Text Summarization using Deep Learning

Priscilla Sugumar
1451 Portofino Point
Orlando, FL-32765
+14074060592
sugumarpriscilla@gmail.com

Rashi Jain
1451 Portofino Point
Orlando, FL-32765
+13212023015
rashijain013@gmail.com

Swati Lathwal
12221 E Col Dr
Orlando, FL-32826
+14076402524
swati.lathwal@gmail.com

1. ABSTRACT

In this paper, we describe an effective technique for text summarization. Text Summarization is the process of automatically producing a summary of the given document. The main approaches include extractive and abstractive techniques. Abstractive text summarization generates the summary utilizing words outside the given document. The objective of our project is to create a concise and fluent summary for the Amazon Fine Food Reviews dataset by using the abstraction based approach.

2. INTRODUCTION

It is a well-known fact that client feedback plays an essential role in developing strategic marketing plans for various online websites. The textual data is multiplying exponentially with an ever-increasing quantity of customer evaluations. Reading the whole text document to get an idea behind the text is a time-consuming task. This makes extraction of usable information from the humongous customer reviews to deliver actionable insights a tedious process. While analyzing each review is complicated, the abstractive text summarization technique can result in a concise and fluent summary of the customer reviews.

The sequence modeling technique can predict future outcomes based on currently available data information. Sequence modeling is not just limited to sentiment analysis. It works on almost all types of data formats, like speech recognition, DNA sequence analysis, and even video activity recognition is possible through sequence modeling. In this project, we implement the abstractive text summarization approach, where we use the deep neural network to identify the critical sentences and form a new summary to understand the context of the large documents.

3. RELATED WORK

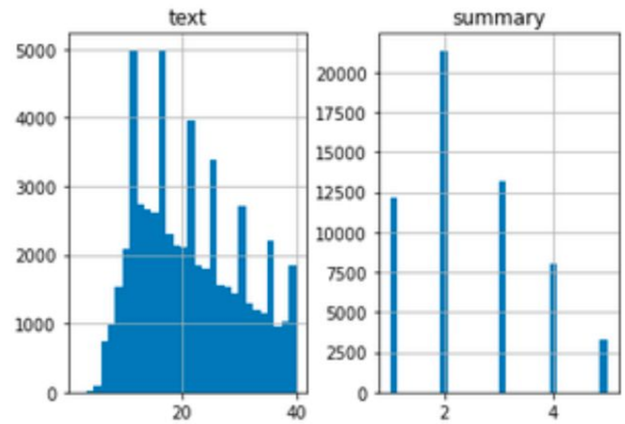
Several mentionable techniques have been used previously, such as word to vector. The vector representation of the text gives us the key points of the text and finds similarity or dissimilarity. The traditional approach for abstractive text summarization uses a rule-based method that uses manually compiled features. This demands a depth of knowledge about the text-domain to come up with the human compiled features. Several structure-based approaches use predefined structures to be used as templates, tree-based structure, ontology-based structure, lead, body phrase structure, and rule-based structure. Semantic-based approaches involve using the semantic representation as input into NLP algorithms to obtain document summarization. Some of the semantic models are the Multimodal semantic approach, Information item-based method, and semantic graph-based method.

4. PROBLEM STATEMENT

The amount of data in the form of text multiplies exponentially with the ever-increasing quantity of customer evaluations. It is a very tedious and time-consuming task to read all of the text available on the internet. It is impossible to read the whole data manually to provide actionable insights and make some sense out of the available raw form of the data. The process of Abstractive text summarization will give us a fluent summary of the available customer reviews on the Amazon food market.

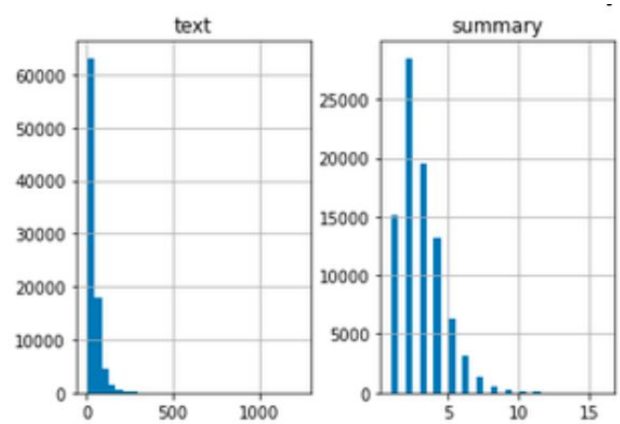
5. EXPLORATORY DATA ANALYSIS

The exploratory data analysis on the Amazon food market customer reviews provides a fascinating insight into the data. As we can see from the figure, a little less than ninety percent of the text data contain the number of words equal to a hundred. In the Summary data, most of the text contains words equal to five.



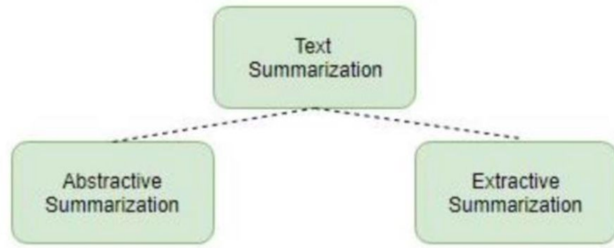
6. FILTERED DATA

We performed filtration of the dataset so that the review text contains a forty-word count and the summary contains only a five-word count.



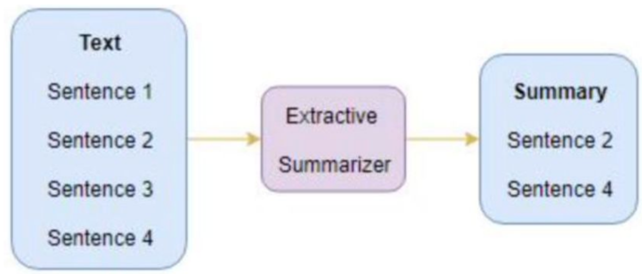
7. TEXT SUMMARIZATION

Text summarization is done to extract meaningful and essential information from the text. The technique of text summarization is used to produce concise and informative text. This process focuses on preserving critical information and preserving the meaning in the text. It is a time-saving technique at the same time, preserving the overall meaning of the text data. There are two approaches for the text summarization:



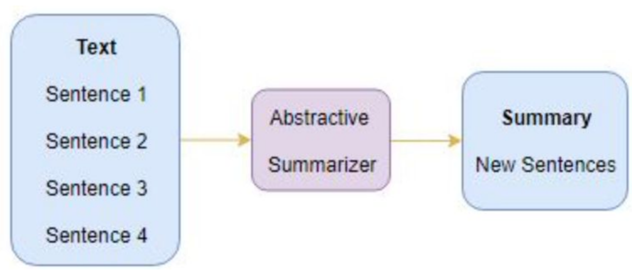
(a) Extractive Summarization

The important sentences and phrases are identified from the original piece of text. Example:



(b) Abstractive Summarization

This approach is the opposite of the extractive summarization technique. As the raw texts are manipulated so that the overall meaning is extracted from the original text. Example:

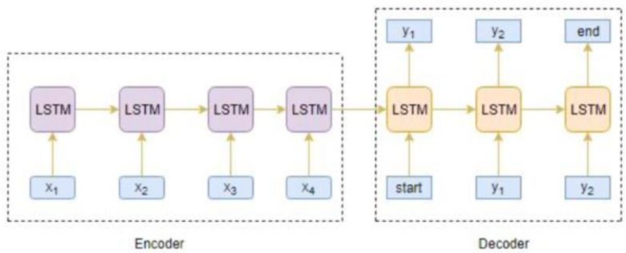


From the figure above, we can see that new sentences are created which contain the overall summarized meaning of the text.

8. SEQUENCE-TO-SEQUENCE MODEL

This technique is effective on sequential information. This type of modeling predicts the output sequence for a given input sequence. The uses of Sequence-to-Sequence modeling are Sentiment classification, Neural Machine Translation, Named Entity Recognition. The key steps which were done in the modeling are as follows:

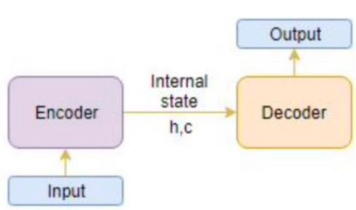
- The long word sequence of customer review is fed as input in the Encoder.
- The output is the short word sequence of summary.
- The data was modelled as a Many-Many Seq2Seq problem.



The objective of Sequence-to-Sequence modelling is to build the text summarizer where the input is the sequence of information.

8.1. ENCODER AND DECODER ARCHITECTURE

The Sequence modeling architecture has two components Encoder and Decoder. This type of architecture is used when the length of the input and output sequence is different. The input will be a long sequence of words, and the output will be a short version of the input sequence.

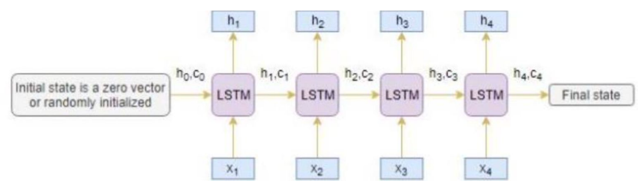


From the figure above, we can see that the output of the decoder is the shorter and concise version of the input fed in the input of Encoder. This is how the architecture works. The Variants of RNN like Gated Recurrent Neural or LSTM are preferred as the encoder-decoder components. In our model, we use the Long Short-Term Memory model (LSTM) as our encoder and decoder approach. There are two phases in the Encoder-Decoder phases:

8.1.1. Training Phase

In the training phase, the Encoder and Decoder were set up, and the model was trained in order to predict the target sequence offset by one timestep.

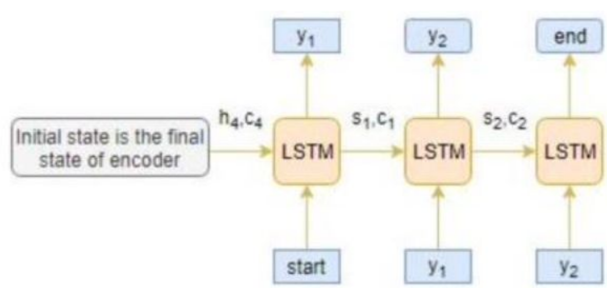
Encoder: The Encoder LSTM reads the entire input sequence at each time step t, where the information is processed at every timestep. The encoding process will capture the contextual information from the input.



The figure above contains the following:

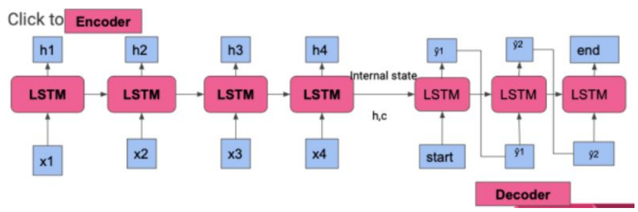
- Hidden state (h_i)
- Cell state (c_i) of the last time step were used for initializing the decoder. This is because the encoder and decoder are two different sets of the LSTM architecture

Decoder: In the Decoder phase, the LSTM network reads the target sequence word-by-word and predicts the next word in the sequence based on the given previous word as input.



From the figure above the **<start>** and **<end>** are the special tokens. These tokes are added to the target sequence before feeding as input into the decoder. The target sequence is not known during the process of decoding the test sequence. We begin to predict the target sequence by feeding the first word into the decoder, and that is always the **<start>** token—the **<end>** token signals the sentence ends.

8.1.2. Inference Phase



The inference phase decodes the test sequence in the following steps:

1. The entire input is encoded and sequence.
2. Decoder is initialized with internal states of the Encoder.
3. **<start>** token is pressed that works as an input to the decoder.
4. Decoder is run for each timestep.
5. The word with the maximum probability will be selected as the output.
6. The sampled word is passed to the decoder as the input in the next timestep, and internal states are updated with the current time step.
7. The above process is repeated from steps 3 – 5 until <end> token is generated or the maximum length of the target sequence is achieved

8.2. LIMITATION OF THE ENCODER-DECODER ARCHITECTURE

(a) The Encoder and Decoder only work for the short sequence modeling.
(b) The Decoder is incapable of memorizing the long sequence in the length of the fixed vector.

The Potential issue that we face with the architecture of Encoder and Decoder is that the neural network should be able to compress the critical information from the input into a fixed-length vector format. This is not compatible with the long sequence resulting in performance deterioration as the input length of sentences increase. To overcome this problem, we have used the mechanism called Attention Mechanism.

9. ATTENTION MECHANISM

To make the model work more effective we used the third-party attention model. In the attention mechanism we give importance to some parts of the source sequence that results in the target sequence.
For Example:

Source sequence: "Which sport do you like the most"

Target sequence: "I love cricket"

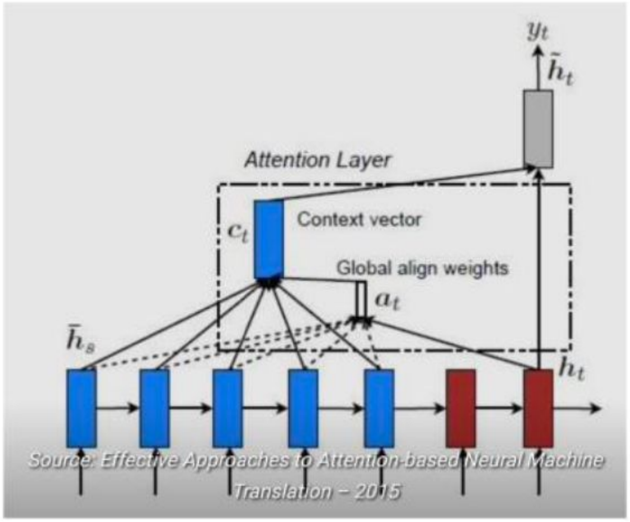
The first word **'I'** in the target sequence is connected to the fourth word **'you'** in the source sequence. Similarly, the second word **'love'** in the target sequence is associated with the fifth word **'like'** in the source sequence.

9.1 KEY INTUITION BEHIND ATTENTION MECHANISM

Before proceeding to the summarization, it is essential to understand the text not to lose any information. For that part, we must know the weightage of each text and how much attention needs to be paid to the text. After understanding the amount of attention to be paid, the focused text is generated at the timestamp T. The attention mechanism works on the basic idea, which is dependent on the two classes:

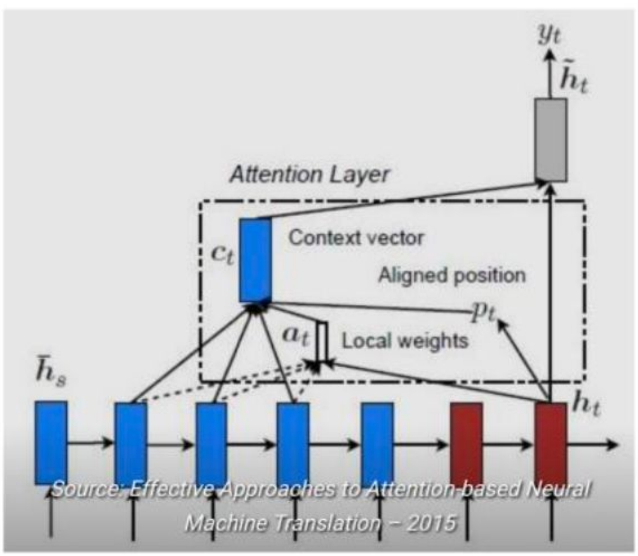
- **Global Attention**

  Global attention takes the context of all words before generating the summary.



- **Local attention**

  In this type of attention, only a few numbers of hidden states are considered to derive the attended vector context.



10. EVALUATION METRIC

The abstractive text summarization technique generates a summary using words that are not available in the document, preserving the semantic meaning of the text. An evaluation of such a technique is a complicated procedure and requires a separate model to check for text similarity. Therefore, we are planning to perform a human evaluation of the model currently. Additionally, we would keep exploring any efficient automatic evaluation algorithms that would be a good approximation of human evaluation.

11. CONCLUSION

The Abstractive Summarization was performed on the Amazon Fine Food Reviews dataset. The techniques used were as follows:

i. Encoder-Decoder model(Seq2Seq Model)

ii. LSTM (Long Short Term Memory)

iii. Attention Mechanism

In the end, the performance of a machine-generated summary was evaluated against a human-generated summary.

12. FUTURE WORK

Future work can be done on the dataset, which is bigger in size than the one used in our project. The bigger the data size, the better the output can be predicted. Also, the distributed processing can be implemented in future work in order to cut some more time from the existing time that was used. The Bi-Directional LSTM can be a promising and better technique to be implemented on this type of text summarization. The Beam Search Strategy for Decoder can be an effective strategy for future work.

13. REFERENCES

[1] Bowman, M., Debray, S. K., and Peterson, L. L. 1993. Reasoning about naming systems. *ACM Trans. Program. Lang. Syst.* 15, 5 (Nov. 1993), 795-825. DOI= http://doi.acm.org/10.1145/161468.16147.

[2] Ding, W. and Marchionini, G. 1997. *A Study on Video Browsing Strategies*. Technical Report. University of Maryland at College Park.

[3] Fröhlich, B. and Plate, J. 2000. The cubic mouse: a new device for three-dimensional input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (The Hague, The Netherlands, April 01 - 06, 2000). CHI '00. ACM, New York, NY, 526-531. DOI= http://doi.acm.org/10.1145/332040.332491.

[4] Tavel, P. 2007. *Modeling and Simulation Design*. AK Peters Ltd., Natick, MA.

[5] Sannella, M. J. 1994. *Constraint Satisfaction and Debugging for Interactive User Interfaces*. Doctoral Thesis. UMI Order Number: UMI Order No. GAX95-09398., University of Washington.

[6] Forman, G. 2003. An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.* 3 (Mar. 2003), 1289-1305.

[7] Brown, L. D., Hua, H., and Gao, C. 2003. A widget framework for augmented interaction in SCAPE. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology* (Vancouver, Canada, November 02 - 05, 2003). UIST '03. ACM, New York, NY, 1-10. DOI= http://doi.acm.org/10.1145/964696.964697.

[8] Yu, Y. T. and Lau, M. F. 2006. A comparison of MC/DC, MUMCUT and several other coverage criteria for logical decisions. *J. Syst. Softw.* 79, 5 (May. 2006), 577-590. DOI= http://dx.doi.org/10.1016/j.jss.2005.05.030.

[9] Spector, A. Z. 1989. Achieving application requirements. In *Distributed Systems*, S. Mullender, Ed. ACM Press Frontier Series. ACM, New York, NY, 19-33. DOI= http://doi.acm.org/10.1145/90417.90738.