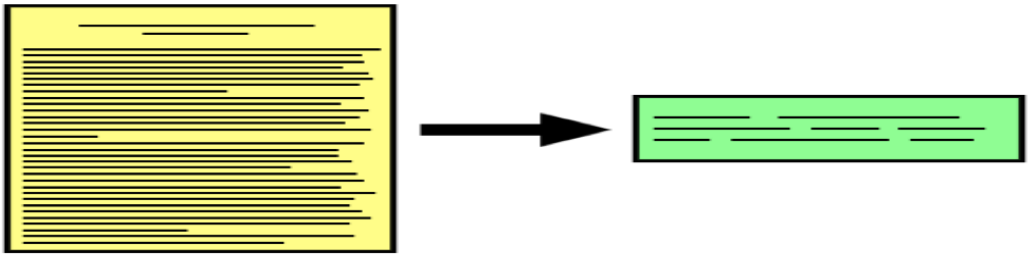
Capstone Project Proposal about Text Summarization using Machine Learning techniques

Domain background

With the rise of information technologies, globalization and Internet, an enormous amount of information is created daily, including a large volume of written texts. The International Data Corporation (IDC) projects that the total amount of digital data circulating annually around the world would sprout from 4.4 zettabytes in 2013 to hit 180 zettabytes in 2025. Dealing with such a huge amount of data is a current problem where automatization techniques can help many industries and businesses.
For example, hundreds or thousands of news are published around the world in a few hours and people do not want to read a full article for ten minutes, So, the development of automatic techniques to get short, concise and understandable summaries would be of great help to many global companies and organizations. Other use cases to consider: Media monitoring, legal contract analysis, question answering and bots, etc.

But this is not a recent subject, studies and projects have been carried out in this field for decades in order to achieve automatic text summarization systems. For example, Luhn in 1958 [2] or the DimSum in 1997 [3] are two examples of papers where Natural Language Processing techniques are applied to text summarization. In the 80s and 90s the classical NLP techniques where the foundations of the automatic text summarization systems but recently the advances in deep learning algorithms has become a new source of inspiration to face and solve this problem.



Source: https://sflscientific.com/data-science-blog/2016/11/17/text-summarization-in-natural-language-processing

Problem description and statement

Text Summarization is a challenging problem these days and it can be defined as a technique of shortening a long piece of text to create a coherent and fluent short summary having only the main points in the document. But, what is a summary? It is a *text that is produced from one or more texts, that contains a significant portion of the information in the original text(s), and that is no longer than half of the original text(s)* [1]. *Summarization clearly involves both these still poorly understood processes, and adds a third (condensation, abstraction,*

generalization). At this moment is a very active field of research and the state-of-the-art solutions are still not so successful than we could expect.

The problem, we are facing in this project, is to produce a summary (about 20-word length) from a single document much more longer, 300-400 words. It will be a single document summarization and its output could be an *extractive* (containing pieces of the source text) or *abstractive* (a new text is created) approach to the original text.

Dataset and inputs

When searching for information and data about text summarization I found hard to obtain a "good" dataset. Some of the most popular data sets are intended for research use, containing hundreds of thousands of examples and gigabytes of data that require high computational capacity and days or weeks to train. But we are interested in a dataset that could be trained faster, in a few hours, where we can experiment and develop easily.
We will use a dataset from Kaggle, [4], that consists in 4515 examples of news and their summaries and some extra data like *Author_name, Headlines, Url of Article, Short text, Complete Article*. This data was extracted from Inshorts, scraping the news article from Hindu, Indian times and Guardian.
An example:

- Text**:** *"Isha Ghosh, an 81-year-old member of Bharat Scouts and Guides (BSG), has been imparting physical and mental training to schoolchildren ..."*
- Summary**:** *"81-yr-old woman conducts physical training in J'khand schools"*

For a better performance, we should consider using a greater dataset like the CNN dataset that contains more than 93,000 news articles.

Solution Statement

Our goal will be to build a machine learning model to produce a short summary from a source text. We will analyze and apply one or two commonly used methods, compare each other and measure its performance. There are two main approaches to summarization:

- *Extraction-based summarization*: identify and extract keyphrases from the source, evaluate its *"importance"* and make a summary with the most valuable phrases.
- *Abstraction-based summarization*: entails paraphrasing and shortening parts of the source document where new sentences are created.

The latest advances in deep learning has been a boost for the development of abstraction based model but they require a lot of data and expensive computational resources so we will try to compare both methods, evaluate them and choose the one with an appropriate trade-off between performance and resources consume.

Benchmark Model

The purpose of this project is not to define a state-of-the-art solution to this problem because it will require a long investigation and a very large dataset to

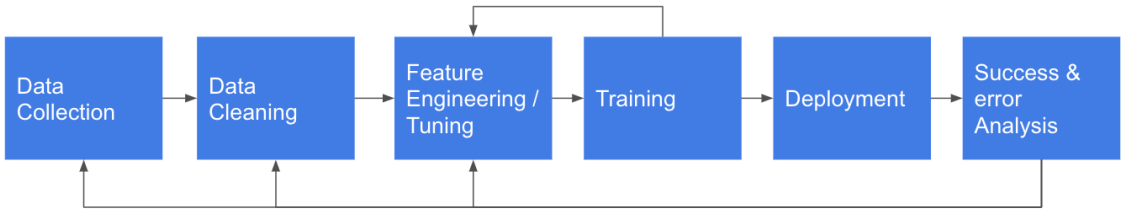
train for many days. But we want our model to be compared to a baseline model, a Gensim module for summarization based on the TextRank algorithm [5]. It is good enough to set a level of performance to compare with.

Evaluation Metric

To evaluate the model, we will use the ROUGE metric, *"ROUGE, or Recall-Oriented Understudy for Gisting Evaluation, is a set of metrics and a software package used for evaluating automatic summarization and machine translation software in natural language processing. The metrics compare an automatically produced summary or translation against a reference or a set of references (human-produced) summary or translation."*.[6] There are some variations of this metric: ROUGE-N, ROUGE-S and ROUGE-L For example, ROUGE-1 refers to overlap of unigrams between the system summary and reference summary. ROUGE-2 refers to the overlap of bigrams between the system and reference summaries. We will measure this metric on our model and the benchmark model to compare both approaches.

Project Design

We will follow the basic steps of a machine learning end-to-end project:



Once the data is collected, first we will explore it, extract some insights and apply some cleaning (apply lowercase, remove contractions, remove stopwords,..). Text data usually requires many preprocessing to get a useful set of string to work with. Then we will “manipulate” them to be consumed by the model (split, tokenize, padding,). The model building stage will require some research and experimentation to reach a good approach and finally we will deploy the model, so a user can call the model to get a summary from a source text.

We will design a workflow for data transformation to feed the training models. For experimentation and training we will use Google Cloud Platform resources and AWS Sagemaker, developing notebooks for data analysis and running scripts for training on the cloud. Finally, we will deploy our models on cloud for prediction tasks

Links

[1] - Hovy, E. H. Automated Text Summarization. In R. Mitkov (ed), The Oxford Handbook of Computational Linguistics, chapter 32, pages 583–598. Oxford University Press, 2005
[2] - Luhn, H., P. The Automatic Creation of Literature Abstracts. In Inderjeet Mani and Mark Marbury, editors, Advances in Automatic Text Summarization. MIT Press, 1999
[3] - Aonet, C., Okurowskit, M. E., Gorlinskyt, J., et al. A Scalable Summarization System Using Robust NLP. In Proceedings of the ACL’07/EACL’97 Workshop on Intelligent Sclalable Text Summarization, pages 66-73, 1997
[4] - Kaggle Dataset, https://www.kaggle.com/sunnysai12345/news-summary
[5] - https://rare-technologies.com/text-summarization-with-gensim/
[6] - https://en.wikipedia.org/wiki/ROUGE_(metric)