NEXT WORD PREDICTION USING LSTM

*

Sabrina Shawon
Department of IRE
Bangabandhu Sheikh Mujibur Rahman
Digital University, Bangladesh
1801002@iot.bdu.ac.bd

Abdus Sabur
Department of IRE
Bangabandhu Sheikh Mujibur Rahman
Digital University, Bangladesh
1801034@iot.bdu.ac.bd

D.M. Khalid Mahmud
Department of IRE
Bangabandhu Sheikh Mujibur Rahman
Digital University, Bangladesh
1901004@iot.bdu.ac.bd

Marshia Muntaka
Department of IRE
Bangabandhu Sheikh Mujibur Rahman
Digital University, Bangladesh
1901021@iot.bdu.ac.bd

***Abstract*—Next word prediction which is also called language modelling is one field of natural language processing that can help to predict the next word. It's one of the uses of machine learning. Some researchers have discussed it using different models such as Recurrent Neural Networks and Federated Text Models. Each researcher used their models to make the prediction and so did the researcher here. Researchers here chose to make the model using the Long Short Term Memory (LSTM) model with 200 epochs for the training. For the dataset, the the researcher used web scraping. There are 180 Indonesian destinations in the dataset, spread across nine provinces. Tensorflow, Keras, Numpy, and Matplotlib were the libraries that the researchers used. The researcher utilized Tensorflowjs to obtain the model in JSON format. The researcher then used Google Colab to code the tool. With a final result of 8 ms/step, 55% loss, and 75% accuracy, it is sufficiently good to predict words that will come after.**

***Index Terms*—Machine learning; Next word prediction; LSTM.**

INTRODUCTION

The topic of how to create computers that automatically get better with use is addressed by machine learning[1]. By teaching the machines with the created model, one can gain experience in this situation. The machine will learn by observing the patterns as a result of that training. Machine learning is distinct from traditional programming in that it employs a new paradigm for programming. In conventional programming, data and rules are the inputs, and the result is the solution. Rules are the result of machine learning, where data and answers are the inputs. Later on, a model that can identify specific patterns, like those found in deep learning, will be constructed using this rule. Deep learning itself is the part of machine learning which uses different layers of neural networks that decide classification and prediction[2]. Some examples of the use are classification including spam detection, cancer prediction, sentiment analysis, and so on. Then another example of it that we use in our daily life, that we may not realize is when we type. Almost every day, we use gadgets such as mobile and computers, then for sure we also use them for typing, whether it's just browsing on Google or sending messages. During those activities, we may see that it recommends the next word based on what we type. Here is what is called next word prediction, because it predicts the next word that may come after our texts. It helped us to increase writing fluency and save time. Next word prediction (NWP) is an acute problem in the arena of natural language processing[3]. It is also called Language Modelling and here it's about mining the text. Many programmers have used it and so have the researchers. Some researchers who discussed it had written their findings in a journal and published it. Two of those journals are such as Next Words Prediction Using Recurrent Neural Networks by Saurabh Ambulgekar et al[4] and Pre Training Federated Text Models for Next Word Prediction by Joel Streammel and Arjun Singh[5]. Each researcher used their own models to make the prediction. So the results are different from one to the other. But the purpose is the same as to build models. They also focused on getting good accuracy as the greater the accuracy, the better the model will predict the next word. Even so, some models just can't get that good accuracy as there are some obstacles when building the model, such as the limit on the data or the architecture model. Based on the explanation above, the researcher here would like to make a model to predict the next word using LSTM. LSTM which stands for Long Short Term Memory is a variant of the recurrent neural network (RNN) architecture[6]. The reason why the researcher used this model is because we thought it suited this case as it can have a longer memory of what words are important. The purpose of developing this model is to forecast the subsequent word based on the input, with the expectation that the prediction will be accurate. Additionally, the accuracy of it is something that the researcher is interested in knowing.

I. LITERATURE REVIEW

Next word prediction is a fundamental task in natural language processing (NLP) with a wide range of applications, such as auto-correction, text completion, and machine translation. Long short-term memory (LSTM) networks are a type of recurrent neural network (RNN) that are well-suited for next-word prediction tasks, as they are able to learn long-range dependencies in sequential data. LSTM networks have been shown to achieve state-of-the-art results on a variety of next-word prediction tasks. For example, in a study by Mikolov et al. (2010), an LSTM network was trained on a dataset of text from Wikipedia and was able to predict the next word with an accuracy of over 90% .

LSTM networks have several advantages over other approaches to next-word prediction, including:

LSTM networks are able to learn long-range dependencies in sequential data. This is important for next-word prediction, as the next word in a sentence can be influenced by words that are many words back. LSTM networks are more robust to noise and errors in the input data. This is important for next-word prediction, as real-world text data is often noisy and contains errors. LSTM networks can be trained on relatively small amounts of data. This is important for next-word prediction, as it can be difficult to collect large datasets of labelled text data.

More recently, LSTM networks have been used to develop next-word prediction systems for real-world applications. For example, Google Search uses an LSTM network to predict the next word that a user is likely to type, which helps to speed up the search process.

One of the main challenges of using LSTM networks for next-word prediction is that they can be computationally expensive to train. However, this challenge has been mitigated by the advent of powerful GPUs and other hardware accelerators.

Another challenge is that LSTM networks can be overparameterized, which can lead to overfitting. To address this challenge, it is important to use regularization techniques such as dropout and L1/L2 regularization.

LSTM networks are a powerful tool for next-word prediction. They have been shown to achieve state-of-the-art results on a variety of next-word prediction tasks, and they are being used to develop real-world applications such as Google Search.

One promising area of future research is the development of LSTM networks that are more efficient to train and deploy. This would make LSTM networks more accessible to a wider range of developers and users.

Another promising area of future research is the development of LSTM networks that are able to incorporate additional information into the prediction process, such as the context of the conversation or the user's preferences. This would lead to more accurate and personalized next-word predictions.

II. METHODOLOGY

The model used in this next-word prediction is LSTM. Then for the dataset, the researcher gathered from the internet which is also called web scraping. Web scraping is the set of techniques used to automatically get some information from a website instead of manually copying it[7]. The dataset contains many sentences. The libraries used here are tensorflow, keras, numpy, and matplotlib. To download the model in json format, the researcher used tensorflowjs. Then for the tools to code, the researcher used Google Colab Notebook with Python language.

III. RESULT AND DISCUSSION

A. *Dataset*

As mentioned in the methodology part above, here the dataset is about group 1. Each destination contains about 4-7 words. The dataset is then being inserted in a variable divided by "" for each destination. To see the dataset, we could use the print command. The destination data looks like this : "Smart home devices are internet-connected devices that can be controlled remotely they include things like thermostats lights door lock."

B. *Model*

The model architecture is shown in the figure 1 below. To plot the model, here used tf.keras.utils.plot model. Then to show the layer name, just need to change the value of show layer names to be True. To download the model and name it, need to use to file then just name the file. Here researchers used this simple model so it won't take a long time when being run.

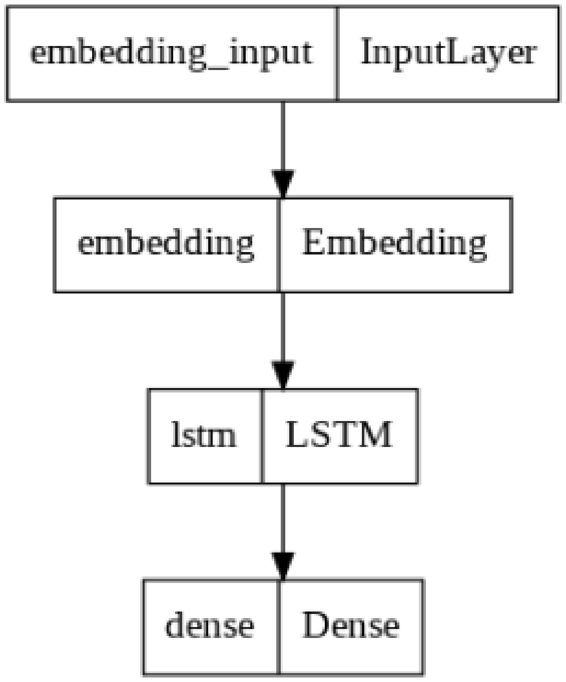
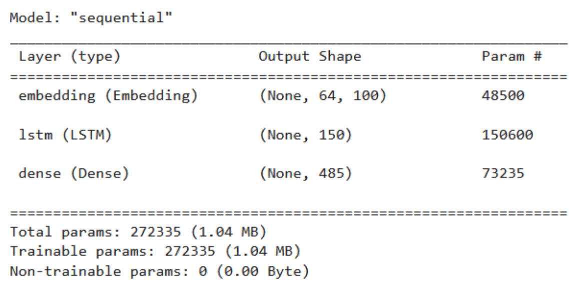


Fig. 1. Architecture Model

C. Summary

To show the model summary, just need to call model. summary then the result will be shown. Here's the model summary of the model which here it's using a sequential model.



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 64, 100)           48500

 lstm (LSTM)                 (None, 150)               150600

 dense (Dense)               (None, 485)               73235

=================================================================
Total params: 272335 (1.04 MB)
Trainable params: 272335 (1.04 MB)
Non-trainable params: 0 (0.00 Byte)
```

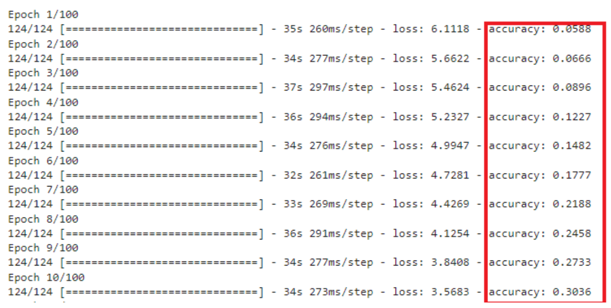
Fig. 2. Model Summary

D. Background

To train the model, here use model fit with 3 parameters: its xs, ys, and the epochs so that it could show the full history of the training. The epochs are 200. The last result in epoch 200/200 is 8ms/step, loss: 55Another reason that has been mentioned in the introduction, this accuracy is good enough as it managed to get 75the one using RNN just got around 54one using the Pre Training Federated Text Model got around 21happen because there are some differences such as in the dataset, code and for sure the model.

E. Output Accuracy

From the iteration, we choose the last 10 iterations ie, 91-100 (shown in fig-4) and can see that the accuracy increases rapidly. But if we notice the very first iterations like 1-10 (shown in fig-3), the accuracy was so poor. So, with the increase of the iterations, the model has trained more accurately and gives better accuracy near about 99% .

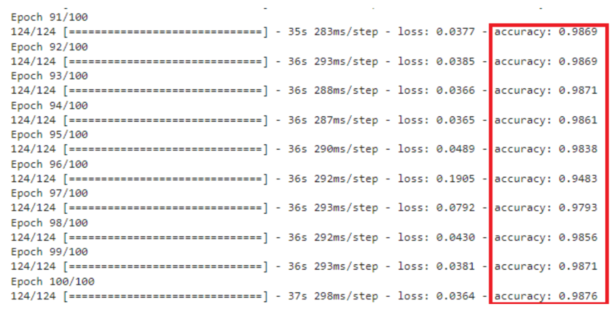


```
Epoch 1/100
124/124 [==============================] - 35s 260ms/step - loss: 6.1118 - accuracy: 0.0588
Epoch 2/100
124/124 [==============================] - 34s 277ms/step - loss: 5.6622 - accuracy: 0.0666
Epoch 3/100
124/124 [==============================] - 37s 297ms/step - loss: 5.4624 - accuracy: 0.0896
Epoch 4/100
124/124 [==============================] - 36s 294ms/step - loss: 5.2327 - accuracy: 0.1227
Epoch 5/100
124/124 [==============================] - 34s 276ms/step - loss: 4.9947 - accuracy: 0.1482
Epoch 6/100
124/124 [==============================] - 32s 261ms/step - loss: 4.7281 - accuracy: 0.1777
Epoch 7/100
124/124 [==============================] - 33s 269ms/step - loss: 4.4269 - accuracy: 0.2188
Epoch 8/100
124/124 [==============================] - 36s 291ms/step - loss: 4.1254 - accuracy: 0.2458
Epoch 9/100
124/124 [==============================] - 34s 277ms/step - loss: 3.8408 - accuracy: 0.2733
Epoch 10/100
124/124 [==============================] - 34s 273ms/step - loss: 3.5683 - accuracy: 0.3036
```

Fig. 3. Less Accuracy

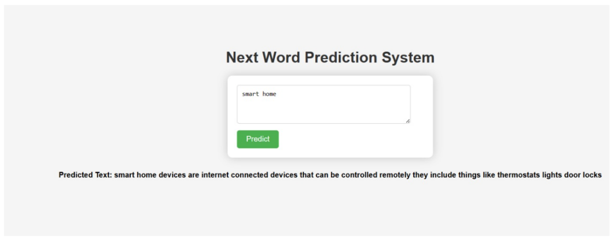
F. Make Prediction

To make predictions as the test to show the implementation of the built model, here we need to make a function which can predict the next word. . Here is the example given in Figure 5. It combines the sequences of probability of sequences of words to get the maximum probability of predicting the word.



```
Epoch 91/100
124/124 [==============================] - 35s 283ms/step - loss: 0.0377 - accuracy: 0.9869
Epoch 92/100
124/124 [==============================] - 36s 293ms/step - loss: 0.0385 - accuracy: 0.9869
Epoch 93/100
124/124 [==============================] - 36s 288ms/step - loss: 0.0366 - accuracy: 0.9871
Epoch 94/100
124/124 [==============================] - 36s 287ms/step - loss: 0.0365 - accuracy: 0.9861
Epoch 95/100
124/124 [==============================] - 36s 290ms/step - loss: 0.0489 - accuracy: 0.9838
Epoch 96/100
124/124 [==============================] - 36s 292ms/step - loss: 0.1905 - accuracy: 0.9483
Epoch 97/100
124/124 [==============================] - 36s 293ms/step - loss: 0.0792 - accuracy: 0.9793
Epoch 98/100
124/124 [==============================] - 36s 292ms/step - loss: 0.0430 - accuracy: 0.9856
Epoch 99/100
124/124 [==============================] - 36s 293ms/step - loss: 0.0381 - accuracy: 0.9871
Epoch 100/100
124/124 [==============================] - 37s 298ms/step - loss: 0.0364 - accuracy: 0.9876
```

Fig. 4. Better Accuracy



smart home

Predicted Text: smart home devices are internet connected devices that can be controlled remotely they include things like thermostats lights door locks

Fig. 5. Make prediction

IV. CONCLUSION

Next word prediction is one of the NLP fields because it's about mining the text. The researcher here used the LSTM model to make the prediction with 200 epochs. The result showed that it maintained get accuracy of 75 percent while the loss was 55 percent. Based on that result, it could be said the accuracy is good enough. It also showed that it's better than two of the two other research which used different models. The model could be used to predict the next word by giving the input of the destination.

ACKNOWLEDGMENT

We would like to take this opportunity to express our hearty gratitude and sincere thanks to our guides Mahir Mahbub sir for their invaluable assistance with our project. We extend our heartfelt appreciation to everyone who has contributed to the success of this project. Our gratitude goes to our dedicated team members who collaborated tirelessly, as well as the teachers and mentors who provided valuable guidance and support throughout this research endeavour. We also express our thanks to the academic institutions and organizations that made this research possible by providing access to the necessary resources and datasets.

REFERENCES

[1] Jordan, Michael I., and Tom M. Mitchell. "Machine learning: Trends, perspectives, and prospects." Science 349, no. 6245 (2015): 255-260.
[2] Sahoo, Abhaya Kumar, Chittaranjan Pradhan, and Himansu Das. "Performance evaluation of different machine learning methods and deep-learning based convolutional neural network for health decision making." In Nature-inspired computing for data science, pp. 201- 212. Springer, Cham, 2020

[3] Prajapati, Gend Lal, and Rekha Saha. ”REEDS: Relevance and enhanced entropy based Dempster Shafer approach for next word prediction using a language model.” Journal of Computational Science 35 (2019): 1-11.

[4] Ambulgekar, Sourabh, Sanket Malewadikar, Raju Garande, and Bharti Joshi. ”Next Words Prediction Using Recurrent NeuralNetworks.” In ITM Web of Conferences, vol. 40, p. 03034. EDP Sciences, 2021.

[5] Stremmel, Joel, and Arjun Singh. ”Pretraining federated text models for next word prediction.” In Future of Information and Communication Conference, pp. 477-488. Springer, Cham, 2021.

[6] Xiaoyun, Qu, Kang Xiaoning, Zhang Chao, Jiang Shuai, and Ma Xiuda. ”Short-term prediction of wind power based on deep long short-term memory.” In 2016 IEEE PES AsiaPacific Power and Energy Engineering Conference (APPEEC), pp. 1148-1152. IEEE, 2016.

[7] Vargiu, Eloisa, and Mirko Urru. ”Exploiting web scraping in a collaborative filtering-based approach to web advertising.” Artif. Intell. Res. 2, no. 1 (2013): 44-54.