OPINION SUMMARIZATION

Report submitted in fulfillment of the requirements

for the Exploratory Project of

Second Year B. Tech

by

Pacha Venkata Sri Harsha

Under the guidance of

Dr. Ravindranath Chowdary

Department of Computer Science and Engineering

INDIAN INSTITUTE OF TECHNOLOGY (BHU) – VARANASI

Varanasi 221005, Uttar Pradesh, India

May 2022

Declaration

I declare that,

- The work contained in this project is original and has been done by myself and under the general supervision of my supervisor.

- The work has not been submitted for any project.

- Whenever I have used materials (data, theoretical analysis, results) from other
  sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.

- Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

Place: IIT (BHU) Varanasi

Date: 03-05-2022

Pacha Venkata Sri Harsha,

B. Tech,

Department of Computer Science and Engineering,

Indian Institute of Technology (BHU) Varanasi,

VARANASI, India – 220005.

Certificate

This is to certify that the work contained in this report entitled "**OPINION SUMMARISATION**" is being submitted by **Pacha Venkata Sri Harsha (20075063)** carried out in the Department of Computer Science and Engineering, Indian Institute of Technology (BHU) Varanasi, is Bonafede of our supervision.

Place: IIT (BHU) Varanasi

Date: 03-05-2022

Dr. Ravindranath Chowdary,
Department of Computer Science and Engineering,
Indian Institute of Technology (BHU) Varanasi,
Varanasi, INDIA 221005.

Acknowledgments

It is a great pleasure for us to express respect and a deep sense of gratitude to my supervisor Dr. Ravindranath Chowdary C, Assistant Prof., Department of Computer Science and Engineering, Indian Institute of Technology (BHU) Varanasi, for his wisdom, vision, expertise, guidance, enthusiastic involvement and persistent encouragement during the planning and development of this work.

I also gratefully acknowledge his painstaking efforts in thoroughly going through and improving the manuscripts without which this work could not have been completed. We are also highly obliged to Prof. Pramod Kumar Jain, Director, Indian Institute of Technology (BHU) Varanasi, and Prof. Sanjay Kumar Singh, Head of Department, CSE for providing all the facilities, help and encouragement for carrying out this exploratory project work. We are also obliged to our parents for their moral support, love, encouragement and blessings to complete this task. Finally, we are indebted and grateful to the Almighty for helping us in this endeavour.

Place: IIT (BHU) Varanasi

Date: 3 May 2022

Pacha Venkata Sri Harsha

Contents

Introduction

The opinion summarization, in which the reviews given by several users, describe a product in a paragraph, was interpreted by the machine to understand how the customer is reacting, like if he was actually loving it or disliking it, and then return a short summary about the product in one line. This program was adopted by many websites so that users who want to see the reviews can look according to their view, whether they want positive or negative reviews.

There are many methods to implement this program. Basically, its methods are classified into Extractive and Abstractive methods. Usually, extractive models are unsupervised where we don't need user input. It doesn't need machine learning algorithms. It uses a ranking algorithm, which had a need for graphs. For Example, HITS, LexRank and PageRank. Another one is the abstractive type method. It requires machine learning and it is basically supervised (user input is required for training the model). Here, neural networks are used. In the case of extractive methods, dangling anaphora is not a big issue, but in the case of abstractive methods, dangling anaphora is considered.

In this project, we are using the **Sequence-to-Sequence** method.

Working Algorithm - Seq2Seq

In this model of opinion summarization, we are going to use the Many-to-Many Seq2Seq model. In this model, we are going to have two major components, namely the encoder and decoder sequences. The encoder and decoder parts are called Long Short Term memory, as they have the capability of capturing long-term dependencies against the problem of vanishing gradient. There are two phases to set the Encoder-Decoder, namely the Training Phase and the Inference phase.

In the training phase, the program reads the input, and each word is fed into the encoder and then captures the contextual information present in the input sequence. The two variables for storing the hidden state and the cell state (h_i and c_i respectively) are used to initialize the decoder. Meanwhile, the program also reads the entire target sequence word by word and the decoder **predicts the next word** in the sequence using the previous word. While predicting the sequence, **sostok** and **eostok** are two tokens representing the start-of-sequence token and end-of-sequence token respectively, where sostok is passed to start the prediction and eostok is used to stop the prediction.

After the training, the model is now ready to predict the sequence where the target sequence is not given. For that, the inference phase is set up to decode the test sequence by

using the trained data set. Here also, the input is taken and each word is fed into the encoder and initialize the decoder using the hidden state and the cell state. sostok is then passed into the decoder and then using the internal states, the decoder outputs the probability of the next words which may come and then selects the word which has the most probability. Now pass the output sample word and pass into the decoder and update the internal states. This will form a loop, and the decoder stops itself when eostok is passed finally or the length of the predicted sequence has reached its maximum.

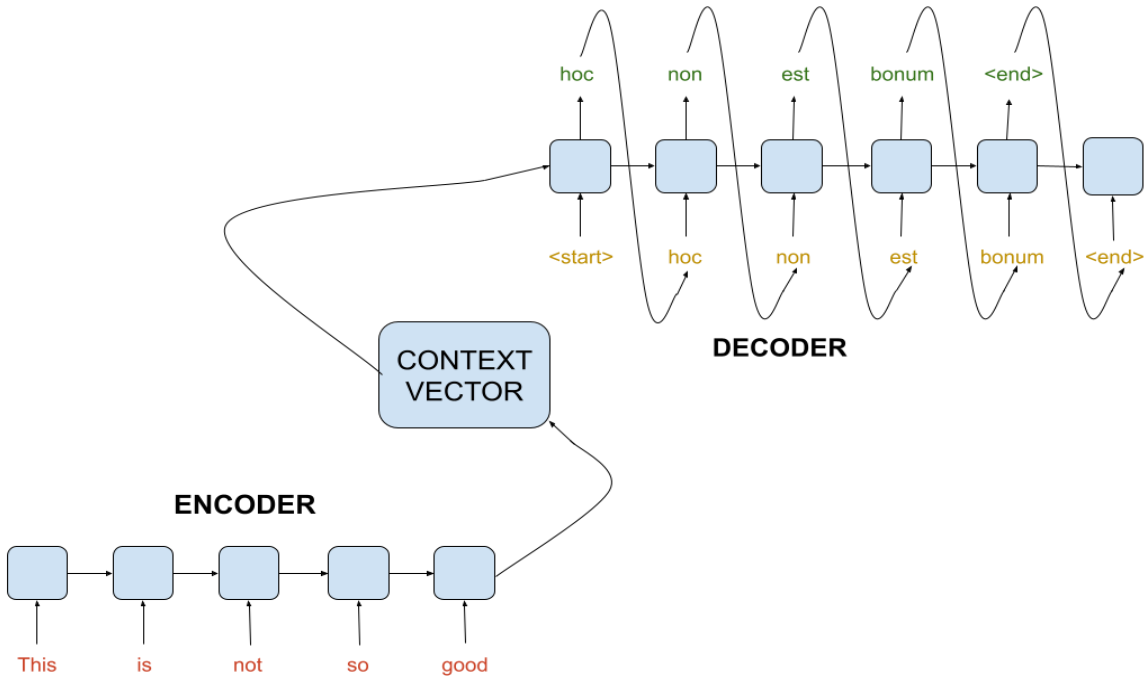

Figure 1 - Encoder-Decoder Architecture

As we know that the encoder updates its internal state values by using the internal state values from the previous

step. This shows the internal state values have some recursive properties.

For encoder LSTM, the formula for the hidden state is given as:

$$h_t = f(W^{(hh)}h_{t-1} + W^{(hx)}x_t)$$

where,

h_t is the hidden state for the present state,

h_{t-1} is the hidden state for the previous state,

x_t is the input vector value,

$W^{(hh)}$ and $W^{(hx)}$ are some appropriate weights.

In the decoder, there will be several recurrent units, where each recurrent unit will be taking the previous hidden state and outputs the predicted word as well as the new hidden state.

Hence the formula for the hidden state will be given as:

$$h_t = f(W^{(hh)}h_{t-1})$$

We can see that the hidden state is only dependent on the previous hidden state.

For the output layer, we are going to input our current hidden state with the ***softmax*** function, which produces the probability distribution from the vector of values with a target class of high probability.

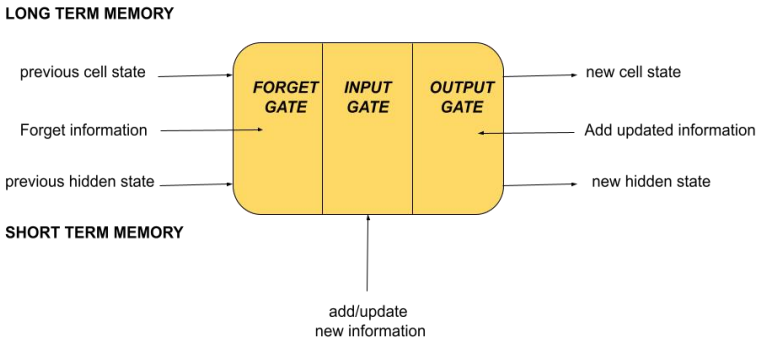

Figure 2 - LSTM Gates

Hence,

$$y_t = softmax(W^S h_t)$$

Now in the brief of LSTM, which is a recurrent neural network, it consists of three parts/gates. They are forget, input and output gates. Forget gate does forgets/erases irrelevant information, input gate does add/update the state values and the output gate does pass the updated state values.

Conclusion

The LSTM of the encoder-decoder model has its huge necessity in the modern present daily life applications, which make our life easier than you think and more efficiently, some of them are,

- Machine language translation
- Chat Bots
- Automatic image caption generator
- Automation application systems
- Text Autofilling

We have used some temporary variables like hidden state value and cell state value, which it is dependent on the previous value, and passing itself into the function again, hence making the nature of the function be recursive and also reflecting the dynamic nature of itself. The best thing is that if a word is given, the context of the word is understood by the machine and keeping a track that how a word is connected to other one in the terms of probability and making the final decision.

References

- Babar, Samrat & Tech-Cse, M & Rit,. (2013). Text Summarization:An Overview.
- Kandoor, Arun. (2021). Tiny Neural Models for Seq2Seq.
- S., Keerthana & R., Venkatesan. (2020). Abstractive Text Summarization using Seq2seq Model. International Journal of Computer Applications. 176. 24-26. 10.5120/ijca2020920401.
- Bhore, Vivek & Bondare, Pratik & Gawande, Rutik & Guntiwar, Vrushabh & Kale, Priti. (2022). Extractive Text Summarization. International Journal of Advanced Research in Science, Communication and Technology. 154-159. 10.48175/IJARSCT-3022.
- A Simple Introduction to Sequence-to-Sequence Models
- Understanding Encoder-Decoder Sequence to Sequence Model
- Encoder-Decoder Models for Natural Language Processing
- Automatic Text Summarization with Machine Learning — An overview
- **JASON BROWNLEE –** Develop Deep Learning Models on Theano and Tensorflow using Keras.
- **JASON BROWNLEE –** Understand your data, Create accurate models, and Work Projects End – to – End.