0.07547169811320754,
  'f': 0.0999999955281252}}
```

```
rouge.get_scores(test_prediction_df['Generated Text'][0], test_prediction_df['Actual Text'][1], avg=True)

{'rouge-1': {'r': 0.14814814814814814,
  'p': 0.07547169811320754,
  'f': 0.0999999955281252},
 'rouge-2': {'r': 0.0, 'p': 0.0, 'f': 0.0},
 'rouge-l': {'r': 0.14814814814814814,
  'p': 0.07547169811320754,
  'f': 0.0999999955281252}}
```

Results:

In Conclusion we are using ROUGE score to calculate the matching words under the total count of words in reference sentence(summarizations). Furthermore, the ROUGE score ranges from 0 to 1, if it is 1 it means sentences are exactly matched the same. Otherwise, it means it's not matched exactly.

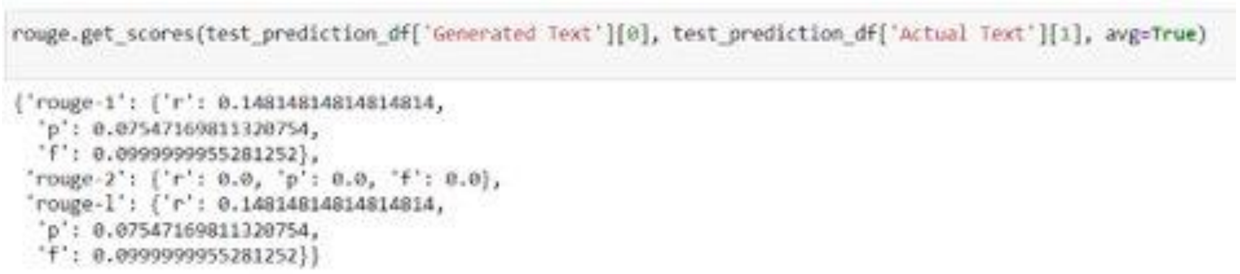
ROUGE is branched into three, ROUGE-1, ROUGE-2, ROUGE-L

In each category we are calculating PRECISION (P), RECALL(R), F1 SCORE

In PRECISON and RECALL we are calculating similarity between the predicted and target (model generated).

So, threshold has been given to 0.9.

- If it is more than 0.9 then we consider as 1(It means target and predicts are similar).
- If it is less than 0.9 then we consider has 0 (it means both target and predicted are not similar).



```
rouge.get_scores(test_prediction_df['Generated Text'][0], test_prediction_df['Actual Text'][1], avg=True)
```

```
{'rouge-1': {'r': 0.14814814814814814,
  'p': 0.07547169811320754,
  'f': 0.0999999955281252},
 'rouge-2': {'r': 0.0, 'p': 0.0, 'f': 0.0},
 'rouge-l': {'r': 0.14814814814814814,
  'p': 0.07547169811320754,
  'f': 0.0999999955281252}}
```

```
rouge.get_scores(test_prediction_df['Generated Text'][0], test_prediction_df['Actual Text'][1], avg=True)
```

```
{'rouge-1': {'r': 0.14814814814814814,
  'p': 0.07547169811320754,
  'f': 0.0999999955281252},
 'rouge-2': {'r': 0.0, 'p': 0.0, 'f': 0.0},
 'rouge-l': {'r': 0.14814814814814814,
  'p': 0.07547169811320754,
  'f': 0.0999999955281252}}
```

Limitation:

This project has a limited scope of development with moderate accuracy and can process only the text is in the English language.

References:

https://github.com/huggingface/transformers

https://towardsdatascience.com/text-summarization-with-nlp-textrank-vs-seq2seq-vs-bart-474943efeb09

Github:

https://github.com/thriloknadh/NLP_TEXTSUMMARIZATION.git