Unique Combinations of LSTM for Text Summarization

Anvitha Aravinda
Department of CSE
B.M.S. College of Engineering
Bangalore, India

Gururaja H S
Department of ISE
B.M.S. College of Engineering
Bangalore, India

Padmanabha J
Department of ISE
Bangalore Institute of Technology
Bangalore, India

***Abstract*— Manual conversion of a huge text file into summarized one takes up immense amount of time. Text summarization is the process of shortening the data, such that only relevant and important data is represented from the original data. There are two types of text summarizations, Abstractive text summarization and Extractive text summarization. This paper concentrates on different ways of implementing the Abstractive text summarization. A comparative study of the different implementations is carried out on state-of-art benchmark dataset. The implementations carried out are seq2seq model with LSTM, seq2seq model with bidirectional LSTM and hybrid seq2seq model. This paper tells which among these implementations are the best for abstractive text summarization and why.**

Keywords*— *Abstractive text summarization, Bidirectional LSTM, LSTM, Seq2Seq Models

I. INTRODUCTION

On the worldwide web data keeps growing and there is always plenty of information available. To have to read all the information is a short period of time is difficult. This is when the text summarization comes in handy. Nowadays text summarizers are used in summarizing newspaper, articles and even for headline generations, weather forecast, stock market etc. Due to its wide range of applications text summarization has become a hot topic of research for top-notch universities.

Web provides plenty of data. But dealing with such enormous amount of data and trying to take all the data as input and provide an output summarizing all is a tedious process. With the invention of text summarizers, it's just a matter of few seconds to convert huge amount data to short version maintaining the meaning of the original data. Prior to this time entire summarization was done manually taking up lot of time and labour. With automation this has been greatly reduced.

Abstractive text summarising and extractive text summarization are the two main types of text summarization [11]. Extractive text summarization makes use of already present sentences in the corpus which are of importance and arrange them into a summary. Abstractive text summarization on the other hand uses natural language to summarize the text i.e., the text present in this summary will be different from what is present in the corpus. This paper mainly concentrates on the Abstractive text summarization. The major focus of this study is on recent progress in sequence-to-sequence models for abstractive text summarization.

II. RELATED WORK

Abstractive text summarization has achieved a lot importance lately and many researched have been conducted in this field. A paper [1] threw light upon deep learning model, Seq2Seq for providing excellent abstractive summary. The drawbacks of using the deep learning model [9]were that the model produced repetitive word and even the sentences had very limited vocabulary. The idea in paper [1] explored a way to get over these drawbacks. Firstly, transform the script into a single document using preprocessing techniques, followed by usage of word vectorization implemented along with pretrained word embeddings like Glove 6B to vectorize words into vector which is later used in the proposed model. Lastly using Bidirectional LSTM to encode the text and Unidirectional LSTM to decode. This paper implements this model proposed and compares the model with usage of only LSTM and improvement of using both decoder and encoder as bidirectional LSTM.

The sequence-to-sequence models are classified into CTC, RNNs and Attention based model, which are common models of connectionist temporal classification (CTC) [2]. The definition [2] stated CTC algorithm as "This algorithm is a method of preparing start to finish models without a requirement of casing level arrangement of the objective names for a preparation articulation" [2]. CTC's use with respect to Sequence-to-Sequence models is that it deals with problems based on length of sequences. Attention layer is used along with sequence to sequence model as it applies more weight on some layers producing the output. The paper [2] explored and stated the difference between RNN and CTC as "The RNN transducer differs on the encoder usage from the CTC alignment model by different repeat lease expectation arrangement over the output sequences". This paper mainly concentrates on RNN based approach using a special type of RNN called LSTM (long short-term memory) which will be elaborated further on in the paper.

III. LONG SHORT TERM MEMORY

Long short term memory (LSTM) are blocks which build up the layers of recurring neural network RNN. A cell, an input state, an output state, and a forget state make up each LSTM block.The cell is the key part of LSTM which helps memorizing the values over a certain interval of time. This memorizing feature of LSTM gives its name as Long-short term memory. The three gates act as flow controllers which regulate the flow of values in the LSTM. In a neural network [7] these gates act as neurons which have the capability of computing the activation of a weighted sum.[3].This is model where the short-term memory lasts for a longer time duration. Fig 1 provides the general depiction of RNN and LSTM cell.

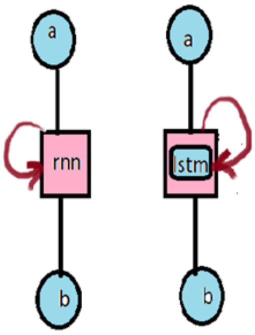

Fig. 1.RNN cell and LSTM cell general depiction; LSTM is very similar to RNN with an additional memory element.

In the Fig 2, LSTM is a single layered LSTM for better understanding with all layers having sigmoid activation function except candidate layer, which has Tanh activation function.

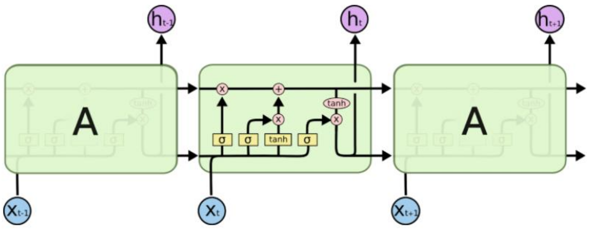

Fig. 2. LSTM Cell, taken from analytical Vidhya [6]

The three major inputs of an LSTM are the current input, the prior memory state, and the previous hidden state. The output produced by LSTM is current memory state and current hidden state. The Candidate gate, Forget gate, Input gate, and Output Gate each accept an input vector and a hidden vector as inputs and combine them to generate a series of output vectors, one from each of the gates named. The memory cell state stores the input and it can be modified by using the forget gate. If the value of the forget gate is 0 the memory forgets its previous state else it continues to remember. This is the simple logic used behind Long Short-Term Memory.

*Ct = Ct-1*ft*

Here Ct is the current state, Ct-1 is previous state and ft is the forget gate state. Each element is multiplied with forget gate state to determine if the memory is retained or not.

The output of this LSTM is the combination of Ct and Ht (hidden state got by taking tanh of Ct) which is passed on to the next time step and the same process continues again.

Component	Description
Forget Gate	Sigmoid Neural Network
Memory State	Vector
Hidden State	Vector
Output State	Sigmoid Neural Network
Input State	Sigmoid Neural Network
Candidate Layer	Tanh Neural Network

Fig. 3. A total of six set of components makes up the LSTM.

$$
\begin{aligned}
f_t &= \sigma\,(X_t * U_f + H_{t-1} * W_f)\\
\bar{C}_t &= \tanh\,(X_t * U_c + H_{t-1} * W_c)\\
I_t &= \sigma\,(X_t * U_i + H_{t-1} * W_i)\\
O_t &= \sigma\,(X_t * U_o + H_{t-1} * W_o)
\end{aligned}
$$

$$
\begin{aligned}
C_t &= f_t * C_{t-1} + I_t * \bar{C}_t\\
H_t &= O_t * \tanh\,(C_t)
\end{aligned}
$$

Fig. 4. * Represents multiplication by element, + represents addition by element, W, U stand for weight vectors, C stands for cell memory, H stands for hidden memory, X stands for input vector, The letters I and F stand for input gate and forget gate, respectively..

IV. BIDIRECTIONAL LSTM

Bidirectional LSTM [4] is a special kind of LSTM which allows data flow in both the directions. In regular LSTM data flows in one direction, i.e., It can be in forward direction or it can be in backward direction. But the use of Bidirectional LSTM is that input data flows in both forward and backward direction. The major usage of this is to allow the model to remember both past and future information. In bidirectional LSTM the past information is preserved carefully.

The example for bidirectional LSTM [6] can be if a sentence says “friends go in…..” and the sentence provided is “friends came in car”. This can be used to predict the past part of the sentence that is “friends go in car”. For this kind of deriving the past from future or future information from past information Bidirectional LSTM can be used.

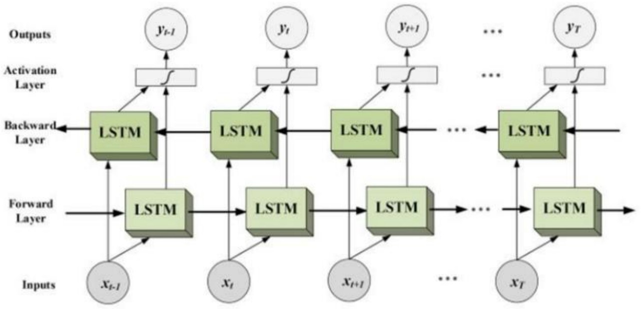

Fig. 5. Simple understanding of bi-LSTM with forward and backward layers.

V. SEQUENCE-TO-SEQUENCE (SEQ2SEQ)

In order to understand the implementation of various different models for text summarization [11] it is important to know what is sequence to sequence modelling. This paper provides a brief explanation of it.

Seq2Seq is a process of training a model to convert a sequence from one realm to another realm. For example, to convert a sentence in German language to English. Machine translation is not the only application of seq2seq model [10]. It has various applications in the field of question answering AI models as well. This is mostly used to generate new texts. In this paper main focus is on RNN based seq2seq model.

In most of the cases in seq2seq entire input sequence is required to predict the target or the output sequence. This is especially when the length of input sequence and output sequence is different. This basically consists of two layers, i.e., encoder and decoder layer. Encoder layer takes the input and creates a state which acts as a context or a condition to the decoder layer. The decoder layer is used to predict the upcoming characters given the previous characters of the target sequence. Decoder [10] predicts targ(t+1…….) if targ(…….t) is given, which is the context of the input sequence.

A different process is followed for decoding different input sequences, i.e., sequences unknown. Firstly, input is sent through encoder and state vectors are obtained, followed by injecting the state vectors and a target char of size 1 into the decoder, the new character obtained is sampled and added to target and this continues until and end is indicated. In known sequences decoder gets the information from state vectors of encoder itself.

VI. IMPLEMENTATION

Seq2Seq model with just LSTM is the first model the paper will elaborate on. This is a simple model with both encoder and decoder having just LSTMs. Second is a Seq2Seq model with encoder and decoder having bidirectional LSTM. Third and the last one is hybrid where encoder is having bidirectional LSTM and decoder has just LSTM.

A. *Dataset*

An online published NEWS SUMMARY dataset is used, which is most commonly used dataset for text summarizations. Usually, a news summary related dataset [5] is used for such tasks as they enrich in vocabulary and help in providing the model new and more insight.

The collection comprises 4515 instances [5], each of which includes the author's name, headlines, article URL, brief text, and the whole article. It's made up of Inshorts news summaries and scraped news articles from the Hindu, the Times, and the Guardian. This is an open-source text summarising dataset [3].

B. *Text Preprocessing*

Text preprocessing is the process of making the data ready to be modelled and fed into the model for it to learn from. The cleaned dataset consists of the headlines and short text of the original dataset. All the contractions are expanded, punctuations are removed, stopwords are removed, numbers are removed, hyphen and all email id are removed and replaced with empty strings with the help of regular expressions. Once the text is cleaned _START_ and _END_ is inserted to each text to mark the starting and ending of the texts. So, decoder keeps decoding until it finds end keyword. The max length of summary and max length of headlines is determined to pad the sequences after tokenizing.

Tokenization is done to break the raw text into words, sentences called tokens. This helps in understanding the meaning of text based on the sequence of words got from tokenization. One hot encoding is the process of converting words into sequence of numbers. Usually, the numbers are 1s and 0s sequences. Each word represented with unique sequence of numbers in the form of vectors. Thus, each sentence will be an array of vectors. Moreover, if there are many sentences it will be an array of array of vectors or an array of matrices. Therefore, a three-dimensional tensor has to be fed into the neural network for it to learn the sequences. A number sequence is easily understood and matched by the neural network than textual sequence.

C. *Modelling*

The three models to be compared are seq2seq model with only LSTM, with bidirectional LSTM and a combination of both LSTM and bidirectional LSTM. Word embeddings are text representations of words with a real-valued vector that encodes the meaning of the word and allows it to be extended into words with comparable meanings. In this implementation GloVe word embeddings are used to produce new words of similar meaning to have a well sentenced, partially or completely grammatical summary. GloVe is termed as global vectors for word representation. The definition in [8] stated GloVe as a Stanford-developed unsupervised learning technique for constructing word embeddings from a corpus of words by aggregating word-word co-occurence matrices.Model_1 consists of encoder and decoder having only LSTM,Model_2 consists of encoder and decoder having bidirectional LSTM, Model_3 consists of encoder having bidirectional LSTM whereas decoder having only LSTM.

D. *Results*

Model_1 has a higher overfitting as the gap between the training curve and the validation curve suggests in the accuracy vs epoch.Model_1 has a higher learning rate as the curve shown in the loss vs epoch suggests that higher learning rate requires lesser number of samples or epochs for the model to learn but at the same time it makes the model come to a suboptimal result rapidly. Therefore, the words in the summary have the tendency to get repeated.

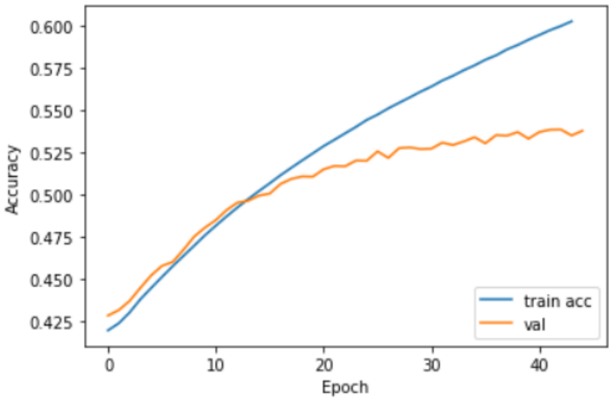

Fig. 6. Accuracy against Epoch for Model_1

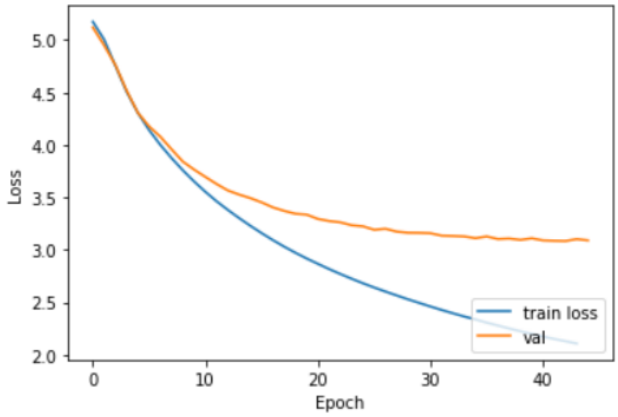

Fig. 7. Loss against Epoch for Model_1

Model_2 as suggested by the accuracy vs epoch graph has a very little overfitting as the gap between the training curve and the validation curve is very less. This indicates that the error of testing and training is less. Model_2 has a good learning rate as suggested by the loss vs epoch graph meaning the weighted are learnt by the model at a good rate and at the right pace to yield good results.

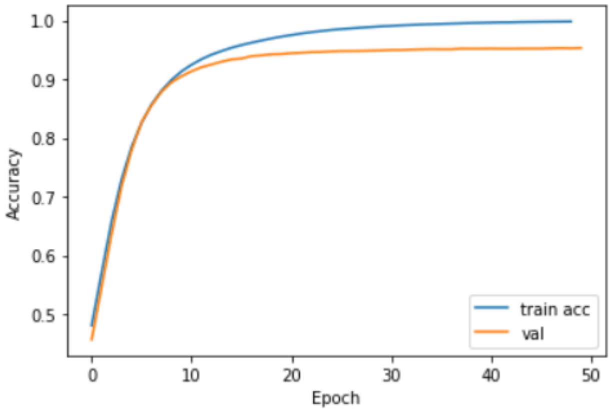

Fig. 8. Accuracy against Epoch for Model_2

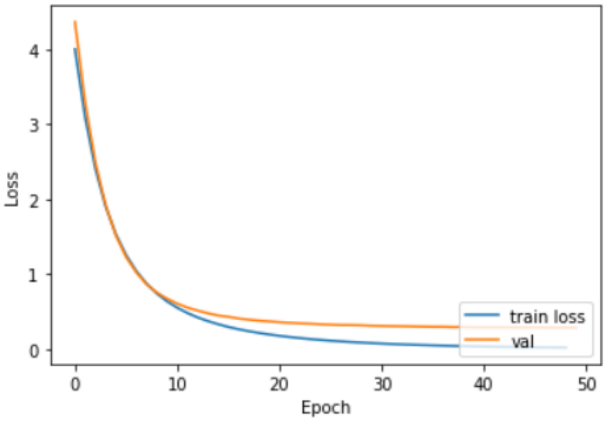

Fig. 9. Loss against Epoch for Model_2

Model_3 as suggested by the accuracy vs epoch graph has a larger gap indicating significant overfitting having uneven testing errors in the model as in model_3 the output decoder is just LSTM and not bidirectional LSTM. Again model_3 has a higher learning rate for validations as the output decoder is just LSTM.

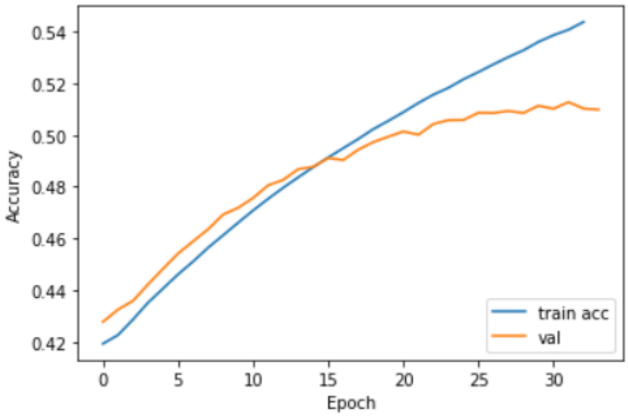

Fig. 10. Accuracy against Epoch for Model_3

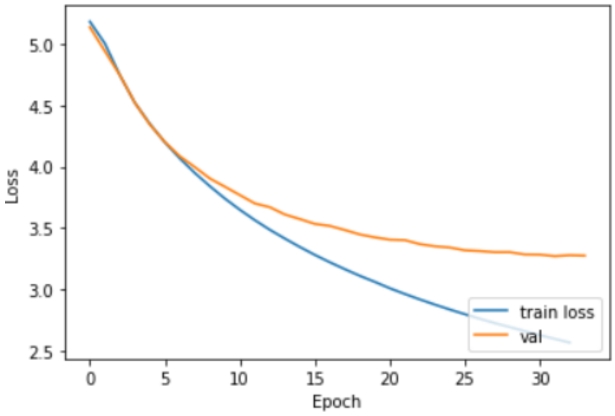

Fig. 11. Loss against Epoch for Model_3

VII. CONCLUSION

The results suggest that Bidirectional LSTM is preferred for text summarization tasks as this helps in remembering the future possibilities and to store the past information. Using both the input flows of forward and backward the summary can be easily predicted. This paper explains to an extent how bidirectional LSTM[12] can be used to find the text summary for difficult task like providing the gist of newspaper articles within milliseconds with a good accuracy as well.

REFERENCES

[1] Sirohi, Neeraj; Rajshree, Mamta; and Rajan, Siddhi,. "Text Summarization Approaches Using Machine Learning & LSTM",.Revista Gestão Inovação e Tecnologias, 2021.

[2] Yousuf, Hana; Gaid, Michael; Salloum, Said; and Shaalan, Khaled, "A Systematic Review on Sequence to Sequence Neural Network and its Models", International Journal of Electrical and Computer Engineering, 2020.

[3] [Online].Available:https://en.wikipedia.org/wiki/Long_short-term_memory

[4] 2021. [Online]. Available: https://analyticsindiamag.com/complete-guide-to-bidirectional-lstm-with-python-codes/

[5] [Online]. Available: https://www.kaggle.com/sunnysai12345/news-summary

[6] Basaldella, Marco; Antolli, Elisa; Serra, Giuseppe; and Tasso, Carlo written,"Bidirectional LSTM Recurrent Neural Network for Keyphrase Extraction", 2018.

[7] 2019. [Online]. Available: https://towardsdatascience.com/useful-plots-to-diagnose-your-neural-network-521907fa2f45

[8] 2018.[Online]. Available: https://medium.com/analytics-vidhya/word-vectorization-using-glove-76919685ee0b

[9] 2019.[Online].Available: https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/

[10] 2017. [Online]. A ten-minute introduction to sequence-to-sequence-learning in Keras is available at https://blog.keras.io/a-ten-minute-introduction-to-sequence-to-sequence-learning-in-keras.html.

[11] Rajendran, G.B., Kumarasamy, U.M., Zarro, C., Divakarachari, P.B. and Ullo, S.L., 2020. Land-use and land-cover classification using a human group-based particle swarm optimization algorithm with an LSTM Classifier on hybrid pre-processing remote-sensing images. *Remote Sensing*, *12*(24), p.4135.

[12] TK. Shashank, N. Hitesh, HS. Gururaja, Application of Few-Shot Object Detection in Robotic Perception, Global Transitions Proceedings,2022,ISSN-2666-285X, https://doi.org/10.1016/j.gltp.2022.04.024.