Extractive One Sentence Summation of News Articles.

Oct 2023

Authors

Peter Harmer: pharmer@wisc.edu
Girish Dodda: gdodda@wisc.edu
Rohan Avadhanam: ravadhanam@wisc.edu

Github Link

https://github.com/PJSHX/CS-ECE-539-Group-16-Project

Overview

A recurrent neural network based implementation for a one to two sentence extractive summarization of news articles.

Background

The main task of this project is to generate a one to two sentence extractive summation of news articles. Extractive summations is a technique that pulls words from the text itself to create a summary. The current known ways to achieve this vary wildly, but the current State of the art appears to be a neural network implementing Diffusion or BERT(Bidirectional Encoder Representations from Transformers). BERT uses two-stages of fine tuning to improve the extracted result [1]. Diffusion on the other hand generates a representative summary and extracts the words or sentences that best match [2]. Which is contrary to the previous method which is based on predicting and sequencing them. In either case the outcome was a general improvement of summation accuracy in comparison to previous models.

Our Dataset

We are using the BBC News Archive data set found on Kaggle [3]. It consists of BBC news articles from between 2004-2005 in five different topic areas. These areas being business, entertainment, politics, sport, and tech. The columns of the dataset are Category, Filename, Title, and Content. Of those the most useful to our model will be Title and Content, though Category may have a role. There are about 1517 unique titles in the dataset, however there are some null values in the Content column so some pruning may be necessary.

Our Method

Our plan is to make a neural network for the extractive summarization of news articles using the BBC News Archive data set found on Kaggle. With the summary being one to two sentences in length. This will be achieved primarily using a recurrent neural network or RNN. This is one of the older ways of achieving extractive summation but it is also the foundation upon which many of the later methods built upon and thus a great learning tool for the basic elements of natural language processing Which ties into the goals of this project, being to gain experience in the basics of natural language processing as well as the use of recurrent neural networks in general. A similar work to what we are implementing was created back in 2016 [5]. Their particular RNN extractive summarization used a sequence model for extraction with the goal of improving human interpretability of results and sentence level labels. The data used for this implementation was derived from CNN and Daily mail articles. This differs from our implementation primarily in that we are using BBC data and will be primarily operating on the word rather than sentence level.

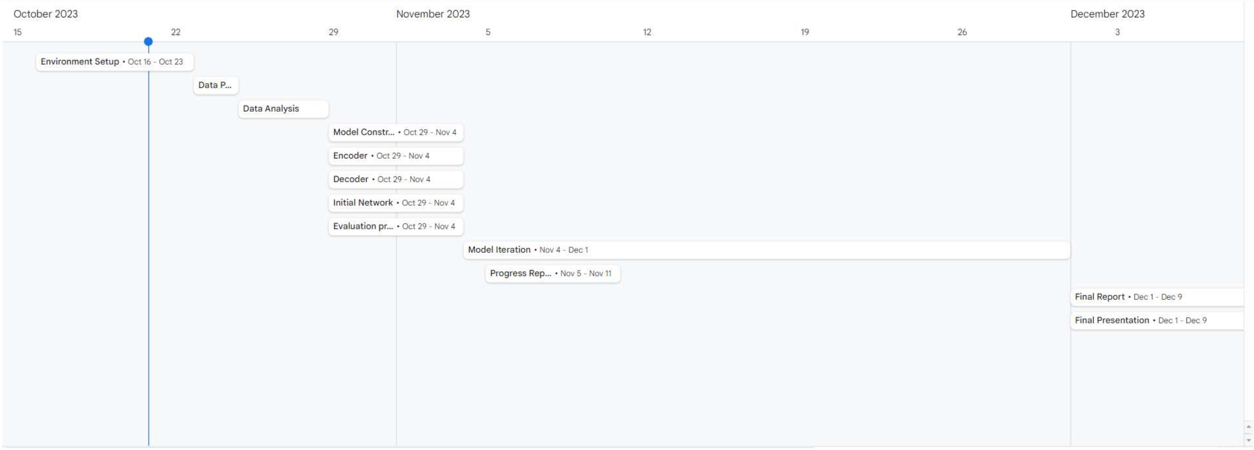
How is Performance Measured?

Performance will be measured by how close the extracted summary is to the provided article title. The particular method for this will be by utilizing a ROUGE metric, also known as Recall-Oriented Understudy for Gisting Evaluation. It's a set of metrics used to evaluate summarizations compared to human generated ones [4]. In our case, it would be comparing the machine generated summaries with the articles' title. This allows the generation of precision, recall, and F1 scores from which that model's success will be gauged. With the higher the scores the better the model. With the F1 score goal being somewhere close to above 85% accuracy. Though it is anticipated that early versions of the model will likely score lower.

Project Plan

The first tasks will be setting up an environment and data pre-processing. Github will be the primary environment from which the project will be managed and stored. The next step is data preprocessing which will consist of removing nulls, filling in missing titles with a manually generated one if necessary, setting all text to lowercase, replacing contractions with component or full words, removing and/or replacing special characters, and removing punctuation. After that will be analysis where the processed data will be examined to determine initial model parameters like the number of network layers. Next will be model construction where the encoders, neural network, and decoders will be built. The encoders convert the words into values the network can interpret and the decoders turn those numbers back into words. Following construction will begin an iteration stage where the model's results will be used to tweak the model repeatedly. This will be achieved by testing various neural network parameters to optimize the ROUGE derived F1 score. This stage may also include adjustments to how the encoders and decoders are designed or changing activation functions. The goal of the iteration stage is an F1 score at or above 85% if possible. Likely sometime during the early stages of the iteration state is when a progress report will be compiled on the project, wherein the current state of the model and early results will be provided. Towards the end of the iteration stage preparations for the final report and presentation will begin. With the report and presentation being delivered in week 7.

Gantt Chart



References

[1] BERT: https://paperswithcode.com/paper/text-summarization-with-pretrained-encoders

[2] Diffusion: https://paperswithcode.com/paper/diffusum-generation-enhanced-extractive

[3] Link to Dataset: https://www.kaggle.com/datasets/hgultekin/bbcnewsarchive

[4] ROUGE
https://www.freecodecamp.org/news/what-is-rouge-and-how-it-works-for-evaluation-of-summaries-e059fb8ac840/

[5] https://paperswithcode.com/paper/summarunner-a-recurrent-neural-network-based

[6] Towards data science sequenced implementation: https://towardsdatascience.com/text-summarization-using-deep-neural-networks-e7ee7521d804

#Database of papers for future reference
https://paperswithcode.com/task/extractive-summarization