APPLICATION OF NEXT WORD PREDICTION

S. HARSHA VARDHAN, SAI AKASH ADDALA, ANIRUDH NAGALUTI

S. CHANDRASEKHAR (Guide)

Sreenidhi Institute of Science and Technology

Hyderabad

ABSTRACT

Next Word Prediction is additionally called Language Modeling. It is the undertaking of predicting what word comes straightaway. Attempting to make model utilizing nietzsche default text record which will foresee clients sentence after the clients composed 40 letters, the model will comprehend 40 letters and anticipate impending top 10 words

utilizing RNN neural organization which will be executed utilizing Tensorflow. Our Aim of creating this model to predict 10 or more then 10 words as fast as possible utilizing minimum time. As RNN is Long Short Time memory it will understand past text and predict the words which may be helpful for the user to frame sentences and this technique uses letter to letter prediction means it predicts letter after letter to create a word.

INTRODUCTION

Natural Language Processing (NLP) is a significant part of artificial Intelligence, which incorporates AI, which contributes to finding productive approaches to speak with people and gain from the associations with them. One such commitment is to give portable clients anticipated ”next words,” as they type along within applications, with an end goal to assist message conveyance by having the client select a proposed word as opposed to composing it.

LSTM

When back propagation is taken into account, neural networks may suffer from disappearing gradient descent. The weight update mechanism was significantly impacted by the Brobdingnag Ian effect, and the model was rendered worthless. A memory cell containing a concealed state

and three gates—a unit forgotten gate, a scan gate, and an input gate—make up an LSTM.

Since there may be lags of uncertain length between significant occurrences in a time series, LSTM networks are well-suited to categorising, processing, and making predictions based on time series data. To solve the vanishing gradient issue that can arise when training conventional RNNs, LSTMs were created. LSTM has an advantage over RNNs, hidden Markov models, and other sequence learning techniques in that it is rather insensitive to gap length.

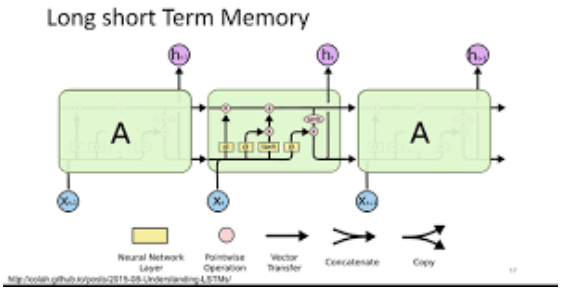

Fig1 : LSTM

RNN

It is a kind of neural network in which the results of one stage are used as inputs for the next. Because it performs the same procedures for each knowledge entry, the RNN is repetitive in nature, but at the same time, the outcome is dependent on the prior calculation. The decision is based on a degree-level examination of both the result from the prior input and this input. RNNs can method input sequences in a way that direct communication neural networks cannot because of their internal state (memory). The inputs to RNNs are all connected.

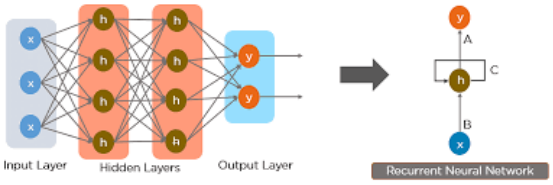

Fig2 : RNN

Bi-LSTM

Bidirectional LSTM is the process of making any neural network having the sequence of both directions backward or forward. Bi-LSTMs effectively increase the amount of information available to the network, improving the context available to the algorithm.

Bidirectional long-term dependencies between time steps of time series or sequence data are learned via a bidirectional LSTM layer. When you want the network to learn from the entire time series at each time step, these dependencies can be helpful.

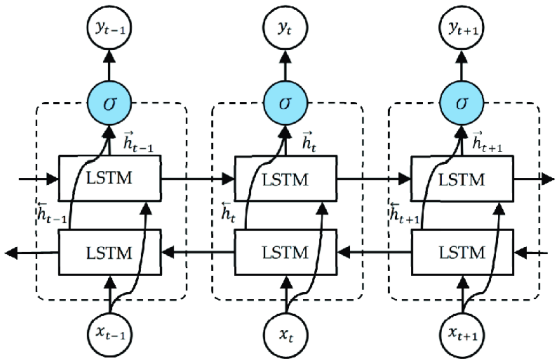

Fig3 : BI-LSTM

DATA PREPROCESSING

The first step is to purge the Metamorphosis dataset of all extraneous information. The dataset's beginning and finish will be removed. Save the file after completing this step. We'll use utf-8 encoding to retrieve the Metamorphosis. We then proceed to replace all extra new lines that are superfluous, the carriage return, and the Unicode character. Finally, we will make sure that all of our words are original. Each word will only be taken into account once, and all further repeats will be dropped. The model train will run more smoothly as a result of the repetition of terms causing less confusion.

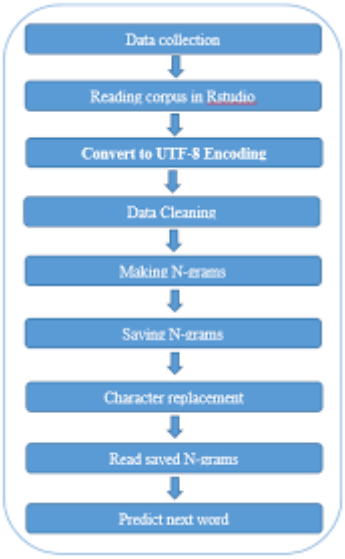

Fig4 : Data Preprocessing

TOKENIZATION

Tokenization is one of the essential social control techniques. It merely divides the continuously flowing text into separate word parts. One really simple method would be to divide inputs by home and give each word its own identity.

By translating each text into either a sequence of integers or a vector with a coefficient for each token that might be binary, based on word count, based on tf-idf, the Keras Tokenizer enables us to vectorize a text corpus.

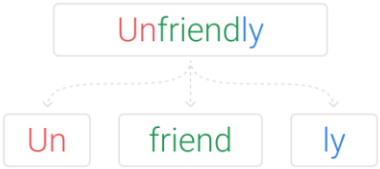

Fig5 : Tokenization

N_GRAM

A n1-order Markov model is used in an n-gram model, a sort of probabilistic language model, to predict the next item in a sequence. Current applications of n-gram models include probability, communication theory, computer linguistics, computational biology, and data compression. The simplicity and scalability of n-gram models are two advantages. With greater n, a model may hold more context with an efficient trade-off between space and time.

PAD SEQUENCES

It remains challenging to give our neural networks inputs of comparable length when converting texts to numerical values. Not all sentences are the same length. The pad sequences function is used to truncate some of the longer sequences and replace some of the shorter sentences with zeroes.

Furthermore, since it is frequently indicated whether or not to pad and truncate from either the start or the termination, relying on the pre settings and post settings for the padding arguments and the truncating arguments By default, truncation and artefact arguments may occur at the beginning of the sequence.

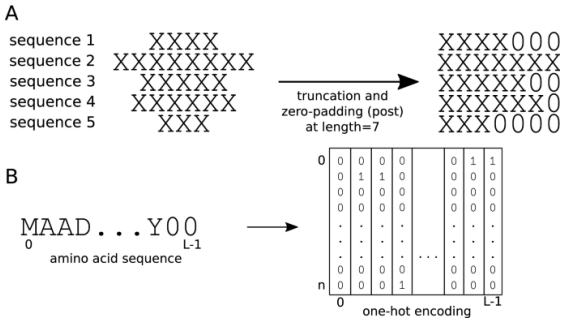

Fig6: Pad Sequences

CALLBACKS

The three necessary callbacks will be imported so that we may train our model. ModelCheckpoint, ReduceLROnPlateau, and Tensorboard are the 3 crucial callbacks. They are:

Modelcheckpoint — After training, the weights of our model are stored using this callback. By setting the save best only True option, we only save the best weights from our model. We'll use the loss metric to track our training progress.

After a predetermined number of epochs, the optimizer's learning rate is reduced using the callback ReduceLROnPlateau. Here, the patience has been set to 3. After three epochs, if the accuracy does not increase, our learning rate is lowered by a factor of 0.2. Loss is another measure that is being watched in this situation.

Tensorboard — The tensorboard callback is used to draw the graph plots for accuracy and loss when visualising the graphs. Only the loss will be considered in this instance.

PREDICTION

We will load the tokenizer file that we have saved in pickle format for the prediction notebook. The following word model, which has been saved in our directory, will subsequently be loaded. Each of the input sentences for which predictions must be made will be tokenized using the same tokenizer. Following this, we may use the model that was previously saved to generate predictions about the input phrase.

```
Enter your line: at the dull
weather
Enter your line: collection of textile
samples
Enter your line: what a strenuous
career
Enter your line: stop the script
Ending The Program.....
```

RESULTS

Now that it has been finished, the next word prediction model does reasonably well on the dataset.

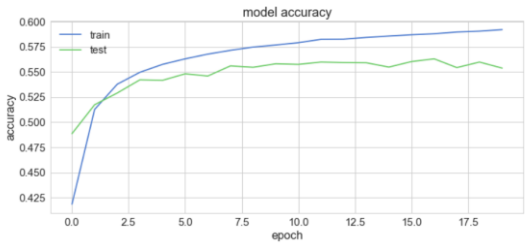

Fig7 : Model Accuracy

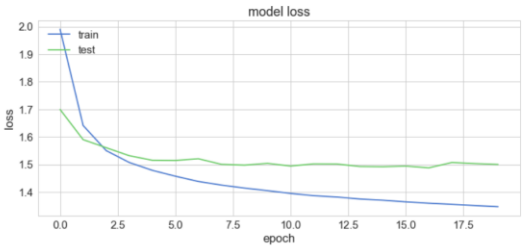

Fig8 : Model Loss

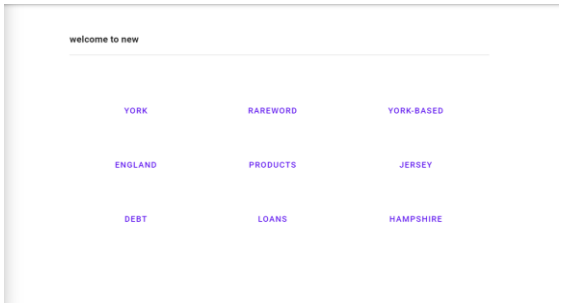

Fig9 : Results

MERITS

Word prediction is based on spelling, syntax, the frequency principle, and the most recent usage of a word. According to a number of factors, words are predicted. The word must first be included in the dictionary being used. A word will appear higher on the list if it has recently been used. If it is commonly used, it will be near the top of the list. Word prediction should improve in accuracy over time as the student uses it with more words, depending on the programme being utilised.

LIMITATIONS

The Markov chain lacks memory. The use of this strategy has numerous drawbacks. As an illustration, the word "sandwiches" after the phrase "I ate so many grilled..." will be predicted based on how frequently the phrase "grilled sandwiches" has appeared together in the training data. There are numerous circumstances when this strategy could fail because we are simply receiving suggestions based on frequency.

SOFTWARE AND HARDWARE REQUIREMENTS

■RAM : 4GB

■Storage : 500GB

■CPU : 2GHz or faster

■Architecture : 32bit or 64bit

■Python 3.5 in google Colab is used for data pre-processing, model training and prediction.

■Operating System : Windows 7 and above or Linux based OS or Mac OS.

CONCLUSION

For the metamorphosis dataset, we can create a high-quality next word prediction. On the available dataset, the next word prediction model we have created is fairly accurate. The prediction's overall quality is good. To improve the model's prediction, specific pre-processing procedures and model modifications might be done.

Increased writing fluency might help kids produce more writing by using word prediction. offer audio confirmation of word choice. reduce the discrepancy between potential and accomplishment as shown via textual expression.

ACKNOWLEDGMENTS

We would like to express our special thanks to our mentor S. Chandrasekhar who provided us with a fantastic opportunity to do this beautiful project on this subject, which also enabled us to conduct extensive study and learn a great deal of fresh information. We are very appreciative of him.

Second, we also want to express our gratitude to my friends, who greatly contributed to the timely completion of this job.

REFERENCES

- R. Sharma, N. Goel, N. Aggarwal, P. Kaur and C. Prakash, "Next Word Prediction in Hindi Using Deep Learning Techniques", *2019 International Conference on Data Science and Engineering (ICDSE)*, pp. 55-60, Sep. 2019.
- F. Rakib, S. Akter, M. A. Khan, A. K. Das and K. M. Habibullah, "Bangla Word Prediction and Sentence Completion Using GRU: An Extended Version of RNN on N-gram Language Model", *2019 International Conference on Sustainable Technologies for Industry 4.0 (STI)*, pp. 1-6, Dec. 2019.
- S. M. Qaisar, "Sentiment Analysis of IMDb Movie Reviews Using

Long Short-Term Memory", *2020 2nd International Conference on Computer and Information Sciences (ICCIS)*, pp. 1-4, Oct. 2020.

- [online] Available: https://www.kaggle.com/dorianlazar/medium-articles-dataset?select=medium_data.csv.