Text Summarization using Sequence to Sequence LSTM

[1]Srushti Wadekar
Computer Science
Lakehead University
Thunder Bay, Canada
swadekar@lakeheadu.ca

[2]Komal Barge
Computer Science
Lakehead University
Thunder Bay, Canada
bargek@lakeheadu.ca

[3]Nidhi Patel
Computer Science
Lakehead University
Thunder Bay, Canada
pateln@lakeheadu.ca

[4]Shivani Singh
Computer Science
Lakehead University
Thunder Bay, Canada
ssingh73@lakeheadu.ca

***Abstract*—Text summarization is a process to help user to extract and collect relevant information from the vast amount of data available and present that information to the user in the form of a summarized text. Summarization of text shortens the whole text document into a newly evaluated form by keeping in violating the key meaning of the original text document. Nowadays, many applications such as search engines, discussion forums, business analysis and news articles need summarization of the text which will help users to gain more important information within less time. With the amount of textual information present on the internet, text summarization domain is becoming very important in the field of natural language processing. Our model represents abstractive text summarization using LSTM (Long-Short term memory network) based deep learning as a sequence-to-sequence model noting that this is many-to-many problem. Textual data is learned by using word embedding on the news article dataset to pertain the original meaning of the article. This paper contains the sequence to sequence model for text summarization which has given around 84% accurate summary for given text. The design of model and experimental analysis is further discussed in the paper.**

***Index Terms*—Summarization, LSTM, Embedding, Abstractive**

I. Introduction

Summarization systems is used to determine extremely important subjects of document(s) which habitually have extra evidence. Generation of summary from the textual data in theory contains phrases and sentences that may not appear in the source text. It performs the task to concise and shorten the text to capture salient features based on semantic knowledge from the source text to a newly framed sentence. In the current era, textual information on the internet is growing at an exponential rate, because of which information abstraction and summarization has become so important and timely tool for user to quickly analyse the enormous amount of information available. With the amount of textual information present on the internet, text summarization domain is becoming very important in the field of natural language processing. Manual summarization done by humans requires understanding, and reading of an article, website or document to find key points. These key points can be used to generate new sentences which forms the summary. Text summarization tools can help users to grasp main concepts of information sources in a less time, improve the content of a summary, reduce its redundancy and keeps a decent compression rate.

Our model summarises news articles to descriptions or headlines from different news source like Hindu, Indian Times and Guardian. Training model to return sequences of words from a domain to another domain by holding its meaning is termed as sequence-to-sequence learning. This kind of model, i.e., sequence to sequence model directs to map a fixed length input into a fixed length output where the length of input and output may vary. It contains of three parts encode, decoder and intermediate vector. Encoder stacks of a few LSTM units where each acknowledges a solitary component of the input sequence, gathers data for that component and spreads it forward. For Decoder, heap of a few LSTM units where each predicts a yield y at a time step t. Each intermittent unit acknowledges a hidden state from the past unit and creates and yield just as its own hidden state. Encoder vector means to exemplify the data for all input components to enable the decoder to make precise predictions. LSTM's a special kind of RNN, is capable of learning and remembering long term dependencies for treating simple RNN. They are dependent over previous cell state, previous hidden state and input at the current timestamp. The objective of our model is to build text summarizer in which we can consider input as long sequence of words and the output as short summary that is also in terms of sequence.

The pre-trained word embedding is also used in our sequence-to-sequence model. GloVe, algorithm is an extension to the word2vec method developed by Pennington, et al. at Stanford for the efficient learning of word vectors. GloVe is an approach for combining the global statistics of matrix factorization techniques such as LSA in word2vec with local context-based learning. Rather than using a window to define a local meaning, GloVe uses statistics across the entire text corpus to construct an explicit word-context or word-co-occurrence matrix. The outcome is a learning model which can usually lead to better embedding of words [8].

II. Related Research Work

From the previous years, there are numerous text summarization approaches have been created. Different methodologies which have been executed are graph based summa-

rization, AI based models, algebraic strategies. Summarization is likewise done by extraction of essential sentences and content highlights, for example, phrases, watchwords, inquiries, word position and word recurrence. A measurable methodology where a classifier or applicable data is taken from lexicons or WordNet with different techniques has likewise been executed. There are various challenges remarked by abstractive summarization such as generation of insignificant and general summary often involving high-frequency phrases, generation of summaries which have restricted grammaticality and readability and recent studies used sequence-to-sequence models for prediction of the later word in summary.

According to Ramesh Nallapati et al in 2016 in RNN, to reduce the size of the soft-max layer of the decoder which is the main computational bottleneck, the decoder-vocabulary of each mini-batch is restricted to words in the source documents of that batch and the most frequent words in the target dictionary are added until the vocabulary reaches a fixed size [1]. To overcome identification of key concepts and key entities in the document, capture of some linguistic features such as parts-of-speech tags, named-entity tags, and TF and IDF statistics of the words by creating look-up based embedding matrices is done. To create the matrix, continuous feature like TF and IDF are converted into categorical values which allows us to map them into the embedding matrix. The authors used attentional encoder-decoder for abstractive summarization with promising results, outperforming state-of-the-art results on different datasets. But, each of the models proposed by them contained some issue in text summarization [1]. In order to face the problem of unseen (out-of-vocabulary) words, incorporating a pointer generator network in their system attention mechanisms were used to train encoder-decoder models.

Panagiotis Kouris, Georgios Alexandridis, Andreas Stafylopatis proposed an approach that tries to bridge the combination of deep learning models of encoder-decoder architecture and semantic-based data transformations gap by introducing a framework that combines the potential of machine learning with the importance of semantics [4].They optimized parameters of deep learning model to have effective handling of OOV (out-of-vocabulary) and rare words which improves the performance of text summarization. Main three components of the framework are model for extraction of taxonomy of concepts from input text, encoder-decoder deep learning model, approach to makeover generalized summary into human readable form containing principle information.

In recent years Twitter data mining has been a hot research subject. The essence of the data gathered differs considerably according to the purpose and intended outcome. The methods for manipulating data and collecting the information needed are also different. A method to m easure public opinion transition over time and to classify the news that contributed to public opinion breakdowns was proposed in [1]. In a related context, Sriram et al. proposed a method to classify tweets depending on their natures into a set of classes including private messages, opinions and event, etc [3]. Unigrams, bigrams and adjectives in different ways to classify a set of movie reviews into positive or negative. Recently, new models have been built Gao and Sebastiani proposed a new approach based on the distribution or frequency of the types of sentiment they examine [4]. The writers have found that a quantifier algorithm is a better approximation of the frequency than standard classification-driven algorithms from classification to quantification.

An adversarial process for abstractive text summarization is proposed by Linqing Liu, Yao Lu,Min Yang,Qiang Qu,Jia Zhu,Hongyan Li, where their model concurrently trains a generative model and a discriminative model [3]. The generative model acts as an agent to emphasise learning by taking raw input and predicts the abstractive summarisation whereas discriminative model efforts to distinguish the emphasised generative learning from true summary. The generative model uses a bidirectional LSTM encoder for the conversion of input sequence to the hidden sequence. They randomly selected 50 test samples and achieved the best ROUGE 1 and ROUGE 2.

Beforehand extractive summarization was an extensively used research topic which had reached its maturity state. Currently documenting the progress for abstractive summarization, we realise the underlying complexities. Som Gupta and S. K Gupta achieved a comparable performance to deep learning approaches by utilizing semantic based data generalization for the enhancement of sequence-to- sequence (seq2seq) [6]. They have listed various tools and evaluated techniques being used for assessing the abstractive summaries. Finally, Shengli Song, Haitao Huang, and Tongxiao Ruan (2018) proposed a deep LSTM-CNN (convolutional neural network) framework, where the extraction of phrases from source sentences generated summaries [7]. Their experimental results on the datasets CNN and DailyMail show that they achieved competitive results on manual linguistic quality evaluation.

III. Proposed Model

Artificial neural networks (ANN) compose of simple elements called neurons, which can make simple judgments in the area of mathematics. The neurons can together evaluate complicated issues, imitate almost any feature, including very complex, and provide detailed answers. There are three levels of neurons on a superficial neural network: an input layer, a hidden layer, and an output layer. A Deep Neural Network (DNN) includes more than one hidden layer that increases model flexibility and enhances predictive power dramatically. A neural network can perform as a classification model. Artificial neural networks are relatively crude electronic networks of neurons based on the neural structure of the brain. They process records one at a time and learn by comparing their classification of the record (i.e., largely arbitrary) with the known actual classification of the record. The errors from the initial classification of the first record is fed back into the network and used to modify the networks algorithm for further iterations. In this paper, the model is generated using a combination of LSTM and Embedding layers for text summarization using news summary dataset.

The dataset we have used is news summary dataset which contains the details such as author, date, headlines, URL of the article and text. The dataset is generated by gathering the news from the articles of Hindu, Indian Times and Guardian between the time period of February to August of year 2017. From the dataset, we have taken text as input for our model to predict the summary and the headlines from the dataset is used as ground truth which contains summarized version of the text.

The text and summaries from the dataset are preprocessed through regular expression library which is "re". Also, URLs like "https://xyz.abc.net/search/" is replaced with "xyz.abc.net" and named entity recognition is disabled as well as pipe() method from spacy library of python to speed up the cleaning process. Then, the data is converted into series using series method of panda's library which can hold any type of data into a one-dimensional labeled array. After that, the pre-trained GloVe (The Global Vectors for Word Representation) embedding has been applied.

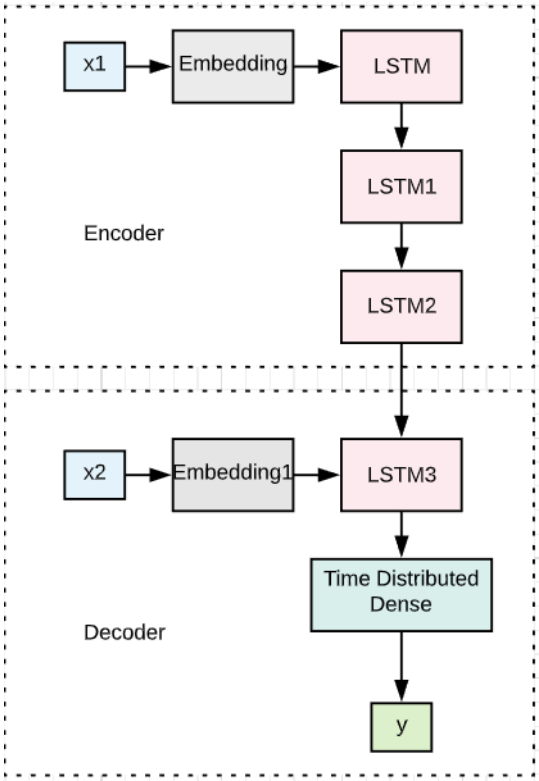

Fig. 1. Flow diagram of proposed model

In this paper, we tried to get the performance by using sequence to sequence model with LSTM on the news summary dataset. While building this model, we have used pre-trained GloVe embeddings in embeddings layer before passing it to the LSTM layers. A word embedding is a learned representation of text where words with the same meaning are expressed in a similar representation. In fact, word embedding is a class of techniques in which individual words in a predefined vector space are represented as real-valued vectors. Each word is mapped to a single vector, and the vector values are learned in a manner that resembles a neural network, and thus the technique is often lumped into the deep learning domain.

An embedding layer is a word embedding that is learned along with a neural network model on a task of processing the natural language, such as language modeling or document classification. It requires cleaning and preparing document text in such a way that each word is one-hot encoded. The vector space size is defined as part of the model, such as dimensions 50, 100, or 300. In this paper, vector space dimension used is 100. Small random numbers initialize the vectors. Using the back-propagation algorithm, the embedding layer is placed on the front end of a neural network and works in a supervised fashion.

GloVe, algorithm is an extension to the word2vec method developed by Pennington, et al. at Stanford for the efficient learning of word vectors. GloVe is an approach for combining the global statistics of matrix factorization techniques such as LSA in word2vec with local context-based learning. Rather than using a window to define a local meaning, GloVe uses statistics across the entire text corpus to construct an explicit word-context or word-co-occurrence matrix. The outcome is a learning model which can usually lead to better embedding of words.

TABLE I
PROPOSED MODEL SUMMARY

Layer	Output Shape	Params	Connected To
Input Layer	(None,200)	0	-
Embedding	(None,200,100)	7200100	input1[0][0]
LSTM	[(None,200,300),]	481200	embedding[0][0]
Input Layer	(None,None)	0	-
LSTM	[(None,200,300),]	721200	lstm[0][0]
Embedding	(None,None,100)	3144200	input2[0][0]
LSTM	[(None,200,300),]	721200	lstm1[0][0]
LSTM	[(None,None,300),]	481200	embedding1[0][0] lstm2[0][1] lstm2[0][2]
TimeDistributed	(None,None,31442)	9464042	lstm3[0][0]

To predict the summary of given text, we used a encoder-decoder sequence to sequence model using deep neural network of which flow diagram can be seen in Fig. 1. The first layer in our model is a pre-trained weight initialized word embedding layer which transforms words into a feature map in sentences and preserves the spatial (contextual) information for each word with the help of GloVe embeddings. It is accompanied by three LSTM as encoder layers. These encoder LSTM layers are applied with values of return state and return sequence as true as well as dropout and recurrent dropout as 0.5. Here, return sequence represents whether to return the last output in the output sequence, or the full sequence. Similarly, return state represents whether to return the last state in addition to the output. The dropout and recurrent dropout values are used to overcome an overfitting issue which generally overfits the model on training data and doesn't perform better while testing this model. After the encoder model, we passed the output of the encoder to the decoder

which again used the LSTM layer and the TimeDistributed layer which process the output from decoder's LSTM hidden layer. This TimeDistributed wrapped Dense layer mainly used to return the text sequences.

To summarize the model, This proposed sequence to sequence model comprised of an encoder which has 3 LSTM layers and a decoder which consists of an LSTM and a TimeDistributed wrapper Dense layer. Table.1 shows the generated model summary.

IV. Experimental Analysis

The training model was implemented on the News Summary dataset. The dataset gives information about various news headlines and its possible corresponding summary. The dataset is split into the ratio of 70:30 for training and testing with the help of sklearn library. It was observed that the text in the dataset contained non-alphabetic characters like escape characters, punctuations, multiple spaces and so on. Such characters are removed first. The preprocessing of the dataset is done in which tokenization is done and then text is converted to one-hot encoding with padding. While using the GloVe embeddings, we updated the weights feature matrix according to the pre-trained word embeddings for our dataset and passed this matrix to the embedding layer in the model.

The model was trained using LSTM layers. In the LSTM layer, different dropout and recurrent dropout values are compared to generate summary of news. Also, the values for maximum length of news text and summary of the text are used. We have used value of text length as 100 and summary length as 15 initially which did not provide accurate summary. Although, we were able to achieve better accuracy with maximum text length of 200 and maximum summary length of 50. The pre-trained word embeddings technique works in improving the performance raising the accuracy to 84.63% whereas maximum accuracy of around 55% was reached without using the Glove embedding technique. Fig.2 shows the plot for accuracy during the training and testing progress.

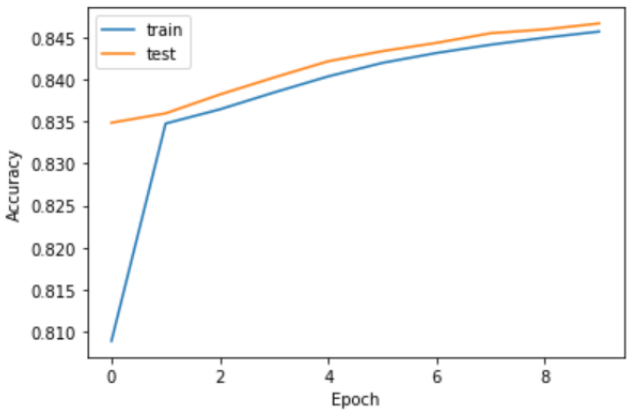

Fig. 2. Plot for accuracy during training and testing progress

A. Training parameters

There were several training parameters such as batch size, no of epochs, loss function and optimizer that were considered while building the prediction model. Batch size of 64 and 128 was considering during training the model where batch size of 128 trained the model in a more accurate way. Optimum number of epochs set while training the model were 10 by considering the accuracy and inference time. Different embedding dimension and tokenizer length were also considered while training the model. Table.2 shows the optimum value of the hyperparameters that suited the best for the model.

TABLE II
Optimum HyperParameter

Hyperparameter	Value
Optimizer	Adam
No of Epochs	10
Batch Size	128
No of Layers	9
Loss Function	Sparse Categorical Cross Entropy
Embedding Dimension	100
Tokenizer Length	5000

Loss function for a model is necessary to evaluate the performance and back propagate to adjust the parameters to generate better performance. Various optimizer were used in the loss function to optimize the model at every step. Optimizer such as SGD, AdamW and Adam were taken into consideration in the loss functionality. Table.3 shows the performance with respect to accuracy and loss by considering different optimizer.

TABLE III
Evaluation based on different Optimizer

Optimizer	Accuracy	Loss
Adam	84.63	1.02
AdamW	79.67	1.57
SGD	71.23	1.84

These results lead in selection of Adam as an optimizer while training the model. Loss function used in this model was sparse categorical cross entropy. Fig. 3 shows the plot for loss generated over 10 epochs for training and testing. The final generated output summary is represented in Table.4 for 4 different review inputs.

TABLE IV
Output of the proposed model

Original summary	Predicted summary
army jawan booked for raping minor in andhra	man arrested for raping minor girl in delhi
gaming firm highest paid exec with cr salary quits	us firm ceo quits after being hit by
sorry bhai pandya to after throw hits him	kohli takes dig at boundary in his match
us startup becomes after raising	startup raises 1 million in series funding

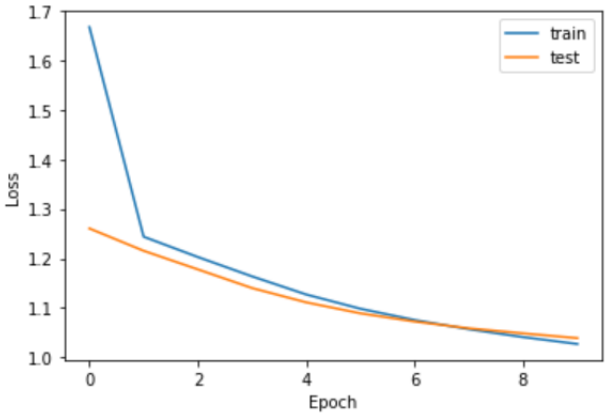

Fig. 3. Plot for loss during training and testing progress

V. Conclusion

In this paper, we have presented the model for summarization of text on news summary dataset. The model has performed with the accuracy around 84.63%. In this model, dropout and recurrent dropout used with LSTM layers played major role in resolving overfitting issue which provided comparatively better result. Experimental analysis shows that increasing the maximum text length as well as maximum summary length resulted to give more accuracy by 10%. This analysis aims on comparing the performance in training a deep learning model with new versus using pretrained word embeddings, where GloVe word embeddings has performed 40% better than new word embeddings which can be considered as a significant improvement in the sequence to sequence model on news summary dataset.

VI. Appendix

The dataset used for this project can be downloaded from this link:

https://www.kaggle.com/sunnysai12345/news-summary

The github link for the code can be found at this below link:

https://github.com/komalbarge45/NaturalLanguageProcessing/

The given model files on github are in JSON and h5 format. Json file is an acutal saved keras model which you can load and with the h5 file you can load the weights into the model. Without h5 file, you will not be able to run the loaded json model.

Below are list of tasks that were carried out for the execution of the project. Research on the text summarization techniques, studying python functions and lab codes, Selection of dataset, Preprocessing of dataset, Model enhancement using the basic encoder-decoder sequence to sequence LSTM model, Model evaluation with pre-trained embeddings, Trial and errors of hyperparameters, Code cleanup, Rigorous testing.

Contribution of group members:

- Srushti Wadekar - 25%
- Komal Barge - 25%
- Nidhi Patel - 25%
- Shivani Singh - 25%

References

[1] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos and Bing Xiang, “Abstractive text Summarization using Sequence-to-sequence RNNs and beyond”, SIGNLL Conference (CoNLL), 2016.

[2] Atif Khana, Naomie Salim, ”A review on abstractive summarization methods”, Journal of Theoretical and Applied Information Technology, January 2014.

[3] Linqing Liu, Yao Lu,Min Yang,Qiang Qu,Jia Zhu,Hongyan Li, ”Generative Adversarial Network for Abstractive Text Summarization”, The Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[4] Lei 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, August 2019.

[5] Joel Larocca Neto, Alex A. Freitas and Celso A. A. Kaestner, “Automatic Text Summarization using a Machine Learning Approach”, Brazilian Symposium on Artificial Intelligence, January 2003.

[6] Som Gupta and S. K Gupta, ”Abstractive summarization: An overview of the state of the art”, Expert Systems with Applications, Volume 121, May 2019.

[7] Shengli Song, Haitao Huang, and Tongxiao Ruan, ”Abstractive text summarization using LSTM-CNN based deep learning”, Multimed Tools Appl 78, 857–875 (2019).

[8] Jeffrey Pennington, Richard Socher, and Christopher Manning, ”Glove: Global vectors for word representation”, Conference on Empirical Methods in Natural Language Processing, October 2014.