Machine Learning Engineer Nanodegree

Capstone Proposal

Wong Teck Meng
October 11, 2021

Automatic Text Summarization Application

Proposal

Domain Background

In this Information Age, the abundance of text content has created value for summarized text content. Summarization consist of condensing text to a concise summary of the initial text while retaining its most essential points and overall context.

Automatic Summarization is not new. First notable attempt dates back to 1958 made by IBM[1]. Since then, there are two main approaches to summarize text content:

1. Extractive summarization [2]– Essential sentence selection from a/set of documents.
2. Abstractive summarization [3]– Rewriting the text content in a more precise manner.

Problem Statement

The goal of this project is to create a web application that automatically summarize text input while retaining its most essential points to a text output.

Datasets and Inputs

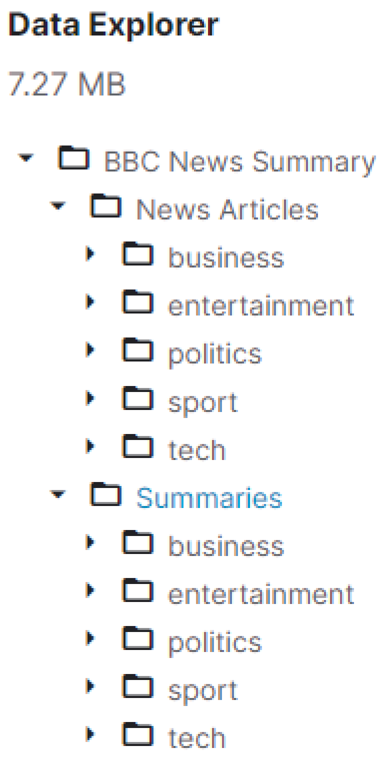
BBC News Summary[4] will be used for this project dataset.

[1] Luhn, H.P.: The automatic creation of literature abstracts. IBM J. Res. Dev. 2(2), 159–165 (1958)

[2] Sinha et al. : Extractive Text Summarization using Neural Networks (2018)

[3] Rush et al.: A Neural Attention Model for Abstractive Sentence Summarization (2015)

[4] https://www.kaggle.com/pariza/bbc-news-summary



Data Explorer

7.27 MB

- BBC News Summary
  - News Articles
    - business
    - entertainment
    - politics
    - sport
    - tech
  - Summaries
    - business
    - entertainment
    - politics
    - sport
    - tech

This dataset has 2225 news articles of BBC news website from 2004 to 2005. The dataset comprised of 5 topics, business, entertainment, politics, sport and technology. Each news article has summary provided in the Summaries folder. The first clause of the text of articles is the respective title. It was used in the paper of D. Greene and P. Cunningham. "Practical Solutions to the Problem of Diagonal Dominance in Kernel Document Clustering", Proc. ICML 2006; whose all rights, including copyright, in the content of the original articles are owned by the BBC. More at http://mlg.ucd.ie/datasets/bbc.html.

This project will use the news articles as text inputs to the automatic text summarization model and the corresponding summaries folder used as reference summary for evaluation purpose.

Solution Statement

This project is to build and deploy a REACT web application that input text and output summarized text. The application will provide evaluation metrics when reference summary is provided.

Benchmark Model

Extractive summarization using TF-IDF[5] will be used as benchmark model. This is the simplest algorithm for automatic text summarization and serve as baseline performance using ROUGE as the evaluation metric.

[5] https://en.wikipedia.org/wiki/Tf%E2%80%93idf

Evaluation Metrics

Recall-Oriented Understudy for Gisting Evaluation aka **ROUGE** [6]is a commonly used metric for summarization and translation in natural language processing. The metrics are based on comparison between a **reference summary** (usually human produced) and an automatically produced **system summary**. However, ROUGE does not take into account the sentence structure or grammatical correctness.

ROGUE comprises of various metrics. Of the following evaluation metrics available, ROUGE-N is the main one.

ROUGE-N

Overlap of N-grams between system and reference summaries. **ROUGE-1** will be measuring unigrams overlap. **ROUGE-2** and **ROUGE-3** would use bigrams and trigrams respectively.

ROUGE-L

Measure the longest matching sequence of words using **LCS**.

The following metric are provided by **ROUGE:**

Recall

Words in the **reference summary** that have been captured by the **system summary**. This measure how many information are captured in the **system summary**.

$$recall = \frac{\text{Number\_of\_overlapping\_words}}{Total\_words\_in\_reference\_summary}$$

Precision

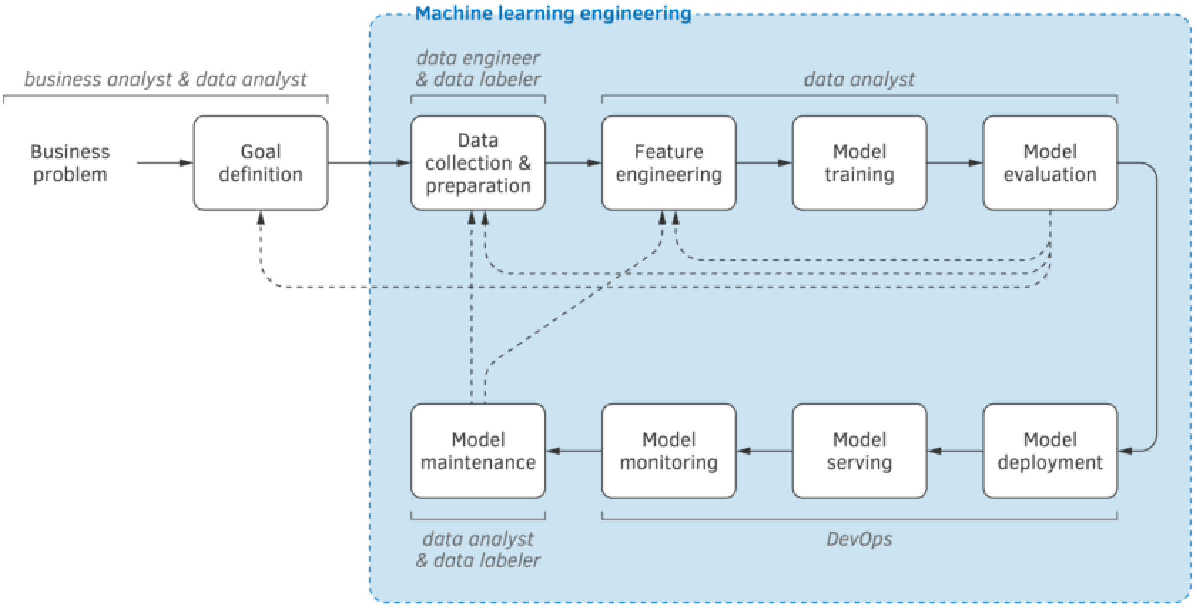
It measures relevance of **system summary**. This measures the number of relevant words.

$$recall = \frac{\text{Number\_of\_overlapping\_words}}{Total\_words\_in\_system\_summary}$$

[6] Chin-Yew Lin: Text Summarization Branches Out (2004)

Project Design

This project will start with the solution statement as proposed above. The workflow for approaching the solution is as in below figure:



[7]

The solid arrows show a typical flow of the project stages. The dashed arrows indicate that at some stages, a decision can be made to go back in the process for revision etc.

Model

This project will start with using **Extractive summarization** as the model. This method will follow these basic steps:

1. Create an intermediate representation of the text.
2. Score the sentences based on the chosen representation.
3. Rank and choose sentences to create a summary of the text.

Model Evaluation

ROUGE will be used as proposed above as the evaluation metric.

Model deployment and serving

FASTAPI[8] will serve as the backend to deploy and serve the solution model. **REACT**[9] will serve as the frontend for this project web application.

[7] Andriy Burkov: Machine Learning Engineering Figure 4 (2021)
[8] https://fastapi.tiangolo.com/
[9] https://reactjs.org/

Finally, **model monitoring** and **model maintenance** will be outside the scope of this project.