Summarizing COVID-19 News Articles Using NLP Techniques and Deep Learning

David Sasu

Abstract

The occurrence of the COVID-19 virus pandemic has recently generated a lot of buzz in the media. However, keeping up with the torrent of online publications about the state of the virus is quite difficult and overwhelming. As a result of this, this paper presents a deep learning model that can be applied to summarize COVID-19 news articles to ensure that the main ideas being communicated by these articles can be quickly and easily assimilated. The deep learning model that was developed for this paper is the attentional encoder-decoder sequence to sequence network model and the quality of the summaries generated by the model was evaluated through the application of ROUGE scores.

1 Introduction

The recent and ongoing occurrence of the COVID-19 pandemic has consistently generated a lot of waves in the media. This has led to a plethora of news articles being generated every day to keep the general public abreast with the current state of affairs with regards to the virus.

Paradoxically, the increase in the information available has made it very difficult to keep up with recent developments since one can get easily overwhelmed with too much information. To address this problem of information overload, this work seeks to present an approach to summarizing the text found within COVID-19 news articles.

Text summarization can be categorized into two distinct classes: abstractive and extractive (Yousefi-Azar and Hamey, 2017). In abstractive text summarization, a summary is generated from any given source text by using the main ideas expressed in the text whiles in extractive text summarization, a summary is constructed by combining the most important sentences in the source text (Yousefi-Azar and Hamey, 2017).

With regards to specific approaches implemented for text summarization, some approaches involve topic representation, while other approaches employ indicator representation (Aggarwal and Zhai, 2012). In topic representation, the major topics represented in a given input text are extracted and used to score the importance of each sentence in the input text and the sentences that have a high score are combined to form the summary of the text. However, in indicator representation, a given input text is represented by a set of indicators which are combined and used to score the level of importance of each sentence within the input text to determine which sentences should be included in the summary.

In contrast with the text summarization approaches described previously, different deep learning models have been developed and applied to the task of text summarization. This work uses deep learning to generate summaries of a given input text. The deep learning model used in this work is an attentional encoder-decoder sequence to sequence network.

2 Related Work

Various machine learning techniques involving neural networks such as those proposed in (Kaikhah, 2004); (Svore et al., 2007); (Nallapati et al., 2016); have been applied in the summarization of text.

In (Kaikhah, 2004), a shallow neural network was developed and successfully used to summarize news articles. This was accomplished by first training the neural network on a corpus of news articles and then modifying the resulting network through a process known as 'feature fusion'. The process of 'feature fusion' involves: 1) eliminating uncommon features and 2) collapsing the effects of common features.

(Svore et al., 2007) present a neural network system called NetSum for automatic text summarization. In the implementation of this system, features based on news search query logs and Wikipedia entities were developed and used to estimate the level of importance of each sentence in the given document. The Rank-Net learning algorithm was then used to train a pair-based sentence ranker to score every sentence in an input document to identify the most important sentences to include in the document summary.

Similar to the work presented in this paper, (Nallapati et al., 2016) applied an attentional encoder-decoder recurrent neural network to summarize text documents. The work presented in this paper is similar to what was presented in (Nallapati et al., 2016) in the sense that, attentional encoder-decoder networks are also applied here, however, this work employs the use of a gated recurrent unit instead of a recurrent neural network for sequence modelling.

3 Data

The data set that was used in the implementation of the deep learning models presented in this paper was obtained from the IEEE DataPort and it was submitted by Ran Geva. It consisted of COVID - 19 articles collated from news websites, message boards and blogs over a period of 4 months, from December 2019 to March 2020.

The dataset was organised into 31 files with each file containing strings of json objects, where each json object string represented a distinct article. There were a total of 2,711,460 json object strings in the dataset. However, 100 strings of json objects were selected from an arbitrary file and used as the development dataset for the deep learning models developed in this work and 300 strings of json objects were selected for training the model since there were limitations in the RAM available for training.

3.1 Reorganising the dataset

Before any preprocessing could be performed on the data, it had to be first reorganised into source and target data. The source data consisted of the main body of text reported in the articles and the target data consisted of the corresponding titles of these bodies of text. The main idea was to develop a deep learning model that could generate suitable text, such as the titles of the articles, to summarize any body of text that it is given as input.

3.2 Preprocessing the dataset

Preprocessing the dataset involved first going through each line of text in the source and target data and getting rid of stop words. After this, all the words were transformed to lowercase and the contracted representations of words were expanded. Regular expressions were also applied to remove all apostrophes, parentheses and punctuation.

To ensure the standardization of the length of the sentences in the source data, the average length of the sentences in the source data were computed and all sentences that exceeded this average value were cropped to to fit it. This same process was replicated for the sentences in the target data.

4 Model

The attentional encoder-decoder sequence to sequence deep learning model was adopted for the objective of text summarization because: 1)the main aim is to convert a given sequence of text, in the form of a news article, into another sequence of text, in the form of a news article summary, and a sequence to sequence model is perfect for this task and 2)the conversion of a given news article to a corresponding news article summary demands a deep learning model that can represent the important segments of the news article in a succinct way so that a summary can be generated from that representation, and an attentional encoder-decoder model is most suitable for this specific task.

A high-level overview of the attentional encoder-decoder sequence to sequence model applied in this work is displayed in Figure 1.

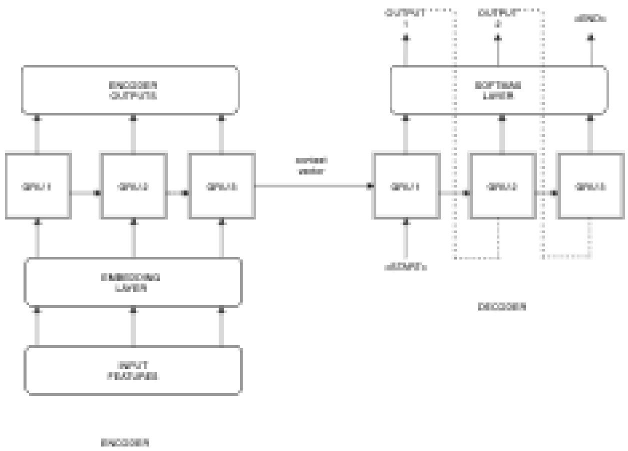

Figure 1: Attentional Encoder-Decoder Sequence to Sequence Model

Gated Recurrent Units (GRUs) were chosen to implement the sequence models in this work because they have a much simpler architecture compared to other recurrent neural networks that perform well on long sequences such as Long Short Term Memory networks (LSTMs). As result of this simple architecture, they are more efficient in computation.

4.1 Training the Model

The initial step taken to train the model involved converting all the words in the preprocessed news articles in the source data into tensor objects that could be fed as inputs to the encoder. After each word had been converted into a tensor, each sentence in the source data,now consisting of tensor objects instead of words, was iteratively supplied to the encoder as an input.

The tensor objects representing the words in the input sentence were however not passed to the encoder sequence model right away. They were first pushed through an embedding layer to ensure that an embedding was generated for each tensor. This was done to reduce the dimensions of the input tensors.

After all the embeddings representing the words of an input sentence from the source data have been pushed through the encoder network, the generated outputs of the network are logged and subsequently used in the computation of attention weights while the vector representing the final hidden state of the encoder network is passed on to the decoder network as a context vector.

In the decoder network, the context vector from the encoder,a computed attention score for the initial network state and an SOS tag, indicating the start of the summary sentence, are fed into the first state of the network to generate the first predicted word of the input sentence summary.

The attention weights for each state in the decoder network are computed by first generating scores to indicate how relevant each source token in the encoder network is in the prediction of a particular decoder state and then applying a weighted sum over the generated scores.

In subsequent states other than the first state in the decoder network, the correct word for a previous state is supplied as an input to a current state in addition to the hidden state from the previous state and the attention value for that current state to generate a word prediction. This approach is known as 'teacher forcing' and it has been demonstrated to lead to higher quality sequence predictions (Goodfellow et al., 2016).

4.2 Testing the model

A test set consisting of 100 articles was used to access the efficacy of the performance of the developed model. These articles were drawn at random from a data file that was arbitrarily chosen from the given set of data files.

During the testing process, each of the articles from the test set were run through the trained model and the generated summary results were compared to the target text for the article, which consisted of the title of the input news article.

The assessment of the quality of the prediction during the process of comparison with the target text was computed by applying the retail-oriented understanding for gisting evaluation (ROUGE). ROUGE provides information on the precision, recall and f-measure scores that are obtained when a sequence of text is compared to another sequence of text (Lin, 2004).

The results displayed in Table 1 were obtained after testing the model:

Average Precision	Average Recall	Average F-measure
0.1807	0.1608	0.1660

Table 1: Attentional encoder-decoder GRU sequence to sequence model test results

5 Experiments

To fully explore the performance of the developed model, experiments were performed to answer the following questions:

1. How well does the developed model perform when the amount of data used to train it is increased?

2. How well does the developed model compare to different sequence to sequence model?

3. How well does the developed model perform when compared to an unsupervised model?

To answer the first question, the experiment that was performed involved adding increments of 50 articles to the training data set and checking the performance of the model after each increment. The results generated from the experiment can be seen in Figure 2

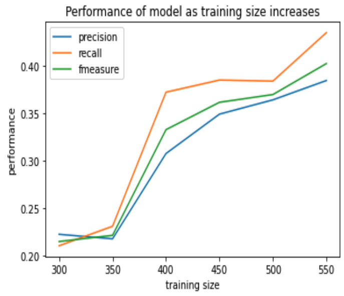

Figure 2: Attentional Encoder-Decoder Sequence to Sequence Model Performance

To answer the second question, an attentional encoder-decoder sequence to sequence network implemented with LSTMs instead of GRUs was developed and used. This model was chosen because LSTMs also perform well when applied to sequence to sequence tasks such as text summarization. When this model was trained and tested with the same data, the results it produced is displayed in Table 2.

Precision	Recall	F-measure
0.0	0.0	0.0

Table 2: Attentional encoder-decoder LSTM sequence to sequence model test results

To answer the third question, an unsupervised model was implemented to summarize any given input text through the application of the ideas of extractive text summarization. This unsupervised model is to act as the baseline to the model developed in this work since it is not generating any text but rather returning salient sentences which already exist in the input data. The results generated by the unsupervised model are displayed in Table 3.

Precision	Recall	F-measure
0.0643	0.3478	0.0975

Table 3: Unsupervised model test results

6 Analysis

From the second experiment, it can be observed that the sequence to sequence model implemented with LSTM performed extremely poorly when trained and tested with the same data that was applied to the sequence to sequence model implemented with GRU. Perhaps this can be attributed to the fact that the sequence to sequence model implemented with the LSTM requires more data than the GRU to train in order to generate better text summaries.

From the last experiment, it can be seen that the unsupervised model produced results that were generally similar to the results produced by the attentional encoder-decoder GRU sequence to sequence model. However, the results from the first experiment indicate that the attentional encoder-decoder GRU sequence to sequence model would outstrip the unsupervised model in terms of performance once it is trained with more data.

7 Conclusion

In this paper, an attentional encoder-decoder sequence to sequence network model was developed and applied to the task of text summarization. The text data that was used in the development of the model consisted of COVID-19 news articles and the sequence to sequence model that was chosen and implemented in the model was the GRU.

In the experiments that were performed to ascertain the performance of the GRU sequence to sequence model, it was demonstrated that the initial performance of the model was similar to that of an unsupervised extractive text summarization baseline model. However, it was also demonstrated with further experiments that the subsequent performances of the GRU sequence to sequence model keeps improving as its training data size is continually increased.

The experiments performed in this work also demonstrated that the GRU sequence to sequence model requires less training data than the LSTM sequence to sequence model in order to perform satisfactorily. Perhaps the simpler architecture of the GRU enables it to learn better with smaller amounts of data.

A possible avenue for future research could be comparing the performance of sequence to sequence models to other deep learning models such as transformers in the task of text summarization.

References

Charu C Aggarwal and ChengXiang Zhai. 2012. *Mining text data*. Springer Science & Business Media.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.

Khosrow Kaikhah. 2004. Text summarization using neural networks.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Krysta Svore, Lucy Vanderwende, and Christopher Burges. 2007. Enhancing single-document summarization by combining ranknet and third-party sources. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pages 448–457.

Mahmood Yousefi-Azar and Len Hamey. 2017. Text summarization using unsupervised deep learning. *Expert Systems with Applications*, 68:93–105.