Abstractive Text Summarization

Dr.J.Anitha[1], M.Raahavi[2], M.Rehapriadarsini[3], S.S.Sudarshana[4]
[1]*Associate Professor,* [2, 3, 4] *Student, Department of Information Technology,*
Sri Ramakrishna Engineering College, Coimbatore, India.

Abstract:
In the current modern Internet age, the textual data is ever increasing day-by-day. We need some ways to condense this massive data while preserving the information as well as the meaning. We need to summarize textual data for that. Abstractive text summarization is the method of automatically generating précised summaries from an input document while retaining the important key points in the actual content. It will help in easy and quick retrieval of information. Text summarization is the process of generating a brief version of a single document or document set, so that the resulting summaries preserve the fundamental information of the original documents. This study addresses motivations regarding automatic summary generation of any large input texts.

Index Terms – Abstractive text summarization, Deep Learning, Encoders and Decoders, Long Short-Term Memory, Natural Language Processing, Neural Networks, Word embedding.

I.INTRODUCTION

Text summarization is the process of generating a crisp set of text that preserves the salient information of the input texts. The proposed system helps to reduce the input text size by building a summary that has the most important ideas in that input document and can give a better understanding as well as lot of information in very short time.[1] The approach comprises of two major steps: Generation of good quality labeled data by exploiting the summary present as headnotes section, utilizing such labeled data in order to extract the key sentences to be added in the generated summary. Text summarization refers to the technique of minimizing long pieces of text sentences. The aim is to create a fluent summary which has only the main points outlined in the input document. This is the process of shortening a set of data computationally, to create a subset that contains very important or relevant information in the original content. The purpose of text summarization is that it reduces time, makes the selection process faster, reduces the number of lines and creates a readable summary. [2] There are two prominent types of summarization algorithms. Extractive summarization systems generate summaries by replicating certain parts of the source text through some metric of importance and then combining those sentences together to render a final precised summary.[3] Abstractive summarization systems generate new phrases, by rephrasing or using strings that were not in the original text thereby generating efficient summary. [4]

II.PROBLEM STATEMENT

The data is in the internet is unstructured in various sources like webpages, news, blogs, articles, status updates etc. And the way to navigate it is to use for search and skim the results is very tedious. Because of increasing amount information in the internet, researches are gaining more and more attention among the researchers. There is a great necessity to reduce much of the data to shorter the focused summaries that spots the salient details, so that anyone can navigate it effectively as well as check whether the larger data contain the information that people are looking for. In text summarization, one of the most difficult problems is to cover all topics in the text. There are many methods to summarize large data by finding topics of the input text first and scoring the individual sentences with respect to the topics but however it is a time-consuming process. To overcome this problem, abstractive text summarization is essential to tackle the overloading of data.

III.OBJECTIVE

The objective of the project is built on the concept of deep learning for abstractive summarizer. The concept of deep learning

algorithm trains the machines with some sample data which makes it capable of producing recapitulation of input text. Neural sequence-to-sequence model has provided a feasible approach for abstractive text summarization. This project work uses a Long Short-Term Memory (LSTM) sequence-to-sequence attention model. The method utilizes a local attention model for generating every single word of the generated summary conditioned on the input sentences. The model is structurally uncomplicated that it can quickly be trained end-to-end and scales to large volumes of the training data. The reconstructed paragraph or the concise summary is evaluated using standard metrics like ROUGE, showing that neural models can encode texts in a way that preserve syntactic, semantic, and discourse coherence. Neural networks are effective in solving almost any machine learning classification problem. Important parameters required in defining the architecture of neural network (NN) are number of hidden layers to be used, number of hidden units to be present in each layer, activation function for each node, error threshold for the data, the type of interconnections, etc. neural networks can capture very complex characteristics of data without any significant involvement of manual labor as opposed to the machine learning systems. Deep learning uses deep neural networks to learn good representations of the input data, which can then be used to perform specific task.

IV.METHODOLOGY

4.1 Recurrent Neural Network (RNN)

Recurrent Neural Networks are gaining popularity from advances to the networks designs and increasing computational power in graphic processing and its units. RNNs are especially helpful in sequential data since each neuron or unit can use its internal memory to maintain actual information about the previous input. This allows the network to gain an in-depth understanding. This is important because reading through a sentence even as a man power, selecting the context of each string from the strings before it. An RNN has loops in them that allow information to be carried through neurons while reading the input. RNNs actually are able to handle context from the starting node of the sentence that will allow more accurate predictions of a string at the end node of a sentence. However, this isn't necessarily applied for vanilla RNNs. This is why RNNs faded out from practice for a while until some great results were achieved with using a Long Short-Term Memory unit inside the Neural Network.[5]

4.2 Long Short-Term Memory (LSTM) Units

The LSTM is RNN architecture can remember the past contextual values and texts. This stored value does not change over time while the model is being trained. There are four components in LSTM which are LSTM Units, LSTM Blocks, LSTM Gates and LSTM Recurrent Components. LSTM Unit will store the values for a very long time or for very short time. LSTM has no activation functions for their recurrent components. Since there are no activation function the values of units never changes for some period until the context is being changed. LSTM's are considered as deep neural networks. These LSTM's are implemented in parallel systems. LSTM blocks have four gates to control the information flow. Logistic functions are used to implement these gates, to compute a value between 0 and 1. To allow or deny information flow into or out of the memory, multiplication of values with these logistic functions is done. For controlling the flow of new values into memory, input gate plays key role. The level to which a value remains in memory is controlled by forget gate. Output gate will control the range to which the value in memory is used to compute the output activation. Sometimes in implementations, the input and forget gates are merged into a single gate, so it is possible to see even 3 gate representations of LSTM. When new value that is worth remembering is available then we can forget the old value. This represents the combining effect of input and forget gate of LSTM.

4.3 Encoders and Decoders

The architecture of encoder-decoder model basically comprised of two components: one for reading the input sequence and encoding them into a fixed length vector and another is for decoding the fixed length vector and then forwarding the predicted sequence. The application of this model in concert gives the architecture its name of Encoder-Decoder LSTM applied specifically for seq2seq applications. The Encoder-Decoder LSTM was developed specially for natural language processing (NLP) problems where it demonstrated state-of-the-art performance,

especially in text translation called statistical machine translation.[6]

4.4 Word Embedding

The abstractive text summarization project uses ConceptNet Numberbatch word embedding. ConceptNet Numberbatch is section of the ConceptNet open data project. ConceptNet gives a lot of ways to compute with word meanings, one of those is word embeddings. ConceptNet Numberbatch is a kind of snapshot of the word embeddings. It is basically built on an ensemble which combines data from ConceptNet, word2vec, GloVe, and OpenSubtitles 2016, using a variation on retrofitting.

V. EXPERIMENTAL ANALYSIS

For abstractive text summarization, the ConceptNet Numberbatch word embedding similar to GloVe is used. The news dataset has been taken from kaggle.com. The dataset is in the format of .csv and it is highly skewed. It contains 29451 records with 6 variables such as Author, Date, Headlines, Read_more, Text, cText. The generated summary is evaluated using the ROUGE score which provides a value of 0.5106. The precision and recall values are 0.4 and 0.705 which shows that the developed model provides better result with the developed algorithm (LSTM).

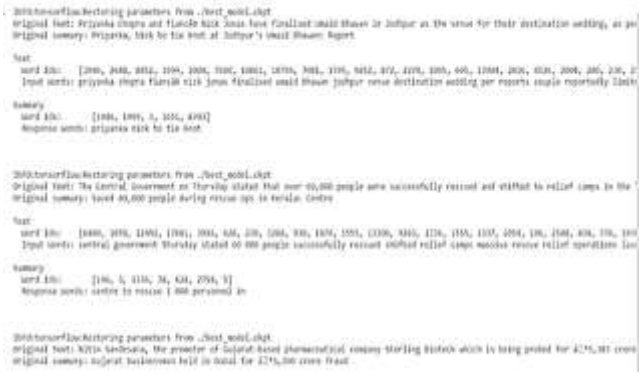

Fig: 1 Summary generated using test data

Fig: 2 Evaluation Measure

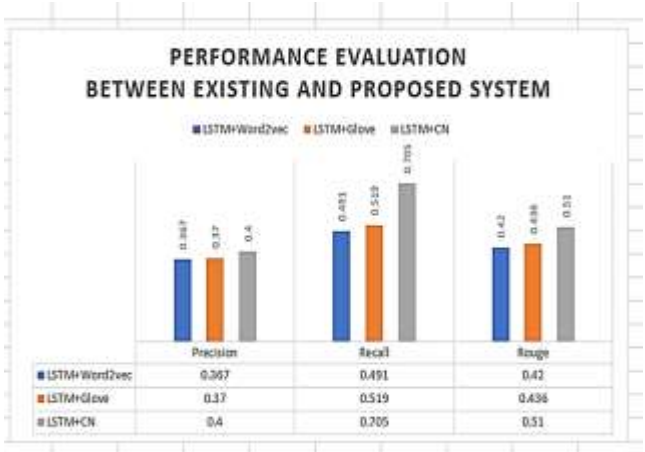

Fig: 3 Performance Evaluation

VI. CONCLUSION AND FUTURE SCOPE

The system aims at implementing a state-of-the-art model for abstractive sentence summarization to a recurrent neural network architecture. The model is a simplified version of the encoder-decoder framework for machine translation. The model is trained on the News_summary corpus to generate summaries of news based on the contents in the text part. Evaluation measures are implemented to enhance the extractive text summarization by improving: summary representation, sentence ranking, and sentence selection process. However, Abstractive text summarization is a challenging area because of the complexity of natural language processing.

Future work can be executed in continuing the study of abstractive text summarization to improve the current algorithms. After reviewing the results produced by abstractive text summarizer, it can be concluded that sometimes repetition of words occur when they deal with large text documents. To handle such large documents more research work is needed by using advanced deep learning techniques. There is a need to propose new methods while considering minute details in massive. To achieve this, a hybrid model that is both able to handle enormous dataset and the real time summary generation problem can be included to have a response during large data processing with an improved accuracy.

REFERENCES

[1]L.D.S. Cabral, R.D. Lins, R.F. Mello, "A platform for language independent summarization", *In Proceedings of the 2014 ACM Symposium on Document Engineering*, 2014, pp.203-206.

[2]M. Gambhir, V. Gupta, "Recent automatic text summarization techniques: A survey", Artif. Intell. Rev. (2016) 1-66.

[3]S. Polsley, P. Jhunjhunwala, R. Huang, "Casesummarizer: A system for automated summarization of legal texts", In: *Proceedings of COLING 2016, 26t International Conference on Computational Linguistics: System Demonstrations*, pp: 258-262.

[4]Y.C. Chen, M. Bansal, "Fast abstractive summarization with reinforce-selected sentence rewriting", arXiv preprint arXiv:1805.11080.

[5]Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Ben, "Neural machine translation by jointly learning to align and translate", CoRR, abs/1409.0473, 2014.

[6] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning, "Effective approaches to attention based neural machine translation", *In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September.

[7]Abeer Alzuhair, Mohammed Al-Dheelan, "An approach for combining multiple weighting schemes and ranking methods in Graph-based Maulti-document summarization", Vol.7,Sep 2019.

[8]Begum Mutlu, Ebru A.Sezer, M.Ali Akcayol, "Multi document extractive text summarization: A comparative assessment on features", Jul.2019, pp.1-13.

[9]J.Mohan, M.Sunitha, A.Ganesha, Jaya, "A study on Ontology based Abstractive Text Summarization", *Fourth International Conference on recent trends in Computer Science and Engineering*, 32-37,India, 2016.

[10] I F Moawad, M Aref, "Semantic Graph Reduction Approach for Abstractive Text Summarizaton", *IEEE International Conference on Natural*, 132-138, Egypt 2012.