Abstractive Summarizer – A meaningful gist

Swapnil Sagar,
Department of CSE,
MIT School of Engineering,
MIT ADT University,
Pune, India.
swapnilsagar09@gmail.com

Monti Patel,
Department of CSE,
MIT School of Engineering,
MIT ADT University,
Pune, India.
montipatel1997@gmail.com

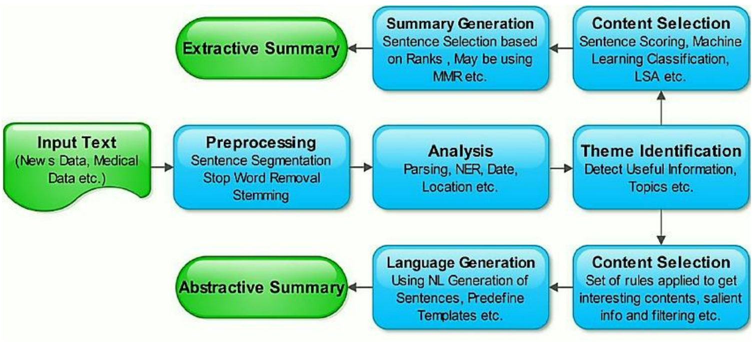
Swapnil Jagtap,
Department of CSE,
MIT School of Engineering,
MIT ADT University,
Pune, India.
swapniljagtap686@gmail.com

Umakant Kumar Yadav,
Department of CSE,
MIT School of Engineering,
MIT ADT University,
Pune, India.
umakant65.aims@gmail.com

Prof. Dipti.D.Pawar,
Assistance Professor
Department of CSE,
MIT School of Engineering
MITADT University
Pune, India

***Abstract*— Abstractive method of Text Summarization means generating a short and concise summary that captures the context of the source text in a smaller form. The generated summaries mostly contain new sentences and phrases that may not appear in the source text. Summaries play a vital role for readers who regularly reads and search for documents. We have created a model using Deep Learning and LSTM which generates a short abstractive summary. This process requires a lot of theoretical implementations and challenges which need to be overcome which a discussed in details in this paper.**

Keywords*— Abstractive summarization, Natural Language Processing, Beautiful Soup, Deep learning, LSTM



I. Introduction

TLDR is a common initialism used frequently by readers. It stands for "Too Long; Didn't Read". Typically, people comment TLDR to say that the original post is too long, and they didn't feel like reading it. A short summary of the text is the most effective approach to understand an overall theme of a document. Summarization is a method to reduce the size of the document while preserving its true meaning. It is most helpful in long formatted texts which required longer read time. Summarization can be of two types depending on the basis of whether the exact sentences are considered as they appear in the original text or new sentences are generated using **Natural Language Processing** techniques, they are known as Extractive and Abstractive techniques respectively.

Extractive summarization technique is an extensively researched topic and has been developed for quite a mature level. Now the research has more focus towards the **abstractive summarization**. The complexities underlying with the natural language text and to generate a new shorter text with similar meaning makes abstractive summarization a challenging task.

This **Abstractive Summarizer** is one of the most challenging tasks in Deep Learning and Natural Language Processing, involving good understanding of long phrases, information compression, and logical language generation. The dominant paradigm for training machine learning models to do this is sequence-to-sequence (seq2seq) learning using (Long Short-Term Memory) **LSTM**, where a neural network learns to map input sequences to output sequences. While these seq2seq models were
initially developed using recurrent neural networks, Transformer encoder-decoder models have recently become favored as they are more effective at modeling the dependencies present in the long sequences encountered in summarization.

II. SYSTEM ARCHITECTURE AND IMPLEMENTATION

Abstractive Summarization

In this type of summarization, new sentences are being generated with the help of original text. The new sentences which are formed might not be present in the original text.
Introduction to Sequence-to-Sequence (Seq2Seq) Modelling
Our goal is to design a text summarizer where the input is a long sequence of words and the output is a short summary.**.** Below is a typical Seq2Seq model architecture:
There are two major components of a Seq2Seq model:

- Encoder
- Decoder

Encoder-Decoder Architecture

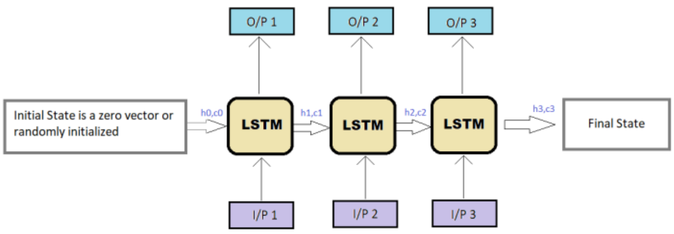
This architecture is used to solve the problems where the input and outputs are of different length.
The input is a long sequence of words and the output will be a short edition of the input sequence.
We have set up the Encoder-Decoder in 2 phases:

- Training phase
- Inference phase

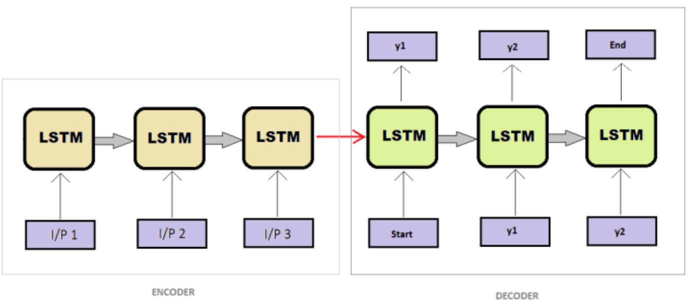
Let's understand these concepts through the lens of an LSTM model.

Training phase
In the training phase, we had set up the encoder and decoder. We have trained the model to predict the target sequence offset by one timestep.

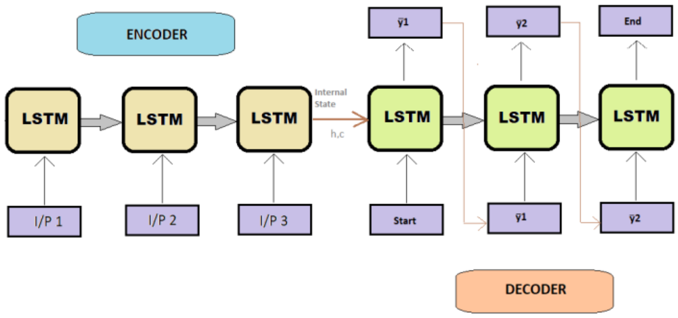
Encoder
An Encoder Long Short-Term Memory model (LSTM) reads the entire input sequence and at every timestep, one word is fed into the encoder, then it processes the information at every timestep and records the contextual information present in the input sequence.



Decoder
The decoder is an LSTM network which reads the entire target sequence word-by-word and forecast the same sequence offset by one timestep. The decoder is prepared to forecast the next word in the sequence given the previous word.



Inference Phase
After training, the model is tested on new source sequences for which the target sequence is unknown.
The Intuition behind the Attention Mechanism



We have taken an example to understand how Attention Mechanism works:

- **Source sequence:** "Which sport do you like the most?
- **Target sequence:** "I love cricket"

The first word **'I'** in the target sequence is connected to the fourth word **'you'** in the source sequence, right? Similarly, the second-word **'love'** in the target sequence is associated with the fifth word **'like'** in the source sequence.
So, instead of seeing all the words in the source sequence, we can expand the significance of specific parts of the source sequence that result in the target sequence. This is the basic idea behind the attention mechanism.
There are 2 classes of attention mechanism depending on the way the attended context vector is derived:

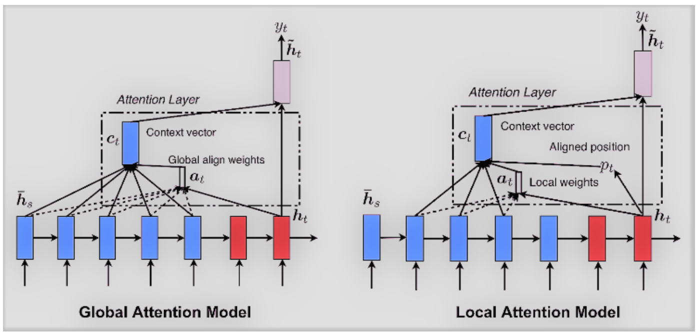
- Global Attention
- Local Attention

.

Global Attention
Here, the attention is placed on all the source positions. In other words, **all the hidden states of the encoder are considered for deriving the attended context vector.**

Source: Effective Approaches to Attention-based Neural Machine Translation – 2015

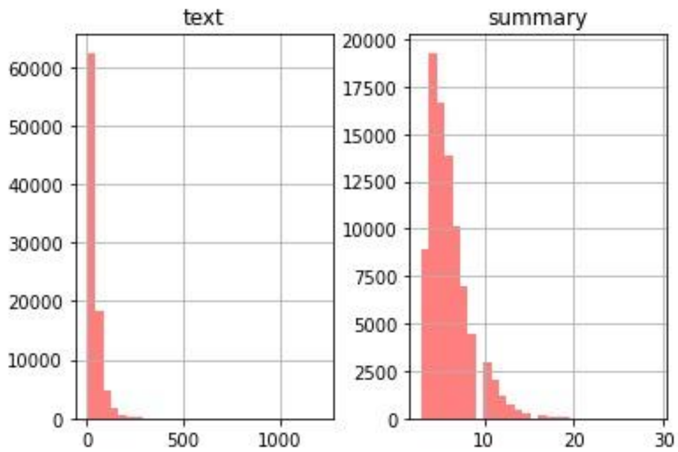
Local Attention
Here, the attention is placed on only a few source positions. **Only a few hidden states of the encoder are considered for deriving the attended context vector.**



III. SYSTEM RESULT

In this model, we analyze the total length of words, reviews and the summary to get an overall stat about the distribution of total length of the provided text. This will help us to set a fixed maximum length of the sequence.
Here we use Python pyplot library to see the comparison graph of the summary and the text generated by the model.

Comparison Output:



Here is a summary generated by the model:

```
Review: used eating flaxseed brownie hodgson mill
brownies super easy wake taste great since like
dark chocolate usually add little cocoa
Original summary: delicious brownie
Predicted summary: best brownie mix
Review: favorite coffee keurig coffeemaker
convenient get amazon cheaper running around
stores trying find lowest price
Original summary: great coffee
Predicted summary: great coffee
```

The fact which makes it Abstractive is that the summary generated by the model does not have common words in the input passage but still both of them convey the same meaning.

IV. CONCLUSION AND FUTURE WORK

In this paper we have discussed the implementation of an effective solution for Text Summarization using Deep learning and LSTM.
The future scope of this project is to increase the training dataset size and build the model. By using larger dataset, the capability of this deep learning model will enhance. Using beam search strategy for decoding the test rather than simple greedy approach (argmax) will result on more logical filtering of words and give better BLEU score.
Bi-Directional LSTM can also be implemented which is capable of capturing the context from both the directions and gives a better context vector as result.

REFERENCES

[1] H. T. Le and T. M. Le, "An approach to abstractive text summarization," 2013 International Conference on Soft Computing and Pattern Recognition (SoCPaR), Hanoi, 2013, pp. 371-376, doi: 10.1109/SOCPAR.2013.7054161.

[2] Ganesan, K., Zhai, C., Han, J. 2010. Opinosis: A GraphBased Approach to Abstractive Summarization of Highly Redundant Opinions. In Proc. of Coling 2010, pages 340–348.

[3] Knight, K. and Marcu, D. 2000. Statistics-based summarization - step one: sentence compression. In Proc. Of AAAI 2000.

[4] Lloret, E., Palomar, M. 2011. Analyzing the Use of Word Graphs for Abstractive Text Summarization. In Proc. of IMMM 2011.

[5] S. Syed, Abstractive Summarization of Social Media Posts: A case Study using Deep Learning, Master's thesis, Bauhaus University, Weimar, Germany, 2017.

[6] A. Khan and N. Salim, "A review on abstractive summarization methods," Journal of @eoretical and Applied Information Technology, vol. 59, no. 1, pp. 64–72, 2014

[7] E. Baralis, L. Cagliero, A. Fiori, P. Garza, "MWI-Sum: A Multilingual Summarizer Based on Frequent Weighted Itemsets", ACM Transactions on Information Systems, vol. 34, no. 1, pp. 1-35, 2015

[8] Meena, Yogesh & Gopalani, Dinesh. (2015). Domain Independent Framework for Automatic Text Summarization. Procedia Computer Science. 48. 723-728. 10.1016/j.procs.2015.04.207.