Text summarization

# Problem Statement

Build a text summarization system to highlight a summary of a given document (news article).

# Data set

[CNN news dataset](https://drive.google.com/file/d/1VGthRzHtBSIO182zMCMiqY-YV-D0mLLG/view) is a news dataset from CNN with 92k files with highlights of summary.

# Solution Link

<https://github.com/prabhudayala/cnn-mail-summarization>

# Overview

Text summarization in NLP is the process of summarizing the information in large texts for quicker consumption.

There are two types of summarization one being **Extractive Text Summarization** and the other as **Abstractive Text Summarization**. Extractive summarization is an old technique and due to availability of higher computing power and advancement of deep learning people are moving towards Abstractive summarization. On the other hand, while training Extractive models are fast – Abstractive models takes much more time to train and requires tons of data.

## Extractive Text Summarization

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.

## Abstractive Text Summarization

It is a more advanced method; many advancements keep coming out frequently. With the advancement of research and emergence of transformers model it has reached human parity and mostly it is only forward from now.

# Use cases

From summarizing news to summarize research papers – from generating title for news article to help researchers in medicine domain – summarization is here.

# Solutions

## Extractive Text Summarization

There are multiple ways to get extractive summarization.

1. Unsupervised
   1. Term frequency
   2. Sum basic
   3. Latent semantic indexing
   4. Non negative matrix factorization
   5. Page rank
   6. Embedding page rank
2. Supervised
   1. Binary Classification of sentences

I have tried page ranking using similarity matrix between sentences by creating a word embedding on all corpus of data using word2vec. **File (“intermediate/ extractive using ranking.py”)**

As the given data did not have exact sentences as highlights, I assumed that supervised model could not be trained.

## Abstractive summarization

There are multiple ways to get abstractive summarization. One being **training from scratch** and the other as to **fine-tune a pre-trained model** and of course get it done by using cloud services.

I have tried below point 1 and 2

1. **Training from scratch**: I have tried training a model from scratch using encoder – decode architecture. Due to less computation power, I could not arrive at a conclusion whether it will perform better or unseen data. **File: (“intermediate/abstractive tf keras.ipynb”)**
2. **Fine-tune a pre-trained model**: I have fine-tuned a pre-trained model (**facebook/bart-large-cnn**) from hugging face hoping it works better as it was trained on CNN daily mail. **File: (“intermediate/** **abstractive finetune bart.ipynb”)**
3. **Use cloud services:** 
   1. <https://docs.microsoft.com/en-us/azure/cognitive-services/language-service/text-summarization/overview>
   2. <https://cloud.google.com/ai-workshop/experiments/abstractive-document-summarization>