NEXT WORD PREDICTION USING LSTM

Article *in* Journal of Information Technology and Its Utilization · June 2022

DOI: 10.56873/jitu.5.1.4748

4 authors, including:

Afika Rianti
Universitas Pendidikan Indonesia
10 PUBLICATIONS **6** CITATIONS
SEE PROFILE

Suprih Widodo
Universitas Pendidikan Indonesia
48 PUBLICATIONS **123** CITATIONS
SEE PROFILE

NEXT WORD PREDICTION USING LSTM

Afika Rianti[1], Suprih Widodo[2] , Atikah Dhani Ayuningtyas[3], Fadlan Bima Hermawan[4]

[1,2] Indonesia University of Education, Indonesia
[3,4] Syarif Hidayatullah State Islamic University Jakarta, Indonesia
afika@upi.edu

***Abstract*– Next word prediction which is also called as 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 before had discussed it using different models such as Recurrent Neural Networks and Federated Text Models. Each researcher used their own models to make the prediction and so the researcher here. Researchers here chose to make the model using Long Short Term Memory (LSTM) model with 200 epoch for the training. For the dataset, the researcher used web scraping. The dataset contains 180 Indonesian destinations from nine provinces. For the libraries, researchers used tensorflow, keras, numpy, and matplotlib. To download the model in json format, the researcher used tensorflowjs. Then for the tool to code, the researcher used Google Colab. The last result is 8ms/step, loss: 55%, and accuracy: 75% which means it's good enough and can be used to predict next words.**

***Keywords*: Machine learning; Next word prediction; LSTM.**

I. INTRODUCTION

Machine learning addresses the question of how to build computers that improve automatically through experience[1]. Experience here can be obtained by training the machines using the model that has been built. By that training, the machine will learn through the patterns. As machine learning uses a new programming paradigm, then its concept is different from traditional programming. In traditional programming, the inputs are rules and data, then the output is the answer. While in machine learning, the inputs are answers and data, then the output is rules. This rule is what will be used later to build the model that can recognize certain types of patterns such as in deep learning.

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 include 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 everyday, we use gadgets such as mobile and computer, then for sure we also use it 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 as 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 too between 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 suits this case as it can have a longer memory of what words are important. The aim of creating this model is to predict the next word based on the input which the result should predict correctly. The researcher also would like to know how accurate it's.

II. METHODE

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 180 Indonesian destinations from nine provinces. The total of destinations taken in each province is 20. Those nine provinces are West Java, Central Java, D.I.Y, East Java, Jakarta, Banten, Bali, West Nusa Tenggara, and East Nusa Tenggara. Those provinces were chosen based on the most visited province for Indonesian destinations. 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 Indonesian destinations. Each destination contains about 4-7 words. The dataset is then being inserted in a variable divided by “\n” for each destination. To see the dataset, we could use the print command. The destination data look like this : "Dago Dreampark Lembang \n Gunung Tangkuban Perahu Bandung \n Museum Geologi Bandung \n Museum Konferensi Asia Afrika \n Curug Citambur Cianjur \n Kawah Putih Telaga Bodas Garut,...” and so on.

B. Model

The model architecture shown in the figure 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.

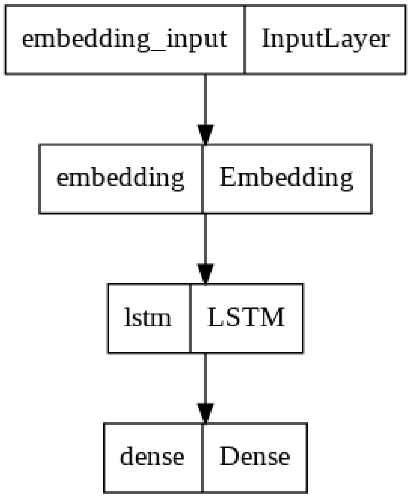
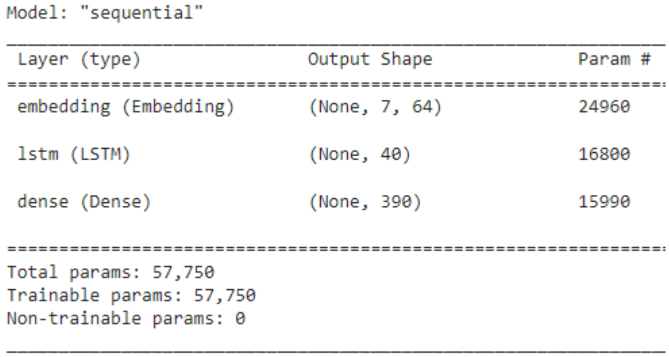


Fig. 1. Architecture model

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, 7, 64)             24960

 lstm (LSTM)                 (None, 40)                16800

 dense (Dense)               (None, 390)               15990

=================================================================
Total params: 57,750
Trainable params: 57,750
Non-trainable params: 0
_________________________________________________________________
```

Fig. 2. Model Summary

D. History

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: 55%, and accuracy: 75%. Compared to the two other reaseaher that has mentioned in the introduction, this accuracy is good enough as it managed to get 75%. While the other research, say the one using RNN just got around 54%-55% and the one using the Pre Training Federated Text Model got around 21%-22%. The difference of the result could happen because there are some differences such as in the dataset, code and for sure the model.

E. Graph

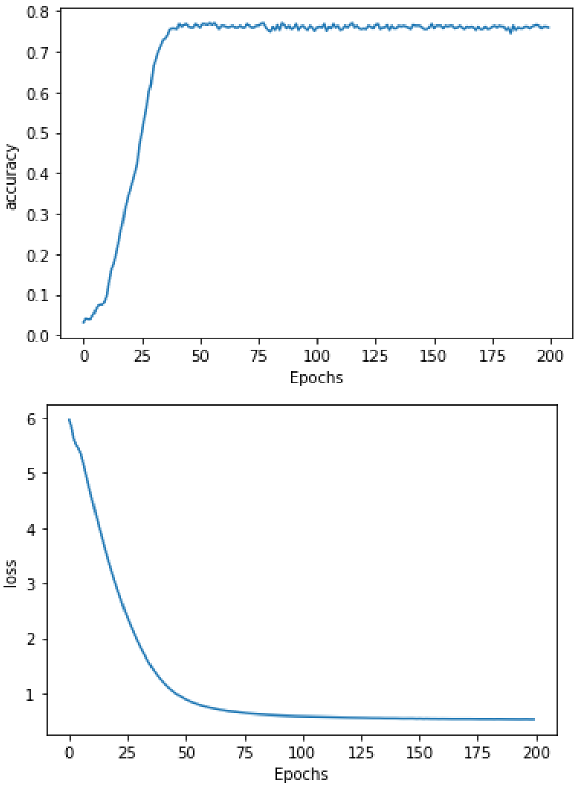
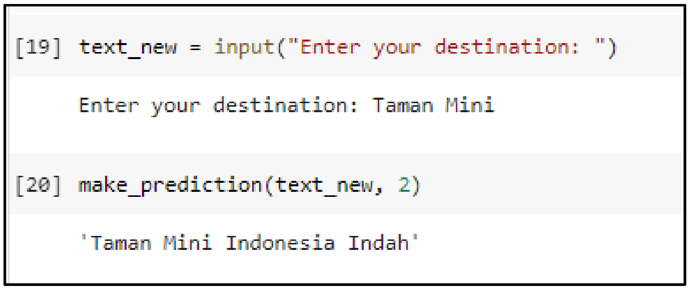


Fig. 3. Accuracy and loss graph

From the figure above, we could see that the accuracy graph increases rapidly in the first 25 epochs, then when it hits the middle of 25 and 50, it slowly stops and just goes up and down with little change. Then for the loss graph, it decreased rapidly at the first 50 epochs then after that it became slow till it hit the loss at the point between 1 to 0.

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. The parameters are the input of what destination is looking for by the user and how many next words want to be predicted. Here's an example of what the input and output looks like.



```
[19] text_new = input("Enter your destination: ")

     Enter your destination: Taman Mini

[20] make_prediction(text_new, 2)

     'Taman Mini Indonesia Indah'
```

Fig. 4. Make prediction

From that figure, we could see that the input was "Taman Mini", the result of two next word predictions is Indonesia Indah which means it's true as what's in the dataset.

G. Download Model

To download the model, the researcher uses SavedModel to save the model in h5 format. Then to get the model in json format, the researcher use tensorflowjs convertor. The saved models which are saved in online files colab storage then can be downloaded to local storage by clicking file on the right side and clicking on download.

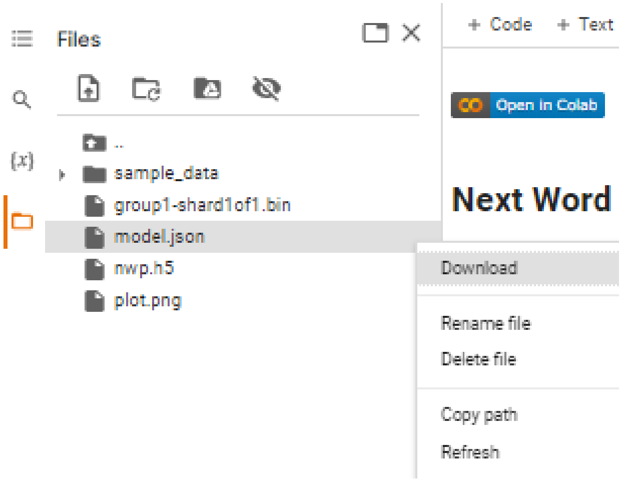


Fig. 5. Download the model

IV. Conclusion

Next word prediction is one of 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 to get accuracy 75% while the loss was 55%. Based on that result, it could be said the accuracy is good enough. As it also showed that it's better than two of the two other researches which used different models. The model could be used to predict the next word by giving the input of the destination.

V. Acknowledgment

The authors gratefully acknowledge the contributions of the Bangkit team which provides us the courses and all the classes we had during the independent study so that we could gain knowledge especially in the machine learning field.

VI. 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 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 Asia-Pacific 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.