Twitter Sentiment analysis using CNN (November 2022)

Pongamorn Trakarnkulphun

***Abstract*—Sentiment analysis is a common natural language processing problem that involves text classification based on their sentiments. There are many approaches to this problem. The most common approaches are RNN, BERN, SVM, etc. This project's goal is to perform sentiment analysis on a Twitter dataset using CNN, which is more common in machine learning tasks involving images. This project also compares the results of the CNN models with and without the numbers removal technique at the data preprocessing stage. The comparison is done by implementing the models of both cases by using the TensorFlow package and comparing the performance based on the accuracy test on the test set. The accuracy of the model with and without the technique is 0.9943 and 0.9952, respectively. The result shows that the difference in the performances of the models is insignificant. Overall, the result of the model is satisfactorily great.**

***Index Terms*— Natural Language Processing, Machine learning, Sentiment analysis**

I. Introduction

SENTIMENT analysis is one of the most popular problems in natural language processing. It is a problem that focuses on classifying the given texts based on their tones. Two main solutions to this problem are machine learning-based and lexicon-based. Both solutions have their own advantages. Briefly, machine learning-based approach has more accuracy, while lexicon-based approach can be applied more broadly. This paper focuses on machine learning-based solution.

Among machine learning-based approaches, there are many machine learning models that can be used to solve this problem. Recurrent neural network is definitely one of the best choices because of its great performance on natural language processing problems. Support vector machine and Naïve Bayes are commonly used as well. Transformer neural network seems to perform the best.

Aside from all of these approaches, convolutional neural network, normally used in computer vision tasks, unexpectedly performs decently well too. This seems to happen because of the ability to detect features of the dataset. Another interesting thing to consider in this problem is whether removing numbers in the dataset affect the performance of the model. This may increase the performance of the model since numbers are believed to not carry meanings significant enough to change the sentiment of the texts.

This project aims to conduct sentiment analysis on a twitter dataset with good performance. This project chooses to use CNN as the approach to sentiment analysis of the selected twitter dataset. Additionally, another goal of the project is to compare the result of the model with and without removing numbers.

The dataset chosen in this project has negative, positive, and other sentiments. In this project, only negative and positive results are considered, and the other sentiments are ignored for the simplicity of the project. This dataset also includes languages aside from English. Those languages are out of the scope of the project as well.

II. Description

A. Objective

This project has two main objectives. The first one is to perform sentiment analysis using one dimensional convolution neural network on a particular dataset. The second objective is to apply numbers removal technique during the data cleaning stage and compare its result to the one without the technique.

B. Data preparation

The dataset is retrieved from Kaggle website [1]. The dataset is mainly in English, but it also has other languages as well. However, English is the only language considered in this project, so the data in other languages is removed. The dataset includes 4 different sentiments namely positive, negative, litigious, and uncertainty. However, this project only considers positive and negative sentiment. After removing languages and sentiments that are out of the scope, the dataset consists of 248516 texts labeled as positive and 244146 texts labeled as negative. The dataset is slightly imbalance, but not to the point that need any special methods.

The data cleaning process is conducted by removing the special characters other than alphabet characters, numbers, and spaces from every word in the dataset. In the data cleaning process, numbers are sometimes removed. Consequently, there are two clean datasets, one with numbers and one without numbers. The dataset is then split into training set, test set, and validation set with the ratio of 8:1:1, relatively. The validation set is used for dropout, while the test set is used to calculate the result of both models with and without numbers removal.

After cleaning and splitting the dataset, the word embedding is performed using a pre-trained word embedding. The chosen pre-trained word embedding is GloVE [2]. Word embedding is

a way to represent a word by a vector. Word embedding is important because, in order to use texts as inputs of the neural network, those texts need to be in numerical form. This can also be done by tokenization, using a number to represent a word. However, word embedding is better for this task because vectors are more meaningful. For this project's chosen word embedding, each word is represented by a 100d vector. In this project, only the top 20000 most appeared words in the training set are recognized, while other words are unknown. Therefore, those other words are represented as vector 0. After the word embedding, the texts are padded to the length of the 95th percentile of texts in the training set. Padding is performed by removing words and adding vector 0 until the text gets to the specified length.

After all of these data preparation processes, each text in the dataset now contains the same number of vectors representing words in the text. This dataset is now ready for training.

C. Model

The chosen model for this project is the convolutional neural network. The selected optimization algorithm is the Adam optimizer. The criterion is a binary cross-entropy loss function. The number of epochs is 30, and the batch size is 256. The architecture of the neural network consists of 3 convolutional layers, 2 max-pooling layers, 1 global max-pooling layer, 2 dense layers, and 1 output layer. The dropout is also performed on the second dense layer to prevent the overfitting problem. The Relu function is chosen as the activation function. The function is applied to every convolutional layer and the second dense layer. The output layer has a sigmoid function as the activation function since the output of this classification task is binary. The diagram showing the architecture of this model is shown below.

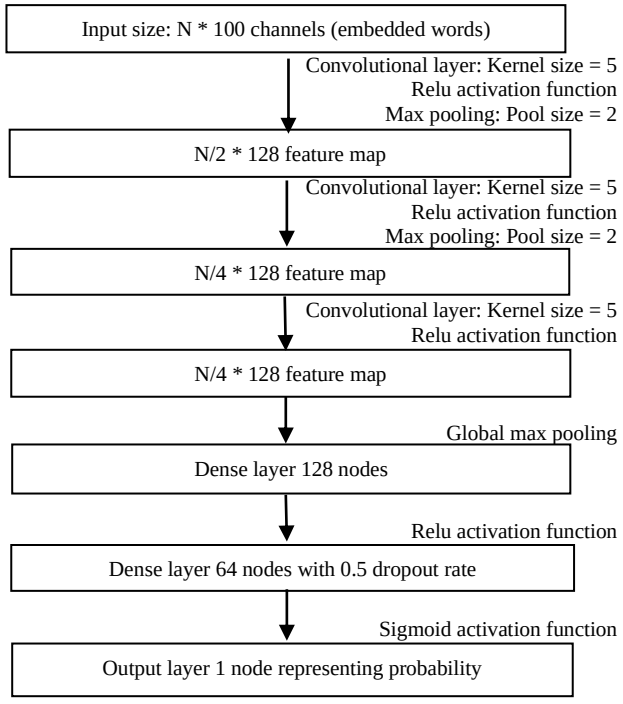

Fig. 1. The diagram shows the architecture of the CNN model. N represents the length of the text input after padding.

D. Implementation

The neural network models are implemented in python using the TensorFlow package. The experiment regarding numbers removal is done by running the source code two times, one with removing numbers and another one without removing numbers, on the same dataset. At the end of the code, the accuracy score is calculated, and the line graph is plotted. These results can be seen in the evaluation section.

III. Evaluation

The dataset is only slightly imbalanced. Therefore, the chosen evaluation metric is the accuracy score since it is still an accurate way to evaluate the performance of the model. The results are shown in the table below.

Numbers removal	Accuracy
Not removing numbers	0.9952
Removing numbers	0.9943

Fig. 2. The table shows the accuracy of two models based on the test set.

Based on the accuracy score on the test set, the difference between accuracy scores is just 0.0009, meaning that removing numbers or not does not have any significant effect on the result. In fact, the result of the experiment suggests that removing numbers during the data preprocessing step slightly reduces the accuracy score on the test set in this case. One possible reason for this result is that the dataset may not contain that many numbers. Therefore, removing numbers does not influence performance.

The two following line graphs are plotted by the matplotlib package using the histories of the model training. It shows the accuracy scores on both the training and validation set across the epoch. As seen in figure 3 below, the accuracy of the training set is close to the validation set, indicating that there are neither overfitting nor underfitting problems. In contrast, figure 4 shows that the model encounters a small amount of overfitting problem. However, the problem is not really severe since the space between the accuracy of the training set and the validation set is not that big.

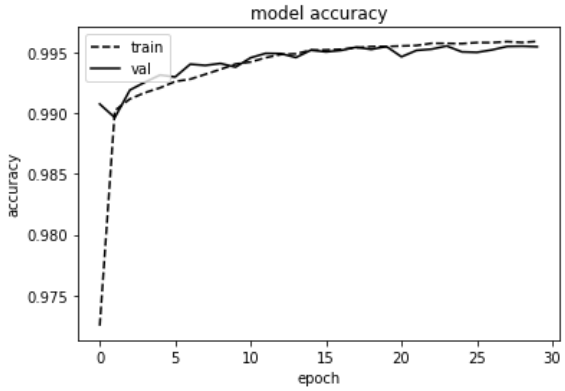

Fig. 3. The line graph shows the accuracy on the training set and validation set of the dataset that does not remove numbers against epoch.

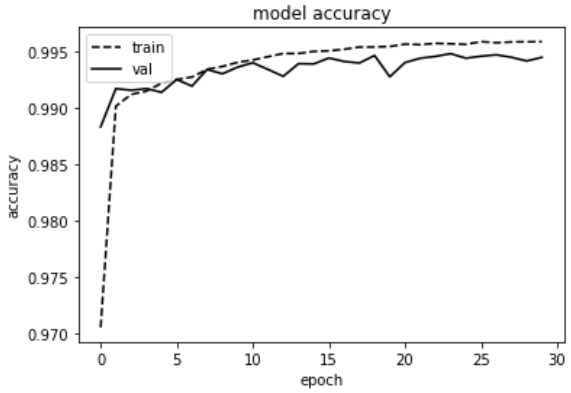

Fig. 4. The line graph shows the accuracy on the training set and validation set of the dataset that removes numbers against epoch.

IV. Related works

There are some studies regarding the effect of removing numbers in sentiment analysis tasks. According to Jianqiang and Xiaolin [3], removing numbers from the texts at the data preprocessing stage does not have any significant impact on the result of sentiment analysis. This study, however, does not use CNN but uses SVM, NB, LR, and RF instead. Nevertheless, the result their study gets is similar to the result I get. This indicates that the result regarding whether to remove numbers or not I received from conducting this project seems to be valid.

According to Goularas and Kamis [4], the chosen GloVE word embedding performs better in sentiment analysis tasks compared to Word2Vec [5], another word embedding open source, most of the time. Furthermore, the performance of the model can be increased by using multiple CNN combined with LSTM instead of pure CNN or LSTM. This means that, by adopting this method, I may be able to improve the performance of the model in this project.

One of the interesting approaches to sentiment analysis tasks is the Word2vec-BiLSTM-CNN hybrid model proposed by Yue and Li [6]. According to their study, this hybrid model outperforms traditional models tested in this study including the CNN model I use. The accuracy the CNN model gets is 0.8125, while the hybrid model gets 0.9148. Therefore, there is a great chance that the performance of this project I do can increase by changing the model to this hybrid model too. Despite its performance, this model is not chosen for this project due to its complexity.

V. Summary and Conclusions

This project tackles sentiment analysis tasks on a Twitter dataset using the convolutional neural network (CNN). The dataset is vectorized using GloVE word embedding. The project also compares the result of the model that removes the numbers in the dataset during the data preparation stage with the model that does not remove the numbers. The result is based on the accuracy score evaluation metric. The one that removes numbers has an accuracy score of 0.9943, while another one's accuracy score is 0.9952. Moreover, the one that removes numbers seems to encounter a little amount of overfitting problem when compared to the one without numbers removal. This concludes that removing numbers does not increase the performance of Twitter sentiment analysis using CNN. This result is also supported by other works about removing numbers.

This result is interesting since numbers are normally considered something that does not have specific meaning and should be removed when doing sentiment analysis. Anyway, this result may change if the dataset changes. Also, since this project only uses CNN as the model, the experiment's result may vary depending on the models chosen. Therefore, when doing sentiment analysis, it is a good practice to always do both cases and compare them by hyperparameter tuning. Nevertheless, with the accuracy score on the test set, the performance of the model seems to be acceptably high.

References

[1] M. Tariq, 2022, " Sentiment Dataset with 1 Million Tweets", *Kaggle*. [Online]. Available: https://www.kaggle.com/datasets/tariqsays/sentiment-dataset-with-1-million-tweets

[2] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global Vectors for Word Representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543.

[3] Z. Jianqiang and G. Xiaolin, "Comparison Research on Text Pre-processing Methods on Twitter Sentiment Analysis," in *IEEE Access*, vol. 5, pp. 2870-2879, 2017, doi: 10.1109/ACCESS.2017.2672677.

[4] D. Goularas and S. Kamis, "Evaluation of Deep Learning Techniques in Sentiment Analysis from Twitter Data," *2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML)*, 2019, pp. 12-17, doi: 10.1109/Deep-ML.2019.00011.

[5] T. Mikolov, K. Chen, G. Corrado and J. Dean, "Efficient Estimation of Word Representations in Vector Space", *arXiv Prepr. arXiv1301.3781*, 2013.

[6] W. Yue and L. Li, "Sentiment Analysis using Word2vec-CNN-BiLSTM Classification," *2020 Seventh International Conference on Social Networks Analysis, Management and Security (SNAMS)*, 2020, pp. 1-5, doi: 10.1109/SNAMS52053.2020.9336549.