TEXT SUMMARIZATION VIA DEEP LEARNING

Kushnazarov Farruh Isakulovich
PhD, iTutorGroup-Ping An, Big Data & AI Manager, Data Scientist
Nosirov Jaloliddin A'zamjon o'g'li
2nd year Master
Tashkent University Of Information Technologies Named After Muhammad Al-Khwarizmi, Uzbekistan

Abstract

This knowledge these days is stored in various formats in huge repositories mostly in the form of documents, sheets, photos, videos. One finds it difficult to comprehend this whole lot of information. There by, here comes the need of text summarization [2]. Text summarization is a process of extracting the context of a large document and summarize it into a smaller paragraph or a few sentences. Text summarization plays a vital role in saving time in our day to day life. It is also used in many bigger project implementations of classification of documents or in search engines [8]. Text Summarization has become an important and timely tool for assisting and interpreting text information. It is generally distinguished into: Extractive and Abstractive. The first method directly chooses and outputs the relevant sentences in the original document; on the other hand, the latter rewrites the original document into summary using NLP techniques. From these two methods, abstractive text summarization is laborious task to realize as it needs correct understanding and sentence amalgamation. This paper gives a brief survey of the distinct attempts undertaken in the field of abstractive summarization [5].

Key Words: *text summarization, deep learning, long short term memory, natural language processing, recurrent neural networks – RNN, abstractive summary, extractive summary.*

1. Introduction

Currently, there are vast quantities of textual data available, including online documents, articles, news, and reviews that contain long strings of text that need to be summarized. Text summarization can be divided into several categories based on function, genre, summary context, type of summarizer, and number of documents. [3]. Text summarization is a process of producing brief and concise summary by capturing the vital information and the comprehensive meaning. Text summarization is achieved by natural language processing techniques by using algorithms like page rank algorithms etc. While these algorithms fulfil the objective of text summarization, they cannot generate new sentences which are not in the document like humans. They can also have grammatical errors. This is where Deep Learning comes to our rescue. The use of deep learning builds an efficient and fast model for text summarization. The use of deep learning methods helps us generate summaries which can be formed with new phrases and sentences and also which are grammatically correct. Text Summarization is broadly classified into two types:

1. Abstractive Text Summarization - The abstractive text summarization can create new phrases and sentences that relay the most useful information from the original text. The sentences generated through this method may not be present in the original document [8]. Motivated by neural network success in machine translation experiments, the attention-based encoder-decoder paradigm has recently been widely studied in abstractive summarization [1].

2. Extractive Text Summarization - The extractive text summarization involves pulling key phrases from the source document and combining them to make a summary. We identify important words or phrases from the text and extract only those for the summary.

2. Implementation

We implement the abstractive method using the deep learning technique called Long Short Term Memory (LSTM) which is a type of Recurrent Neural Network Algorithms. The data used for this project is CNN_dailymail dataset.

Data. The data used is the CNN_dailymail dataset. It has two features: article and highlights. The article includes the document that is to be summarized. It is the news article. Highlights are the headlines of the corresponding news which are used as summaries.

Method. The model used is the abstractive method which is implemented using deep learning techniques.

Algorithm. The algorithm used is the LSTM or Long Short Term Memory model which is a type of Recurrent Neural Network model.

Model. The model used is sequence to sequence model. Sequenceto-sequence learning is a training model that can convert sequences of one input domain into the sequences of another output domain. It is generally used when the input and output of a model can be of variable lengths.

3. Methods

CleanData(). It is used to clean the data by using preprocessing steps mentioned earlier.

BuildDataset(). It is used to build train and test data sets**.**

BuildDict(). It is used to build dictionary where keys are words and values are random and unique numbers. It also builds another dictionary with keys as unique numbers and values as words. These are used in tokenization of words so that the input to the model is a set of numbers rather than words so as to make the computation easier using vectors.

Tokenize(). It is used to tokenize the data and send it to the model. Tokenizing data is important as the networks need numerical data to work on rather than raw data with characters.

3.1. Algorithm. In the current days, we are trying to create algorithms which can help us replicate the human brain and achieve its functionalities. This has been achieved by the neural networks. Neural Networks are the set of algorithms that an recognize patterns in the data. They closely resemble the human brain and have the capability to create models that can work or function like a human brain. Recurrent Neural Network (RNN) are a type of neural networks. They are feedforward neural networks which have an internal memory. In a traditional neural network, the input and the output sequences are independent of the each other. But in order to predict a sequence or a sentence, we need to know the previous words to predict the next word. Hence, we need internal memory. RNN helps us store the previous memory with the help of hidden states which remembers information about previous sequences.

RNNs work perfectly when it comes to short contexts. But when we want to create a summary of a complete article, we need to capture the context behind the complete input sequence and not just the output of the previous input. Hence, we need a network that can capture the complete context like a human brain. Unfortunately, simple RNN fails to capture the context or the long term relation of the data that is it cannot remember or recall data in the input that occurred long before and hence cannot make an effective prediction. RNN can remember data or context only for a short term. This is called vanishing gradient problem. This issue can be resolved by a slightly different version of RNN - The Long Short Term Memory Networks.

Long Short-Term Memory (LSTM) networks are a better version of RNN. They can remember the past data easily by resolving the vanishing gradient problem. LSTM uses back propagation to train the model. LSTM is well-suited for predictions and classifications of data sequences of unknown durations. They can also be used in language translation and text summarization methods.

3.2. Model. The model used here is sequence to sequence model. Sequence-to-sequence learning is a training model that can convert sequences of one input domain into the sequences of another output domain. It is generally used when the input and output of a model can be of variable lengths. It is a method of encoder-decoder based machine translation that maps an input of sequence to an output of sequence with a tag and attention value. The idea is to use two LSTMs that will work together with a special token and try to predict the next state sequence from previous sequence.

Encoder-Decoder architecture: The Sequence to Sequence model uses a method of encoder-decoder based machine translation. Encoder-Decoder architecture is used in predicting sequences when the length of output and input data may vary. The input sequence is read entirely by the encoder and a fixed length internal representation is generated. The internal representation captures the entire context of the input data sequence. The decoder network uses this internal representation to predict the output words until the end of the sequence token is reached (Fig – 1).

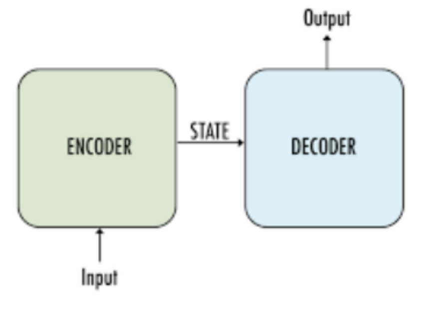

Fig – 1. Encoder Decoder.

Encoder. An encoder is an LSTM network which reads the entire input sequence. At each time step, one word from the input sequence is read by the encoder. It then processes the input at each time step and captures the context and the key information related to the input sequence. It takes each word of input(x) and generates the hidden state output (h) and the cell state which is an internal state(c). The hidden state(h_i) and cell state(c_i) of the last time step are the internal representation of the complete input sequence which will be use to initialize the decoder (Fig – 2).

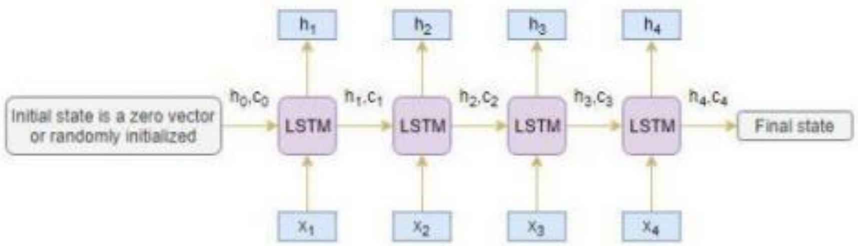

Fig – 2. Encoder.

Decoder. The decoder is also an LSTM network. It reads the entire internal representation generated by the encoder one word at a time step. It then predicts the same sequence offset by one time step. The decoder is trained to predict the next word in the output sequence given the previous word based on the contextual memory stored by the LSTM architecture. Two special tokens and are added at the beginning and at the end of the target sequence before feeding it to the decoder. We start predicting the target sequence by passing one word at a time. The first word of output of the decoder is always token. The end of the output sequence is represented by token (Fig – 3).

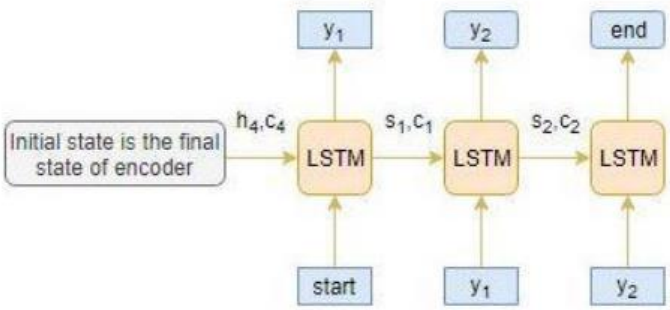

Fig – 3. Decoder.

The above architecture of the model is built using the TensorFlow library which used to build layers in neural networks. The final architecture of the model will be as shown below (Fig – 4).

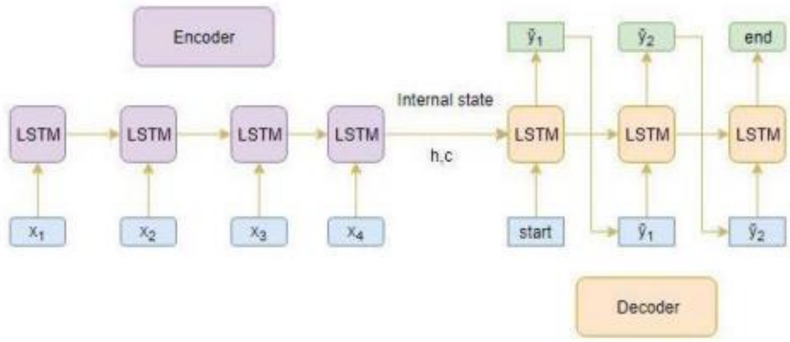

Fig -6. LSTM Seq2Seq model architecture.

Attention layer. A Sequence to Sequence model with an attention mechanism consists of encoder, decoder and an attention layer. Attention mechanism is used to secure individual parts of the input which are more important at that particular time. It can be implemented by taking inputs from each time steps and giving weightage to time steps. The weightage depends on the contextual importance of that particular time step. It helps pay attention to the most relevant parts of the input data sequence so that the decoder can optimally generate the next word in the output sequence

4. Conclusion

The increasing growth of the Internet has made a huge amount of information available. It is difficult for humans to summarize large amounts of text. Thus, there is an immense need for automatic summarization tools in this age of information overload. 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. That's a huge amount of data circulating in the digital world. There is a dire need of algorithms which can be used to automatically shorten the amount of data with accurate summaries that capture the essence of the intended messages. Furthermore, applying text summarization reduces reading time and accelerates the process of researching for information playing a major role in current era of rapid development and digitalization.

In this work, we apply the attentional encoderdecoder for the task of abstractive summarization with very promising results, outperforming stateof-the-art results significantly on two different datasets. Each of our proposed novel models addresses a specific problem in abstractive summarization, yielding further improvement in performance. We also propose a new dataset for multisentence summarization and establish benchmark numbers on it. As part of our future work, we plan to focus our efforts on this data and build more robust models for summaries consisting of multiple sentences

References

[1] Ekaterina Zolotareva, Tsegaye Misikir Tashu and Tomáš Horváth. Abstractive Text Summarization using Transfer Learning. CEUR-WS.org/Vol-2178/paper28 (2020).

[2] Soma Shrenika, GouriPriyaRamini, LunsavathBhagavathiDevi, B Geetavani. Abstractive Text Summarization By Using Deep Learning Models. *Turkish Journal of Computer and Mathematics Education*. Vol.12 No.14 (2021).

[3] Dima Suleiman and Arafat Awajan. Deep Learning Based Abstractive Text Summarization: Approaches, Datasets, Evaluation Measures, and Challenges. Volume 2020, Article ID 9365340. Published 24 August 2020. https://doi.org/10.1155/2020/9365340.

[4] Panagiotis Kouris, Georgios Alexandridis, Andreas Stafylopatis. Abstractive text summarization based on deep learning and semantic content generalization. *Proceedings of the 57th*

Annual Meeting of the Association for Computational Linguistics, pages 5082–5092 Florence, Italy, July 28 - August 2, 2019. c 2019 Association for Computational Linguistics.

[5] Neha Rane, Sharvari Govilkar. Recent Trends in Deep Learning Based Abstractive Text Summarization. International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8 Issue-3, September 2019.

[6] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çaglar Gulçehre, Bing Xiang. Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, pages 280–290, Berlin, Germany, August 7-12, 2016. c 2016 Association for Computational Linguistics.

[7] Chandra Khatri, Gyanit Singh and Nish Parikh. Abstractive and Extractive Text Summarization using Document Context Vector and Recurrent Neural Networks. KDD'18 Deep Learning Day, August 2018, London, UK.

[8] Kasimahanthi Divya, Kambala Sneha, Baisetti Sowmya, G Sankara Rao. Text Summarization using Deep Learning. International Research Journal of Engineering and Technology (IRJET). Volume: 07 Issue: 05 | May 2020. www.irjet.net.

[9] Julien Romero, Thomas Hofmann, Jason Lee. Abstractive Text Summarisation with Neural Networks (2017).

[10] Rana Talib Al Timimi, and Fatma Hassan Al Rubbiay. Multilingual Text Summarization using Deep Learning. International Journal of Engineering Research and Advanced Technology (IJERAT). Volume.7, No. 5 May – 2021. https://ijerat.com/index.php/ijerat.