2021 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT) | 978-1-6654-4494-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/IAICT52856.2021.9532546

Sentiment Analysis Implementation For Detecting Negative Sentiment Towards Indihome In Twitter Using Bidirectional Long Short Term Memory

Muhammad Kemal Hernandi[1], Suryo Adhi Wibowo[2], and Suyanto Suyanto[3]
The University Center of Excellence for Advanced Intelligent Communications (AICOMS)[1,2]
Image Processing and Vision Laboratory, School of Electrical Engineering[1,2]
School of Computing[3]
Telkom University
Jl. Telekomunikasi No. 1, Terusan Buah Batu, Bandung, 40257 INDONESIA
Email: {kemalhernandi@student.,suryoadhiwibowo@,suyanto@}telkomuniversity.ac.id

***Abstract*—Sentiment analysis is the method of extracting opinions from texts written in human language. Sentiment analysis can be used to analyze and evaluate the customer experience of the services that have been provided. With easy access to social media, sentiment analysis can be applied from people's comments on social media. One of the social media that is suitable for sentiment analysis is Twitter. In this paper, we focus on negative sentiment detection using tweets on Twitter by Indihome consumers. The system is designed to apply sentiment analysis using the BiLSTM method. Using BiLSTM, the accuracy 88% is achieved.**

***Index Terms*—Artificial Intelligence; sentiment analysis; BiLSTM; Twitter; Indihome.**

I. Introduction

Along with technological developments, many large companies and home businesses use social media as a means of increasing profits such as marketing, transactions, improving the quality of service, etc. Through social media, a lot of data can be used by certain parties for their own interests. Data that can be accessed is in the form of images, videos, text, sound, etc. In today's digital era, it is easy for people to express their opinion on anything. Opinions that are publicly displayed can be processed and produce useful information.

Artificial Intelligence (AI) is present in the midst of intense business competition. AI is a field developed to study and mimic human intelligence so that it can be applied to a computer system [1]. By implementing AI in the business world, many processes can be done more easily. Natural Language Processing is a branch of Artificial Intelligence that is characterized as the automated computational processing of human language, which involves algorithms that range from text input to producing natural-looking text as output [2]. From a business perspective, people's opinions are very useful for evaluating a company. Not only to evaluate the services provided, but also to find weaknesses of business competitors. Sentiment analysis, also known as opinion mining, is one of the Natural Language Processing tools that can be applied to social media. Sentiment analysis is a process of identifying opinions from texts written in human language [3]. Twitter is a social networking and micro-blogging service known for its unique features, which include user interaction through text-based posts known as tweets [4]. This makes Twitter a very suitable platform for sentiment analysis because it is text-based. Twitter also makes it easy for users to search for posts according to the desired keywords. This allows users to view reviews and comments on a product very easily and quickly. In this paper, sentiment analysis is used to detect negative opinions on Indihome taken from the social media platform Twitter.

II. Related Work

The lexicon-based approach, machine learning approach, and hybrid approach are the three basic techniques in sentiment analysis [5].

Sentiment analysis with a lexicon-based approach using a dictionary containing a collection of words whose sentiments have been determined. Later, the word in the dictionary will be compared with the test data so that it can show the polarity of the word [6]. This method accumulates the total polarity of each positive, negative, or neutral word in the sentence under test. The polarity for the neutral word approaches 0, the positive word approaches 1, and the negative word approaches -1. However, this technique makes it difficult to assess polarization in the case of the same two words in different contexts [7].

Sentiment analysis with the Machine Learning Approach to create a model with a certain algorithm based on the data that has been trained. By using algorithms in machine learning, the computer will make predictions based on the relationship between the resulting input and output [8]. Basically, there are two forms of machine learning, they are supervised learning and unsupervised learning. Supervised learning is identical to the learning process with a dataset that has been labeled so that the output is in the form of classification. While unsupervised learning is a technique that has no definite target, it is usually used for clustering [9]. Machine learning methods, on the other hand, necessitate a large amount of training data in order to produce good performance.

The Hybrid Approach to Sentiment Analysis involves both lexicon-based and machine learning approaches for sentiment analysis. In its application, before the data is trained using machine learning methods, there is a process that uses linguistic rules (using a dictionary that has been made) according to a particular language [10]. By combining the two techniques, it is hoped that the deficiencies in each technique can be overcome and add to the strengths of each technique. The result of the Hybrid approach is determined by the methods used in both the machine learning and lexicon-based approaches.

Based on previous research, sentiment analysis can be carried out by several methods, the popular methods are using machine learning approach. Rahat et al. [11] conducted a study comparing the sentiment analysis performance between Support Vector Machine (SVM) and Naive Bayes method based on accuracy, precision, recall, and F1 parameters . From the results of the performance comparison using the SVM and Naive Bayes methods, SVM excels in every parameter tested.

Meanwhile, Lee et al. [12] found shortcomings in the SVM and Naive Bayes methods, namely that they can only be used in certain domains. It means that by using these trained classifiers into another domain will drop the performance. In addition, Long Short Term Memory (LSTM) method also has a weakness, namely the long training time, this makes it difficult for this method to do parallel processing.

Research conducted by Jozefowicz et al. [13] concluded that GRU has better performance than LSTM in all aspects except for Language Modeling, which is the system's ability to predict patterns in words. Therefore, the aim of this paper is to evaluate the output of sentiment analysis using another method called BiLSTM which is compatible to capture the context information [14], as well as to show the sentiment analysis results on the dataset. We hope this paper can be a reference for sentiment analysis research , especially for sentiment using Indonesian language.

III. Methodology

Bidirectional Long Short Term Memory (BiLSTM) is a deep learning method that combines the Bidirectional Recurrent Neural Network (BiRNN) algorithm with Long Short Term Memory (LSTM). LSTM arises from the problem of RNN where the more data is trained, the more gradients are removed. This makes RNN lack the ability to understand information contextually [14]. The difference between RNN and LSTM is the process in one module. Based on Fig. 1, it can be seen that in one RNN module it only consists of one single tanh layer activation function which functions for the classification of data groups.

In contrast to RNN, LSTM consists of four main components, namely input gate (i_t), output gate (o_t), forget gate (f_t), and cell activation [4]. Components in an LSTM module can be seen in Fig. 2. The t symbol shows the current time, t_{-1} indicates the previous time, and t_{+1} indicates the following time. Meanwhile, C_t and C_{t-1} show the current and previous hidden states. Then x_t represents the current input vector. The h_t represents the current output vector and $h_t - 1$ represents previous output vector. Sigmoid function (σ) and tanh are the activation function.

LSTM has a feature to select the information entered into its cell. When information is entered into the LSTM module, the information will be entered into the forget gate.In the first stage, information will be entered into the forget gate as shown in Fig. 3. Forget gate serves to determine which information should be removed. The forget number output will be a value 1 or 0. The 0 represent the discarded information, while the 1 represents the stored information [15]. In the second stage, the input gate in Fig. 3 will determine the updated information. At the same time, the tanh layer will create a vector with the new values. In the third stage, the results from process one will be combined. So that the current output is in the form of updated information. The last stage is to provide the results in the output. The last stage consists of the output gate which can be seen in Fig. 3 and the tanh layer. The output gate serves to determine what is the output. The tanh layer is used to change the output value between 1 and -1. Then these two processes are multiplied to produce a cleaned output. We can define the whole process mathematically by using equations 1-5 where U and W are weight matrices of the network. Meanwhile b is the bias vector.

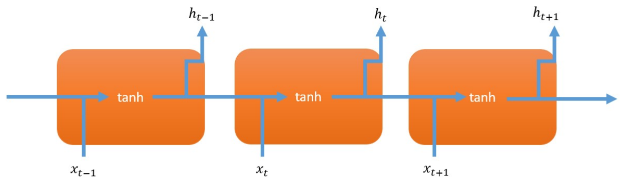

Fig. 1. Standard RNN

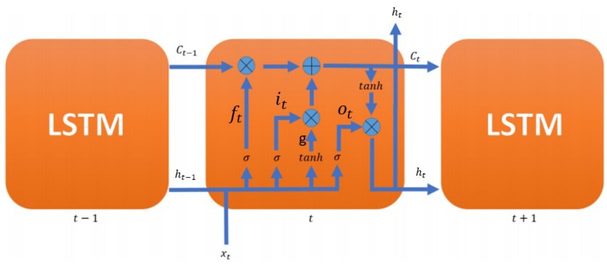

Fig. 2. Repeating Module In LSTM

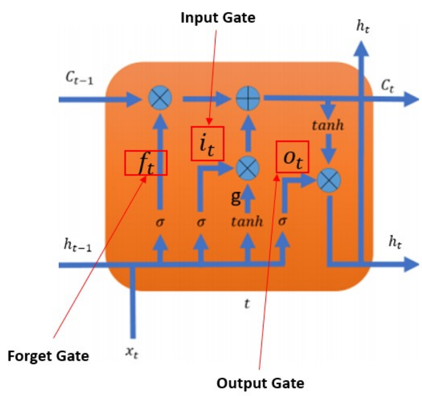

Fig. 3. Forget Gate, Input Gate, and Output Gate

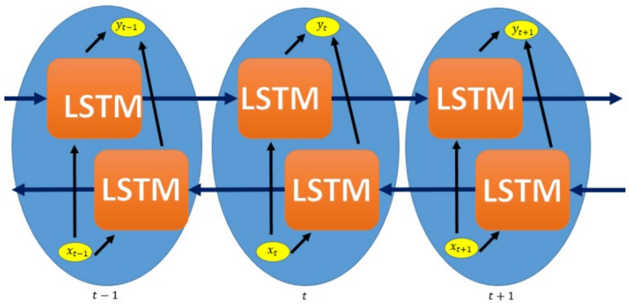

Fig. 4. BiLSTM

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \quad (1)$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \quad (2)$$

$$o_t = \sigma_g(W_0 x_t + U_o h_{t-1} + b_o) \quad (3)$$

$$c_t = f_t o c_{t-1} + i_t o \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \quad (4)$$

$$h_t = o_t o \sigma_h(c_t) \quad (5)$$

However, the information on the LSTM is run unidirectional so that the current output comes only from the previous input. Implementing LSTM bidirectional (two-way) can make the output not only come from the previous input, but also the output can be the input for other modules.

The bidirectional concept can be seen in Fig. 4. The symbols x_{t-1}, x_t, x_{t+1} in Fig. 4 represent the previous, current, and subsequent input vectors. Meanwhile, y_t-1 , y_t , and y_{t+1} represent the previous, current, and subsequent output vectors. By applying two-way LSTM (bidirectional), the model can understand the information provided according to the context as a whole [15].

IV. Experimental Setup

A. Dataset

For this sentiment analysis process, we need a more specific dataset related to the quality of internet services. Therefore, we obtained the data through the web scraping process on the twitter explore page with the keyword indihome. The labeling process is done manually by dividing the tweet into two labels. Label 1 for tweets containing negative sentiment. Tweet labeled 1 contains tweets of people who complain, say harshly, and give criticism to Indihome. Meanwhile, Label 0 consists of positive and neutral sentiments. Tweets labeled 0 contain tweets of people who praise, thank, follow up on services, and have nothing to do with products or services provided by Indihome. After labeling, we balance the number of data so that class imbalance will not occurs. The total data consists of 6045 tweets labeled 0 and 6006 tweets labeled 1. Then, we used 90% of the data for training and 10% for testing.

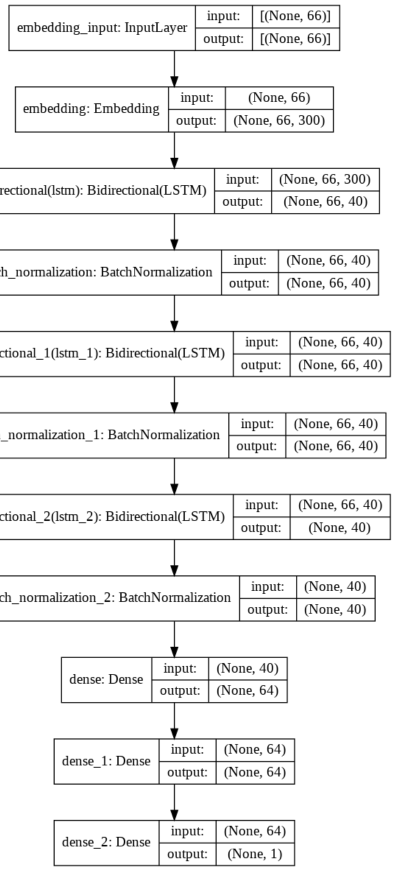

Fig. 5. Proposed Model

B. Data Cleaning

The text preprocessing is needed to clean the words to be trained. The more variations of the word in the word, the more variations of the word that will be trained will interfere with the training process. The preprocessing text this time consists of stopword removal in Indonesian, replacing URL, replacing username, replacing consecutive letters, replacing emoji, removing non-alphanumeric, and removing symbols.

C. Word Embedding

Word embedding is a technique for converting words into vectors containing a collection of numbers by reducing dimensions based on distributional vector representations so that they can be used to compare the similarities in meaning between words [16]. Word embedding can be seen based on the similarity of meanings between words and also based on linear analogy relationships [17]. Three things that need to be considered in order to make an effective word embedding are the model construction, the training corpus, and the parameters used when conducting training. In this experiment, we use a model that has been made by Grave et al. based on the words in Wikipedia using CBOW with position-weights in dimensions 300 [18].

D. Proposed Model

In this paper, the model architecture used can be seen in Fig. 5. The model consists of input layer, embedding layer, BiLSTM layer, batch normalization, dense layer, and output layer. At the input layer, all training data will be collected for the next word embedding process. The embedding layer converts text data into vectors with dimensions of 300. The training data which is already as vectors will be processed in the BiLSTM layer. The output from the BiLSTM layer will be processed through batch normalization. Batch normalization used to normalize activation in the input before moving on to the next layer. This can speed up the training process by reducing the internal covariance shift [19]. Dense layer is a fully connected layer. In this model, the first two dense layer used relu activation function, while the last dense layer used sigmoid activation function. All values of the hyperparameters are shown in Table I.

TABLE I
HYPERPARAMETER LIST

Hyperparameters	Value
Learning Rate	0.001
Batch Size	32
Epoch	12
Optimizer	Adam
Loss Function	Binary Cross Entropy Loss Function
Max Length	66
BiLSTM Nodes	40

E. Performance Evaluation

To evaluate the performance of the model in this experiment, we used the parameters of precision, recall, F1 Score, and accuracy. To get the value of the previous parameter, it can be calculated mathematically through equations 6-9.

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \tag{6}$$

$$Precision = \frac{TP}{(TP+FP)} \times 100\% \tag{7}$$

$$Recall = \frac{TP}{(FN+TP)} \times 100\% \tag{8}$$

$$F1\ Score = \frac{2 \times (Precision \times Recall)}{(Precision + Recall)} \tag{9}$$

True positive (TP) refers to the classification of positive data as true. True negative (TN) refers to a situation in which negative data is counted as true. False positive (FP) refers to the classification of positive data is classified incorrectly. False negative (FN) refers to a situation in which negative data is classified incorrectly.

V. RESULTS AND DISCUSSION

In the training process, we used 10845 tweets as training data and 1206 test data which were separated randomly. After carrying out the training process using an epoch of 12, it can be seen in Fig. 6 that the training process is getting better until the 9th epoch. After entering the 10th epoch, the training performance is getting worse. We can indicate the performance by looking at the loss and accuracy for each step. If the accuracy getting higher and the loss is getting lower, it indicates a good performance. The training process takes 726 seconds or 60.5 seconds/epoch.

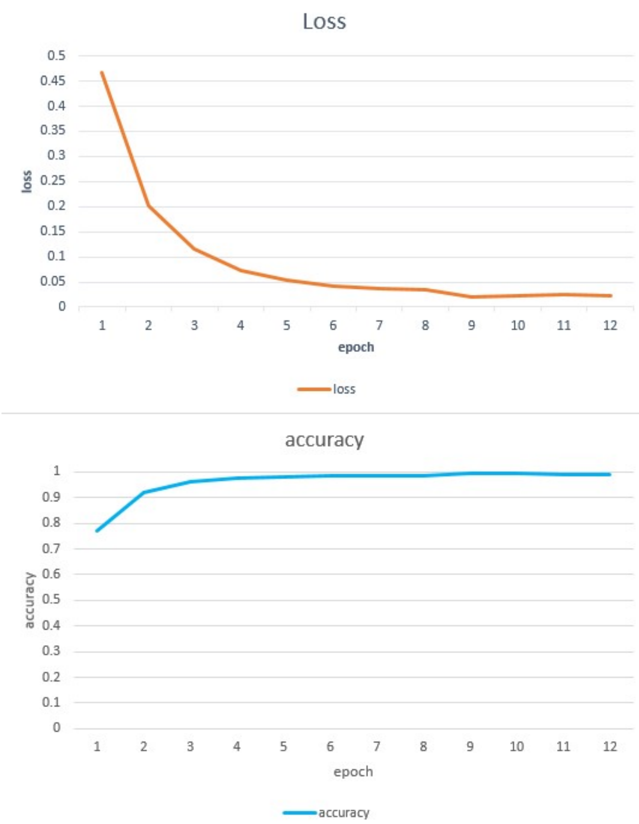

Fig. 6. Training Evaluation

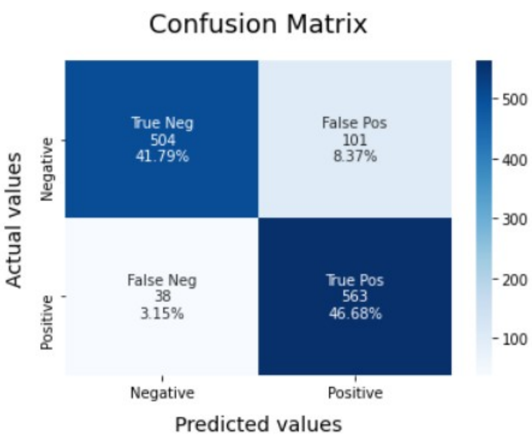

Fig. 7. Confusion Matrix

The confusion matrix in Fig. 7 classifies the test results in the model into true negative, false positive, false negative, and true positive. From 1206 test data, 504 tweets entered into true negative, 101 tweets entered into false positives, 38 tweets entered into false negatives, and 563 tweets entered into true positives. The positive confusion matrix indicates label 1 (negative sentiment towards Indihome) and the negative confusion matrix indicates label 0 (positive and neutral sentiment towards Indihome). From this confusion matrix, we can see that the wrong tweets prediction is 139 out of 1206 total tweet for testing. The result of the confusion matrix will be used to determine performance evaluation.

TABLE II
CLASSIFICATION RESULT

Classes	Precision	Recall	F1-score	Support
0	0.93	0.83	0.88	605
1	0.85	0.93	0.89	601
Macro Average	0.89	0.88	0.89	1206

After getting the value from the confusion matrix, we proceed to evaluate the performance by calculating the F1 score, accuracy, precision, and recall values. We can see the result in Table II. For the parameter that is prioritized in this experiment is the accuracy level of the model. Accuracy is the preferred parameter because the amount of data labeled 0 and 1 is balanced. From the test results, an accuracy value of 88% was obtained using equation 6. The F1 score reached 89%, 88% for recall, and 89% for precision. To evaluate tweets that contain negative sentiments, the output of this experiment is a text file that attaches any tweet that has a value of 0 or 1.

VI. CONCLUSION

In summary, we implement sentiment analysis for business purposes, especially to detect negative sentiment with a tweet-shaped dataset on twitter with the keyword Indihome. The tweets used in this experiment were in Indonesian. The results of the sentiment analysis using the model proposed in this paper achieve an accuracy of 88%. We found that there are many factors that can be developed to improve the performance of this experiment. For future experiments, it would be very interesting if this model could be developed to perform multiclass sentiment analysis so that this model could be implemented in more complex business cases.

VII. ACKNOWLEDGEMENT

This research was supported by the Directorate of Research and Community Service PPM, Telkom University, and was supported by Basic Research Grant funded by Ministry of Research and Technology, and was supported by The University Center of Excellence Grant funded by Ministry of Education, Culture, Research and Technology.

REFERENCES

[1] L. Xian, "Artificial intelligence and modern sports education technology," in 2010 *International Conference on Artificial Intelligence and Education (ICAIE)*, 2010.

[2] Y. Goldberg, *Neural network methods for natural language processing*. San Rafael, CA: Morgan and Claypool Life Sciences, 2017.

[3] K. S. Sabra, R. N. Zantout, M. A. E. Abed, and L. Hamandi, "Sentiment analysis: Arabic sentiment lexicons," in *2017 Sensors Networks Smart and Emerging Technologies (SENSET)*, 2017.

[4] F. Wei and U. T. Nguyen, "Twitter bot detection using bidirectional Long Short-term Memory neural networks and word embeddings," arXiv [cs.CR], 2020.

[5] D. Maynard and A. Funk, "Automatic detection of political opinions in tweets," in *Lecture Notes in Computer Science, Berlin, Heidelberg: Springer Berlin Heidelberg*, 2012, pp. 88–99.

[6] K. Z. Aung and N. N. Myo, "Sentiment analysis of students' comment using lexicon based approach," in *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*, 2017.

[7] S. Chaturvedi, V. Mishra, and N. Mishra, "Sentiment analysis using machine learning for business intelligence," in *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*, 2017.

[8] M. Vakili, M. Ghamsari, and M. Rezaei, "Performance analysis and comparison of machine and deep learning algorithms for IoT data classification," *arXiv [cs.LG]*, 2020.

[9] E. Boiy and M.-F. Moens, "A machine learning approach to sentiment analysis in multilingual Web texts," *Inf. Retr. Boston.*, vol. 12, no. 5, pp. 526–558, 2009.

[10] A. Mukwazvure and K. P. Supreethi, "A hybrid approach to sentiment analysis of news comments," in *2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions)*, 2015.

[11] A. M. Rahat, A. Kahir, and A. K. M. Masum, "Comparison of naive Bayes and SVM algorithm based on sentiment analysis using review dataset," in *2019 8th International Conference System Modeling and Advancement in Research Trends (SMART)*, 2019.

[12] J. S. Lee, D. Zuba, and Y. Pang, "Sentiment analysis of Chinese product reviews using gated recurrent unit," in *2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService)*, 2019.

[13] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *International conference on machine learning*, 2015, pp. 2342–2350.

[14] G. Xu, Y. Meng, X. Qiu, Z. Yu, and X. Wu, "Sentiment analysis of comment texts based on BiLSTM," *IEEE Access*, vol. 7, pp. 51522–51532, 2019.

[15] "Understanding LSTM Networks," Github.io. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/. [Accessed: 19-Apr-2021].

[16] Y. Zhang, A. Jatowt, and K. Tanaka, "Towards understanding word embeddings: Automatically explaining similarity of terms," in *2016 IEEE International Conference on Big Data (Big Data)*, 2016.

[17] A. Mandelbaum and A. Shalev, "Word embeddings and their use in sentence classification tasks," *arXiv [cs.LG]*, 2016.

[18] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning word vectors for 157 languages," *arXiv [cs.CL]*, 2018.

[19] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv [cs.LG]*, 2015.