Text Summarization project

Github: https://github.com/olufemi-olumaiyegun/CSCE4290-PROJ

Team Members

Olufemi Olumaiyegun

Goals and Objectives

Motivation and Significance

Text Summarization is hot field of ongoing research, and it represents one of the best ways for beginners to get immersed in Natural Language Processing. Furthermore, using the skills NLP skills I have learned for the semester to come up with a project I can put on my resume and demonstrate to potential employers is bonus that motivates me to complete the project.

Background

There are literal tons of textual information on the internet, and it is infinitely increasing, it is a no-brainer that the need of automation increases right along with it. As a result, Text Summarization with NLP is useful in multiple ways. Beyond summarizing articles, it can become handy in summarizing multi document texts like headlines, outlines, minutes, reviews and many more. Consequentially, due to the proliferation of data mining and the lure of accurate predictions, Text Summarization cuts across multiple industries and its automation often increases productivity and efficiency in said industries. Text Summarization remains an open field with continuous research and has been so since it was brought into the limelight in the 1980's as a hailed candidate for replacing Machine Translation which was a failed attempt to automatically translate Russian text to English. The history of Text Summarization is interwoven with the genesis of NLP, as they both started to gain interest from researchers right around the same time.

Initial text summarization research focused on Term Frequency (TF) and Term Frequency – Inverse Document Frequency (TF-IDF) that weighs importance based on frequency of words in text and is the basis of Extractive Text Summarization. Later in 1999, Nevill-manning proposed the extra steps of disregarding sentences based on length and evaluating summarization based on position of sentences in the document. Continuously over the years, more instances have been incorporated into Text Summarization models which include calculating sentence vectors based on their similarity to the article's title, keywords, and lot more which keep growing. My work for this project will piggyback off several concepts and techniques discovered by researchers in the field of Artificial Intelligence.

Objectives

The objective of the project is to design and implement a text summarizer API in python that accepts text content in multiple formats and returns a summary as close to what a human author would.

Features:

- The project will feature a Bi-directional LSTM learning model that can summarize the posts in a much understandable manner.
- The text summarizer will provide three types of summaries: very brief, brief, beefy in ascending order of length of summary.

However, having a text summarizer readily available as an API for developers to use in multiple high-level languages may turn out to be useful.

The work this would involve is:

- Creating an API using flask in Python (incomplete).
- A data washing class that can receive different formats of data and convert them to a string of text (complete).
- Training a learning model using Keras that can perform summary on text(incomplete).
  - Sourcing an appropriate dataset for the training model (complete).
  - Preprocessing the dataset (complete).
  - Implementing the Bi-directional LSTM model as a class(incomplete).
  - Training the model(incomplete).
  - Evaluating the model. (incomplete)
- Providing a server where this API will be hosted on.

N/B: All work was and will be completed by the single team member.

Limitations and Constraints:

1. Implementing a very good text summarizer requires heavy, extensive, and well-organized datasets that are difficult to come by.
2. Since very large datasets are required, it often requires high powered expensive computers to train models.
3. It is hard to determine what is a good summary since that is subjective to users and they may expect a different type of summary.

Dataset(s) source(s):

Timeline Dataset: I chose this dataset after doing some research because, the project in which this dataset was generated involved classifying and summarizing articles and timelines related to any date you want. A certified and extensive dataset like that will be useful in a text summarization project.

WikiHow Dataset: This dataset contains articles and their summaries written by professional journalists. The dataset contains a section of articles which follow the inverted Pyramid style which signifies that the most important elements of a story are placed at the beginning. Nevertheless, it also has articles written by non-professionals and this provides ample variation in the data. Currently the dataset has over 230, 000 article-summary pairs which are more than the Document Understanding Conferences texts and the Giga word corpus.

DESIGN OF LEARNING MODEL

Abstractive Text Summarization is the latest cutting edge research trend

While there are two techniques to text summarization (Extractive and Abstractive Text Summarization), Abstractive Text Summarization has recently had a majority consensus to be a more robust technique compared to Extractive Text Summarization because of its ability to retain meaning while after summarization. The main feature of Abstractive TS is that it a text generating technique. Abstractive TS may generate words that are not part of the content being summarized, however, it retains the overall meaning of the text and therein lies the advantage it has over Extractive Text Summarization.

Cons:

- Abstractive TS loses to Extractive TS when performance is a highly necessary factor.
- Abstractive TS is a tad more difficult to implement compared to Extractive TS.
- Abstractive TS cannot properly handle scenarios containing unseen words and would often trip up the entire summary.
- Abstractive TS needs extensive and well architected datasets to be the best of itself.
- Good datasets are hard to find that sufficiently take advantage of the benefits Abstractive TS offers.

Taking advantage of the power of Abstractive Summarization

Implementation of a Recurrent Neural Network variation (Bi-Directional LSTM) combined with an Attention learning model will provide ample features for training a text summarizer. The reason I have chosen the LSTM – Attention model for this project is because of the algorithm's capability to remember long term dependencies as well as provide context for this is very crucial to any text summarization task.

Implementing Bi-Directional LSTM:

Taking advantage of LSTM's past memory retention to make predictions and taking it a step further by adding an extra LSTM model that make predictions using the reverse. The benefit to using this type of model for a text summarization project is that it is one of the best methods to perform abstractive summarization based on its ability to work out a synchronous relationship between the input and the output.

Since Abstraction requires a sequence-to-sequence model which is implemented using an Encoder-Decoder model, my method uses one LSTM model each for the Encoder and the Decoder separately.

Pros:

- Because of its ability to remember long term dependencies, LSTM generates human like summaries of text.
- With the addition of an attention algorithm, the problem of inaccuracies due to larger data sets is addressed and thus, its accuracy has a direct relationship with the size of the dataset, so it gets more accurate with more data.

Cons:

- They are slow because they need to be encoded and decoded.
- It is a complex algorithm model because it requires more steps than other algorithms.

- They are difficult to train because they require a large amount of memory bandwidth

Language Model:

I have chosen to use a Trigram (3-gram) model for this project. Due to efficiency reasons based on machine and GPU constraints, 3-gram is the highest number of n-grams I can use for this project. A high enough n-gram model will improve the training model for the text summarizer since it covers a much longer range than a unigram or bigram.

References

M. Koupaee and W. Y. Wang, “WikiHow Dataset.” https://arxiv.org/abs/1810.09305, online, 18-Oct-2018.

E. Johnson and A. Gutierrez, “Abstractive Text Summarization using Attentive Sequence-to-Sequence RNNs,” dissertation, 2020.

F. Kiyoumarsi, “2nd GLOBAL CONFERENCE On LINGUISTICS And FOREIGN LANGUAGE TEACHING, LINELT-2014,” in ScienceDirect.

P. K. B, “Brief history of Text Summarization,” Medium. Medium, 17-Jul-2021.

Al-Sabahi, Kamal & Zuping, Zhang & Kang, Yang, “Bidirectional Attentional Encoder-Decoder Model and Bidirectional Beam Search for Abstractive Summarization”, 2018.

D. Suleiman and A. Awajan, "Deep Learning Based Abstractive Text Summarization: Approaches,Datasets, Evaluation Measures, and Challenges", Mathematical Problems in Engineering, vol. 2020.

(1.2) G. B. Tran, M. Alrifai and D. Q. Nguyen. 2013. Predicting Relevant News Events for Timeline Summaries In Proc. 22th WWW2013 [pdf]