A sentiment analysis study for twitter using the various model of convolutional neural network

Sartini, Subiyanto* and M M F Alim
Department of Electrical Engineering, Faculty of Engineering, Universitas Negeri Semarang, Indonesia

*Corresponding author: subiyanto@mail.unnes.ac.id

Abstract. Sentiment analysis about natural language processing had been developed by various methods such as machine learning. In Indonesia, fewer studies on sentiment analysis have been studied using machine learning with specific datasets. The use of machine learning results in low accuracy that is applied in a wider scope. Therefore, the deep learning method developing to improve the accuracy of sentiment analysis. This paper presents different configuration parameters based on deep learning using the Convolutional Neural Network (CNN) algorithm to improve accuracy performance. The CNN models are presented by various parameters such as the number of convolutional layers, number of filters, and filter size to analyze the model's performance. Indonesian-Sentiment-Analysis-Dataset which consists of 10,806 tweets has been used with the Word2Vec model for Indonesian as a word vector representation. The CNN models are trained on 80% of the dataset and tested on the remaining 20% of the dataset. The proposed CNN models' results are compared with machine learning algorithms such as SVM, KNN, and SGD. The CNN models performed better than machine learning and got the best accuracy of 81.4% for general sentiment analysis in Indonesian.

1. Introduction
One of the most popular social media sites is Twitter, which has 386 million active users worldwide [1]. The users of Twitter in Indonesia have reached 22.8 million from 2014 to 2019 [2]. On Twitter, users can express their opinions and feelings on trending topics [3]. The reviews and comments captured from Twitter are numerous so it takes a long time and requires large resources. Therefore, it necessary to be analyzed automatically to find out the sentiments that were conveyed [4]. These sentiments can be categorized into positive, neutral, and negative or also categories of different rating points starting from 1 to 5 stars. Some challenges of sentiment analysis on Twitter are using emoticons and slang, informal language, abbreviations, short forms that make it difficult to calculate the polarity [5].

The previous researchers have been conducted sentiment analysis methods with various tools, such as machine learning and deep learning method. Machine learning methods have been used to measure customer satisfaction in online transportation and online retail shop reviews [6–9] for measuring customer satisfaction online and online retail shop reviews. In [10], machine learning is also used to make a decision-maker system for visiting culinary using Naïve Bayes and Support Vector Machine. But, from those studies conducted, the machine learning algorithm's accuracy is still not optimal for sentiment analysis. Therefore, the researchers have been used deep learning methods with the CNN algorithm to improve the performance on the critical task of sentence classification [11–15]. The results of previous studies that using deep learning have better accuracy for sentiment analysis. In
[16] perform the ConvLstm and [17, 18] perform the CNN with Recurrent Neural Networks (RNN) that achieves comparable performance with fewer parameters on the sentiment analysis task. However, sentiment analysis using CNN algorithm has not been investigated further, especially using the general Indonesian dataset. This paper presents sentiment analysis using Indonesia-General-Sentiment-Analysis-Dataset from Twitter reviews with 10,806 tweets based on deep learning techniques to improve performance accuracy. In this work, various CNN models are developed to finish this problem using the Python programming language. The experiment uses different parameters to analyze the performances of the models.

2. Methods

This paper has used the Google Colab development environment for conducting experiments. The CNN models have been developed using TensorFlow as the core open-source library to speed up deep learning models. It is implemented using the Python programming language.

The working of this system consists of three phases. This research using Indonesian-General-Sentiment-Analysis-Dataset in research group cloud experience [19]. This dataset from Twitter consists of 10,806 tweets, including 2592 positive, 2887 negative, and 5327 neutral sentences. The system training and testing are divided into 80% training and the rest as 20% testing with 10% validation data. The CNN model to train this system consists of the input layer as the first, convolution layer, global max-pool layer, and fully connected layer with a softmax activation function.

In the input layer, the neural network using a word embedding as word vector representations become the input of CNN. The words with similar meanings can produce vectors with the same number. In this case, converting text into numbers is done using a pre-trained word vector called the Word2Vec model for Indonesian with 300 vector dimensions. To make the same length in a dataset is padded with zero vectors to each sentence. Next, the global max-pooling layer collects information and reduces representation. It is applied to the sliding window for a set of filters \( m \) with length \( h \) for each sentence. The global max-pooling layer's output is a vector with the same dimensions with the number of filters fed to the fully connected layer which does the classification task. The fully connected layer performs calculations as given in (1) [12].

\[
x = \alpha(W \ast C_{pool} + b)
\]

The activation function \( \alpha \) using ReLU as calculated in (2), the weight matrix using \( W \in \mathbb{R}^{m \times m} \), the bias is \( b \in \mathbb{R}^{m} \), and the feature map matrix is \( C_{pool} \).

\[
\alpha(z) = (0, z)
\]

In this study, the various CNN models were developed with different parameter settings for each layer. The parameters for all models such as the number of epochs are 2, batch size 32, dense 128, and dropout rate 0.5 as shown in Table 1. For the experimentation, various CNN models were developed using different parameter configurations such as the number of convolution layers using 2 or 3 with various filters from 10 to 256. The experiment using various filter sizes such as 3x3, 4x4, 5x5, and 7x7 as shown in Table 2.

| Parameter                  | Value         |
|----------------------------|---------------|
| No. of convolutional layers| 2, 3          |
| Vocabulary size            | 13,277        |
| Activation function        | ReLU          |
| Number of filters          | 10, 50, 60, 100, 128, 256 |
| Filter size                | 3, 4, 5, 7    |
| Input vector size          | 300           |
No. of fully connected layer & 1 \\ 
Dense & 128 \\ 
Dropout & 0.5 \\ 
Batch Size & 32 \\ 
Epochs & 20 \\ 

**Table 2.** Experiment settings of the proposed method.

| Various CNN Models | Convolutional Layers | No. of Filters | Filter Size |
|--------------------|----------------------|----------------|-------------|
| CNN1               | 2                    | 10             | 3, 4        |
| CNN2               | 2                    | 10             | 3, 5        |
| CNN3               | 2                    | 50             | 3, 4        |
| CNN4               | 2                    | 50             | 3, 5        |
| CNN5               | 2                    | 60             | 3, 4        |
| CNN6               | 2                    | 60             | 3, 5        |
| CNN7               | 3                    | 100            | 3, 4, 5     |
| CNN8               | 3                    | 100            | 7, 4, 3     |
| CNN9               | 3                    | 128            | 3, 4, 5     |
| CNN10              | 3                    | 128            | 7, 4, 3     |
| CNN11              | 3                    | 256            | 3, 4, 5     |
| CNN12              | 3                    | 256            | 7, 4, 3     |

3. Results and Discussion

3.1. Results

After conducting several experiments using different parameters, it gets training and testing results as shown in Table 3. The training result consists of training time, accuracy, loss, validation accuracy, and validation loss. Testing performance measures consist of accuracy, precision, recall, and f1-score. The comparison of 12 various CNN models for accuracy and loss training is shown in Figure 1. And the comparison of 12 various CNN models for performance measures is shown in Figure 2. The results obtained from the CNN model were compared with machine learning algorithms such as SVM, KNN, and SGD to analyze increased accuracy as shown in Figure 3.

**Table 3.** Training and testing of performance results for various CNN models.

| Various CNN Models | Training Performance Measures | Testing Performance Measures |
|--------------------|-------------------------------|-------------------------------|
|                    | Val-Accuracy                  | Val-Loss                      | Training Time | Accuracy | Precision | Recall | F1-Score |
| CNN1               | **0.7015**                    | **0.7270**                    | 21            | 0.705    | 0.706     | 0.705  | 0.705   |
| CNN2               | 0.7119                        | 0.7103                        | 20            | 0.710    | 0.709     | 0.710  | 0.708   |
| CNN3               | 0.8089                        | 0.5294                        | 21            | 0.778    | 0.781     | 0.778  | 0.778   |
| CNN4               | 0.8081                        | 0.5355                        | 20            | 0.774    | 0.781     | 0.778  | 0.778   |
| CNN5               | 0.8156                        | 0.5267                        | 20            | 0.798    | 0.773     | 0.774  | 0.772   |
| CNN6               | 0.8141                        | 0.5262                        | 20            | 0.792    | 0.792     | 0.792  | 0.792   |
| CNN7               | 0.8244                        | 0.4806                        | 25            | 0.880    | 0.792     | 0.800  | 0.799   |
| CNN8               | 0.8489                        | 0.4991                        | 32            | 0.806    | 0.799     | 0.800  | 0.799   |
| CNN9               | 0.8444                        | 0.4862                        | 31            | 0.809    | 0.799     | 0.800  | 0.799   |
| CNN10              | 0.8230                        | 0.5374                        | 28            | 0.806    | 0.809     | 0.809  | 0.809   |
| CNN11              | 0.8356                        | 0.5376                        | 40            | 0.812    | 0.806     | 0.806  | 0.805   |
| CNN12              | 0.8237                        | 0.5433                        | 40            | 0.814    | 0.811     | 0.811  | 0.810   |
3.2. Discussion

The performance of training and testing results shown in Table 3 shows that the best CNN model with three convolutional layers. It has filter sizes of 7, 4, 3, achieves 82.3% for validation accuracy and testing accuracy 81.4%. The model with two convolutional layers achieves 81.5% for validation accuracy and testing accuracy 79.8%. It has filter sizes of 3, 4. The model's training time is getting longer by adding the number of convolutional layers and the number of filters.
The comparison of various CNN models for validation accuracy and validation loss is shown in Figure 1. The orange chart shows the best model for validation accuracy represents CNN 12 and the worst is shown by the red chart represents CNN 1. The other color charts get a higher value compared to the red but lower than the orange chart. Same as validation accuracy, the orange chart represents CNN 12 has the best loss and the red chart represents CNN 1 has the worst loss.

In the comparison of accuracy, precision, recall, and f1-score using various CNN models as shown in Figure 2, it has been observed that the value of each model has no significant difference for accuracy, precision, recall, and f1-score. The best performing model is CNN 12 with 81.4% accuracy, 81.1% precision, 81.1% recall, and 81.0% f1-score. The worst performing model is CNN 1 with 70.5% accuracy, 70.6% precision, 70.5% recall, and 70.5% f1-score. The confusion matrix for CNN 12 as the best model as shown in Table 4, represents the predicted results with the actual classification.

| Actual     | Predicted | Neutral | Positive | Negative |
|------------|-----------|---------|----------|----------|
| Neutral    | 366       | 68      | 66       |
| Positive   | 45        | 440     | 15       |
| Negative   | 68        | 17      | 415      |

The results obtained from the CNN model were compared with machine learning algorithms to analyze increased accuracy. The machine learning algorithms such as SVM, KNN, and SGD have been applied using the same dataset. The proportion of training and testing data using the same comparison is 80% and 20%. This refers to research conducted by [19]. The comparative results accuracy of machine learning with CNN models as shown in Figure 3. In this case, after experimenting using various parameters of CNN models, it can be seen that the best CNN models can achieve 81.4% for testing accuracy. It is better than machine learning algorithms with a maximum accuracy of 62.1% achieved by the SGD, while the SVM gets an accuracy of 61.4%, and KNN with an accuracy of 52.3%.

4. Conclusion
In this study, we apply the various CNN models to conduct sentiment analysis of the public Indonesia dataset. The experiments have been carried out to show that the CNN models compared to machine learning methods have better accuracy performance to the sentiment analysis task. In this model, these sentiments can be categorized into neutral, positive, and negative. The experiments were carried out using various CNN models with different parameter configurations such as the number of convolutional layers, number of filters, and filter size. The result shows the best model is CNN 12, it achieves 81.4% for accuracy with 7, 4, 3 filter size, and 256 number of filters. The CNN models get better accuracy performance compared to machine learning algorithms such as SVM, KNN, and SGD.

Acknowledgments
The researchers would like to thank the Faculty of Engineering, Universitas Negeri Semarang for the funds provided. This study was supported by UNNES Electrical Engineering Students Research Group (UEESRG), Department of Electrical Engineering, in facilitating the implementation of our project.

References
[1] Clement J 2020 Global social networks ranked by number of users
[2] Statista 2015 Number of Twitter users in Indonesia from 2014 to 2019 Stat. Res. Dep.
[3] Yoo S Y, Song J I and Jeong O R 2018 Expert Syst. Appl. 105 102
[4] Vashishtha S and Susan S 2019 Expert Syst. Appl. 138
[5] Medhat W, Hassan A and Korashy H 2014 Ain Shams Eng. J. 5 1093
[6] Anastasia S and Budi I 2016 Int. Conf. Adv. Comput. Sci. Inf. Syst. (ICACSIS) p 359
[7] Saragih M H and Girsang A S 2017 Int. Conf. Sustain. Inf. Eng. Technol. (SIET) p 24
[8] Windasari I P, Uzzi F N and Satoto K I 2017 Proc. Int. Conf. Inf. Technol. Comput. Electr.
[9] Fiarni C, Maharani H and Pratama R 2016 Fourth Int. Conf. Inf. Commun. Technol. (ICoICT)
[10] Somantri O and Apriliani D 2018 J. Teknol. Inf. dan Ilmu Komput. 5 537
[11] Ouyang X, Zhou P, Li C H and Liu L 2015 IEEE International Conference on Computer and Information Technology: Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing p 2359
[12] Rani S and Kumar P 2018 Arab. J. Sci. Eng. 44 3305
[13] Alharbi A S M and de Doncker E 2019 Cogn. Syst. Res. 54 50
[14] Nasichuddin M A, Adji T B and Widyawan W 2018 Int. J. Inf. Technol. Electr. Eng. 2 9
[15] Zhang Z, Zou Y and Gan C 2018 Neurocomputing 275 1407
[16] Hassan A and Mahmood A 2017 Int. Conf. Control Autom. Robot. Deep p 705
[17] Rosa R L, Schwartz G M, Ruggiero W V and Rodriguez D Z 2018 IEEE Trans. Ind. Informatics 15 2124
[18] Abid F, Alam M, Yasir M and Li C 2019 Futur. Gener. Comput. Syst. 95 292
[19] Ferdiana R, Jatmiko F, Purwanti D D, Ayu A S T and Dicka W F 2019 J. Nas. Tek. Elektro dan Teknol. Inf. 8 334