IJCCS (Indonesian Journal of Computing and Cybernetics Systems)
Vol.13, No.3, July 2019, pp. 209~218
ISSN (print): 1978-1520, ISSN (online): 2460-7258
DOI: https://doi.org/10.22146/ijccs.41236

Sentiment Analysis of Novel Review Using Long Short-Term Memory Method

Muh Amin Nurrohmat*[1], Azhari SN[2]
[1]Master Program of Computer Science, FMIPA UGM, Yogyakarta, Indonesia
[2]Department of Computer Science and Electronics, FMIPA UGM, Yogyakarta, Indonesia
e-mail: ***[1]amin.nurrohmat@gmail.com**, [2]arisn@ugm.ac.id

Abstrak

Berkembang pesatnya internet dan media sosial serta besarnya jumlah data teks, telah menjadi subjek penelitian yang penting dalam memperoleh informasi dari data teks tersebut. Dalam beberapa tahun terakhir, telah terjadi peningkatan penelitian terhadap analisis sentimen pada teks review untuk mengetahui polaritas opini pada media sosial. Namun, masih sedikit penelitian yang menerapkan metode deep learning yaitu Long Short-Term Memory untuk analisis sentimen pada teks berbahasa Indonesia.

Penelitian ini bertujuan untuk melakukan pengklasifikasian terhadap review novel berbahasa Indonesia berdasarkan sentimen positif, netral dan negatif dengan menggunakan metode Long Short-Term Memory (LSTM). Dataset yang digunakan adalah data review novel berbahasa Indonesia yang diambil dari situs goodreads.com. Dalam proses pengujian, metode LSTM akan dibandingkan dengan metode Naïve Bayes berdasarkan perhitungan dari nilai akurasi, precision, recall, f-measure.

Berdasarkan hasil pengujian memperlihatkan bahwa metode Long Short-Term Memory memiliki hasil akurasi yang lebih baik dibandingkan dengan metode naïve bayes dengan nilai akurasi 72.85%, precision 73%, recall 72%, dan f-measure 72% dibandingkan dengan hasil akurasi metode Naïve Bayes dengan nilai akurasi 67.88%, precision 69%, recall 68%, dan f-measure 68%.

Kata kunci*— analisis sentimen, review novel, Long Short-Term Memory, Naïve Bayes*

Abstract

The rapid development of the internet and social media and a large amount of text data has become an important research subject in obtaining information from the text data. In recent years, there has been an increase in research on sentiment analysis in the review text to determine the polarity of opinion on social media. However, there are still few studies that apply the deep learning method, namely Long Short-Term Memory for sentiment analysis in Indonesian texts.

This study aims to classify Indonesian novel novels based on positive, neutral and negative sentiments using the Long Short-Term Memory (LSTM) method. The dataset used is a review of Indonesian language novels taken from the goodreads.com site. In the testing process, the LSTM method will be compared with the Naïve Bayes method based on the calculation of the values of accuracy, precision, recall, f-measure.

Based on the test results show that the Long Short-Term Memory method has better accuracy results than the Naïve Bayes method with 72.85% accuracy, 73% precision, 72% recall, and 72% f-measure compared to the results of the Naïve Bayes method with 67.88% accuracy, 69% precision, 68% recall, and 68% f-measure.

Keywords*—sentiment analysis, novel review, Long Short-Term Memory, Naïve Bayes*

Received November 23th,2018; Revised July 19th, 2019; Accepted July 23th, 2019

1. INTRODUCTION

The rapid development of internet usage and social media has influenced the decision-making process, this is inseparable from the availability of the main source of information, user opinion [1]. One source of information is information about a novel. The novel is a work of fiction including events, genre of stories, and contents and characters. But not all types of novels have the same quality. Therefore, before deciding to buy or read a novel you should first find out information about the novel based on reviews given by others on social media. The reviews can help to find out whether the novel has a quality worth buying or reading. Reading the entire review can take a long time, but if only a few reviews are read, the evaluation will be biased [2].

To understand the opinions of the review, algorithms and programs are needed to process information and opinion data and to analyze the opinions of social media users called sentiment analysis [3]. Sentiment analysis or opinion mining is a field of study that analyzes one's opinions, sentiments, evaluations, attitudes, and emotions from written language. Sentiment analysis is conducted to assess the review of an object whether it tends to the positive opinion or negative opinion [4].

There are many studies that have applied sentiment analysis to a review, even Kaggle and SemEval 2017 held competitions to identify the best methods for sentiment classification. Machine learning methods such as Naïve Bayes, Maximum Entropy (ME), and Support Vector Machine (SVM) are often used in finding models and features that are appropriate to the target problem. SVM and ME are complex models so training time is longer, while Naïve Bayes is a simple and fast model [5]. But machine learning also has problems in extracting complex features and finding better types of features. Various feature extraction methods have been proposed, including single words, single-character N-gram, multi-word N-gram, and lexical syntactic. However, semantic features are rarely considered in sentiment classification. Semantic features can reveal deep and implicit semantic relationships between words which can be more useful in the classification of sentiments [6]. In this study, word2vec is used in the feature extraction process. Vector representations of word learning using the word2vec model can find semantic meanings between words and are useful in various natural language processing tasks [7].

Deep learning is one technique in machine learning that utilizes many layers of nonlinear information processing to perform feature extraction, pattern recognition, and classification [8]. The use of deep learning in Indonesian text sentiment analysis has been carried out by several researchers, including research [9] and [10].

Long Short-Term Memory (LSTM) method is one of the deep learning methods that can be applied in the field of Natural Language Processing (NLP) including speech recognition, text translator, text summarization and sentiment analysis. But the use of the LSTM method in sentiment analysis research is still rarely used especially in Indonesian texts. The LSTM method is used in research [11] and [12] in conducting sentiment analysis that has better results compared to conventional methods. This shows the LSTM method is suitable to be applied in sentiment analysis.

For these reasons, the Long Short-Term Memory (LSTM) method will be applied for sentiment analysis in the review of Indonesian novels. In this study, the LSTM method will be compared with the Naïve Bayes method.

2. METHODS

2.1 System Architecture

In general, the system architecture built consists of six parts, they are data collection, preprocessing, sentence conversion, vector conversion, sentiment classification, and system accuracy testing. In general, the processes that occur in the system are described in Figure 1.

The first stage of the system architecture was collecting data through the scraping process. The novel review data was taken from the book cataloging site, www.goodreads.com with the category of the best Indonesian novel. In this study, the collected data were grouped into three sentiment polarity, they were positive sentiment, neutral sentiment, and negative sentiment. The next stage was to conduct preprocessing which aimed to obtain clean review data so that the classification calculation process and determination of the sentiment class were more accurate. The data used in the preprocessing process included: Indonesian novels review data and slang words dictionary data. Preprocessing stages included case folding, filtering, tokenization, and slang words conversion.

After completing the preprocessing, the next step was the conversion of sentences that convert raw data into data that were ready to be used as the system input. At this stage, the preprocessing data were converted into numbers. The steps of sentence conversion included making dictionary words, converting words into numbers, and padding sentences. The next stage of vector conversion was representing a dictionary of words that had been made into vector. Vector values are taken from the training results of the Indonesian Wikipedia corpus using word2vec.

The next stage was the classification process of novel reviews using the architecture of Long Short-Term Memory and Naïve Bayes. The classification model was then used to identify sentiment review classifications on new data (test data). The final process was testing the accuracy of the system for novel data reviews. The test included the calculation of the values of accuracy, precision, recall, and f-measure.

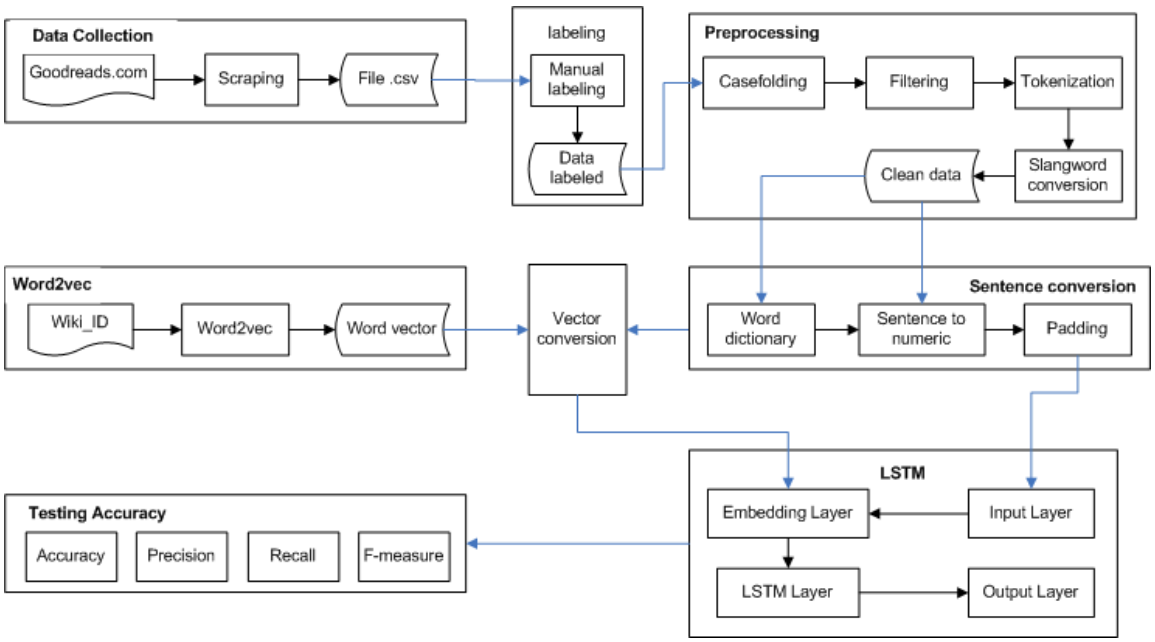

Figure 1 System Architecture

2.2 Data Collection

At this stage data collection is needed in study. The data were a collection of reviews or comments related to the reviews of novels on the site www.goodreads.com.

2. 2.1 Scraping

Data collection in this study was carried out by scraping on the site www.goodreads.com. The scraping process was done by using the python BeautifulSoup4 library (https://pypi.python.org/pypi/beautifulsoup4). Scraping a novel was done in two stages, the first by retrieving the Uniform Resource Locator (URL) data from the Indonesian Best Novel page category in the www.goodreads.com. The second stage was taking all novel reviews based on the URL of the novel on the site www.goodreads.com in the first stage.

Furthermore, the process of storing scraping data into CSV format, all data from the www.goodreads.com scraping were stored in the CSV file format based on the URL of the novel that was scrapped in the first stage. The steps in the novel review scraping process are shown in Figure 2.

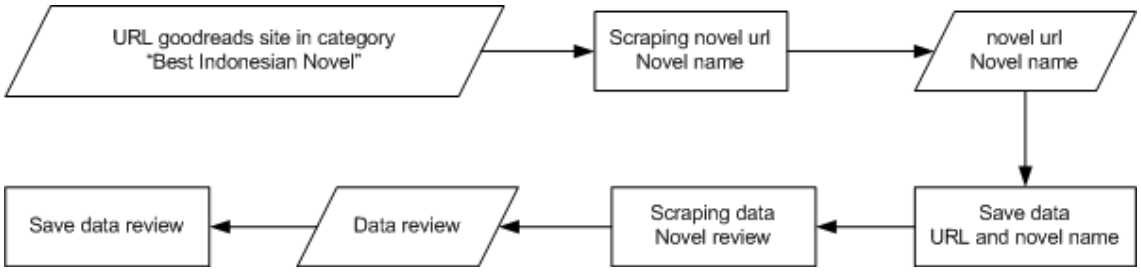

Figure 2 Scraping process

2.2.2 Data Distribution

Based on the results of the scraping process, the data review of the novel collected were 7123 reviews. The novel review data used were only Indonesian. After that, the data were labeled manually which were divided into 3 sentiment polarity, positive, neutral and negative. However, in this study, there were more positive reviews compared to the number of neutral and negative reviews.

The amount of data in each class affects the accuracy of the results. To overcome the amount of unbalanced data, solutions such as under-sampling and over-sampling are proposed. The under-sampling method is to reduce large volume classification data to balance small volume classification data, while over-sampling is to increase small volume classification data to balance large volume classification data [13]. Therefore, in this study, under-sampling method was used by reducing positive reviews to be the result of accuracy. So that the data distribution in each sentiment class were same, 1000 reviews.

2.3 Preprocessing

Preprocessing is one of the important steps in sentiment analysis. The review data that had been collected went into a preprocessing process to get clean data so that the process of making word vectors and sentiment classification were more accurate. Examples of preprocessing processes can be seen in Table 1.

Table 1 Example of preprocessing

Preprocessing	Text
	Baguss…!! gue menikmati mbaca novel ini :) @Dewi
Casefolding	baguss…!! gue menikmati mbaca novel ini :) @dewi
Filtering	baguss gue menikmati mbaca novel ini
Tokenization	[baguss] [gue] [menikmati] [mbaca] [novel] [ini]
Slangwords conversion	[baguss] → [bagus], [gue] → [saya], [mbaca] → [membaca]

The steps taken in preprocessing were as follows:

2. 3.1 Casefolding

This process changed all character letters in the document to be characters in lowercase letters. This was done to homogenize text data in processing to carry out sentiment classification.

2. 3.2 Filtering

In this process, adjustments were made by removing special characters in the review such as punctuation marks (points (.), Commas (,), question marks (?), Exclamation marks (!) And so on), numeric numbers (0-9), and other characters ($,%, *, etc.). This process also removed words that did not match the results of parsing, such as usernames beginning with the symbol "@", hashtag "#", Uniform Resource Locator (URL), and emoticons. This sign/symbol or numbers were omitted because it did not have much effect on determining labels.

2. 3.3 Tokenization

Tokenization served to break down reviews into units of words. The tokenization process was done by looking at each space in the review. Based on these spaces words can be separated.

2. 3.4 Slang words Conversion

Slang words conversion was the process of changing non-standard words into standard words. This stage was done using the help of the slang word dictionary and the equivalent in the standard word. This stage checked whether the words contained slang words or not. If the non-standard word is in the slang word dictionary, the non-standard word will be changed to the standard word. In this study, the number of slangwords dictionaries were 1088.

2.4 Sentence Conversion

Conversion of sentences was done in several steps: making dictionary words, converting words into numbers, and padding sentences. This process was done to be used as input to the system. The process started with making a dictionary of words to give the word id contained in the sentence in the review data that had gone through the preprocessing process. The second was to delete duplicate words in the reviews. The third was giving id to each word in sequence based on the words that often appear. The first word was given id = 1 and so on until the last word. The next step was to change the sentence into a number, changing the word that composes the review sentence into a number based on the dictionary of words that had been made in the previous step. The last step was to give padding to the sentence so that the sentence length was the same. Padding was done by determining the maximum length of words in a sentence or searching for the longest number of sentences in the review data. In this study, the padding given was 50 words.

2.5 Vector Conversion

Vector conversion was done by using the word dictionary that had been made and was represented in the word2vec model that had been created. The word2vec model was created

using the texts of Indonesian Wikipedia. If the word was found in the word2vec model (had a vector representation for a word), then the word2vec vector was used to represent the word. But if the word was not found in the model, it was replaced by a random vector value of all vector values on the model. The vector value of the word dictionary vectors was then obtained based on the word2vec model. The input then became a vector of n*m dimensions where n is the maximum number of reviews may have up to 50 words and we used the 300 dimensions in word2vec model. Therefore, the input was 50 x 300 = 15000 dimensions.

2.6 LSTM Classification

Long Short-Term Memory (LSTM) is a variant of Recurrent Neural Network (RNN). RNN has a problem not being able to do long-term information learning due to exploding and vanishing gradient problems. LSTM can avoid this problem by replacing the RNN node in the hidden layer with LSTM cells designed to store previous information. LSTM uses three gates, input gate, forget gate, output gate to control usage and update previous text information. Memory cells and three gates are designed to allow LSTM to read, store and update previous information [12]. In this study, word embedding and the LSTM method were used to analyze sentiment in Indonesian texts. Each layer is shown in Figure 3.

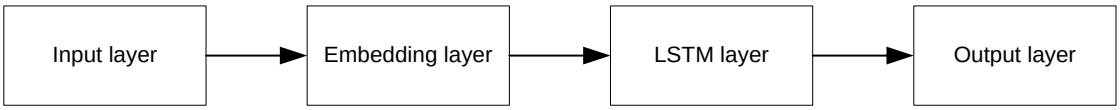

Figure 3 Network layer

2. 6.1 Input layer

This first layer was the input from LSTM which was composed of review sentences of a certain size. In this study, the input used was size 50. This means that 50 is the maximum length of the sentence from the review data that had gone through the preprocessing process.

2. 6.2 Embedding layer

The purpose of embedding layers was to study the mapping of each word in the word dictionary into vectors with lower dimensions. This layer changed the index of positive integers in the input into fixed-size vectors based on vector dimensions from word dictionary based on the word2vec model.

The input turn into n*m dimension vector where n is the longest sentence in the review and m is the dimension of the word vector. In this study, the longest sentence review was 50 words and we use the 300-dimensional word2vec model. So, the input was 50 x 300 = 15000 dimensions.

2. 6.3 LSTM layer

The first step in LSTM was to determine whether information from the X_{t-1} and X_t inputs was appropriate to pass from the cell state. This decision was made by a sigmoid layer called "forget gate". Output 1 means "let pass" and 0 means "forget information" [14]. The calculation of the forget gate value was with equation (1).

$$\mathbf{f_t} = \boldsymbol{\sigma}\left(\mathbf{W_f}.[\mathbf{h_{t-1}}, \mathbf{x_t}] + \mathbf{b_i}\right) \quad (1)$$

Then the next step was to determine the new information that were going to be stored in the cell state. The first sigmoid layer was called the input gate which determined which part to update. Next, the tanh layer created a new candidate value vector, $\tilde{c}_t$, to be added to the cell state. In the next step, the two were combined to make an update to the state. To calculate the input gate value with equation (2) and the new candidate value with equation (3).

$$\mathbf{i_t} = \boldsymbol{\sigma}\,(\mathbf{W_i}.[\mathbf{h_{t-1}},\mathbf{x_t}] + \mathbf{b_i}) \quad (2)$$

$$\tilde{\mathbf{c}}_\mathbf{t} = \mathbf{tanh}\,(\mathbf{W_c}.[\mathbf{h_{t-1}},\mathbf{x_t}] + \mathbf{b_c}) \quad (3)$$

Next, updating the old cell state, C_{t-1}, into the new cell state C_t. By multiplying the old cell state with the forget gate f_t then added $i_t * \tilde{c}_t$. To be clearer, in the equation (4).

$$\mathbf{C_t} = \mathbf{f_t} * \mathbf{C_{t-1}} + \mathbf{i_t} * \tilde{\mathbf{C}}_\mathbf{t} \quad (4)$$

The last was the output gate. First, running the sigmoid layer which determined which cell would be the output, then place the cell state through the tanh and increased the output of the sigmoid gate, so that only the part we specified was the output. Calculation of output gate with equations (5) and (6).

$$\mathbf{O_t} = \boldsymbol{\sigma}\,(\mathbf{W_o}.[\mathbf{h_{t-1}},\mathbf{x_t}] + \mathbf{b_o}) \quad (5)$$

$$\mathbf{h_t} = \mathbf{O_t} * \mathbf{tanh}(\mathbf{C_t}) \quad (6)$$

2.6.4 Output layer

This layer had a full connection to all activations in the previous layer. This layer had 3 neurons in which represented 3 classes of reviews to be classified. This layer used softmax activation so that if the activation values for each neuron was summed it was qual to 1.

2.7 Naïve Bayes

The comparison method used in this study was the Naïve Bayes (NB) method. The design process was done by retrieving the *sklearn* library from *Python.*

2.8 Test

The test process used the k-fold cross validation, where data were divided randomly into 5 parts of data with the same amount. So that the validation process was carried out repeatedly 5 times. Then the model was evaluated with new test data that had never been used in the k-fold process. The test results displayed the values of accuracy, precision, recall, and f-measure.

3. RESULTS AND DISCUSSION

This section discusses the results of sentiment classification test from the model that had been built. Sentiment classification test was done by measuring the values of accuracy, precision, recall, and f-measure from the calculation of the Long Short-term Memory method and Naïve Bayes method.

The total novel review data used were 3000 data with every 1000 data for positive, neutral and negative sentiment. The training data used was 80% of the total data, which were processed with k-fold. While 20% of the total data was used as test data.

Classification test was done by measuring the value of accuracy, precision, recall, and f-measure obtained by comparing each review that had been manually labeled with the results of the calculation of the Long Short-term Memory method carried out by the system. The number of appropriate reviews between the results of the calculation of the Long Short-term Memory method by the system with manual labeling affected the value of accuracy, precision, recall, and f-measure obtained. The greater the number of appropriate reviews, the higher the value of accuracy, precision, recall, and f-measure obtained. There were two architecture of the Long

Short-term Memory method tested, LSTM 1 layer and LSTM 2 layers which were compared with the Naïve Bayes method.

3.1 LSTM 1 layer Classification Test

The classification results of the calculation of novel review sentiment classification using the Long Short-term Memory 1 layer method is shown in
Table *2*.

Table 2 LSTM 1 layer test result

Parameter	Value	Accuracy
Word2vec architecture	CBOW	71.20%
Neuron	100	
Epoch	100	
L2 regularization	0.1	
Activation function	Sigmoid	

Based on Table 1, it shows that the best accuracy result in overall test of the LSTM 1 layer method was 71.20% with the best test parameters of CBOW architecture, 100 neurons, 100 epochs, L2 regularization values of 0.1 and sigmoid activation function.

3.2 LSTM 2 layers Classification Test

While the overall system test results from the calculation of sentiment classification using the LSTM 2 layers method showed that the accuracy value was better than the LSTM 1 layer. The overall accuracy value was 72.85%. This means that by increasing the number of layers in LSTM, it can increase the results of sentiment classification accuracy. The results of the novel review sentiment classification using the Long Short-term Memory 2 layers method are shown in Table 3.

Table 3 LSTM 2 layer test result

Parameter	Value	Accuracy
Word2vec architecture	CBOW	72.85%
Neuron	(75,75)	
Epoch	100	
L2 regularization	0.1	
Activation function	*Sigmoid*	

Based on Table 2, it shows that the best accuracy results in the overall test of the LSTM 2 layers method was 72.85% with the best test parameters of CBOW architecture, 75 neurons each layer, 100 epochs, L2 regularization of 0.1 and sigmoid activation function.

3.3 Naïve Bayes Classification Test

The Naïve Bayes method test uses the same data as the data tested in the Long Short-Term Memory method, with the training data of 2400 data (80% of data) and the test data of 600 data (20% of data). The data had also gone through the same preprocessing process while the feature extraction used was TF (Term-Frequency). The training data were processed using the k-fold which was divided into 5 parts. Furthermore, the model was tested with testing data. The results of the Naïve Bayes method test can be seen in Table 4.

Table 4 Naïve Bayes result test

Accuracy (%)	Precision (%)	Recall (%)	f-measure (%)
67.88	69.00	68.00	68.00

3.4 Comparasion of Accuracy Results

Comparison of the accuracy results from the calculation of sentiment classification using the LSTM 1 layer, LSTM 2 layers, and Naïve Bayes is showed in Table 5. The calculation of sentiment classification using the Long Short-Term Memory method had a better value of accuracy than the Naïve Bayes method.

Table 5 Comparison of accuracy result

Methods	Time	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)
LSTM 1 layer	19.21	71.20	71.00	71.00	70.00
LSTM 2 layer	26.77	**72.85**	73.00	72.00	72.00
Naïve Bayes	9.39	67.88	69.00	68.00	68.00

4. CONCLUSIONS

Based on study, it can be concluded that:

1. Classifying sentiment analysis of Indonesian novel reviews can be done using the Long Short-Term Memory method.
2. The Long Short-Term Memory method has better accuracy than the Naïve Bayes method with 72.85% accuracy, 73% precision, 72% recall, and 72% f-measure compared to the results of the Naïve Bayes method with 67.88 % accuracy, 69% precision, 68% recall, and 68% f-measure.
3. LSTM 2 layers method has better accuracy than LSTM 1 layer which produced 72.85% accuracy, 73% precision, 72% recall, and 72% f-measure while LSTM 1 layer produced 71.20% accuracy, 71% precision, 71% recall, and 70% f-measure. This means adding layers to the LSTM method can improve the results of accuracy.

There are still some weakness in this study that can be improved. Some suggestions for futher research are as follows:

1. This study is still limited to a small amount of data, which the future research case should use a larger amounts of data.
2. Use other methods for word embedding such as the FastText and Glove methods.
3. Try combining long short-term memory methods with convolutional neural networks for sentiment classification.

REFERENCES

[1] T. Parlar and S. A. Özel, "A new feature selection method for sentiment analysis of Turkish reviews," *2016 Int. Symp. Innov. Intell. Syst. Appl.*, pp. 1–6, 2016.

[2] Z. Zhang, Q. Ye, Z. Zhang, and Y. Li, "Sentiment classification of Internet restaurant reviews written in Cantonese," *Expert Syst. Appl.*, vol. 38, no. 6, pp. 7674–7682, 2011.

[3] A. M. Ramadhani and H. S. Goo, "Twitter sentiment analysis using deep learning methods," in *2017 7th International Annual Engineering Seminar (InAES)*, 2017, vol. 9121, no. JUNE, pp. 1–4.

[4] B. Liu, *Sentiment Analysis and Opinion Mining*, no. May. 2012.

[5] Z. Yangsen, J. Yuru, and T. Yixuan, "Study of sentiment classification for Chinese Microblog based on recurrent neural network," *Chinese J. Electron.*, vol. 25, no. 4, pp. 601–607, 2016.

[6] Z. Su, H. Xu, D. Zhang, and Y. Xu, "Chinese Sentiment Classification Using A Neural Network Tool - Word2vec," *Multisens. Fusion Inf. Integr. Intell. Syst.*, 2014.

[7] X. Rong, "word2vec Parameter Learning Explained," pp. 1–21, 2014.
[8] L. Deng and D. Yu, "Deep Learning: Methods and Applications," *Found. Trends® Signal Process.*, vol. 7, no. 3–4, pp. 197–387, 2014.
[9] T. A. Le, D. Moeljadi, Y. Miura, and T. Ohkuma, "Sentiment Analysis for Low Resource Languages: A Study on Informal Indonesian Tweets," pp. 123–131, 2016.
[10] F. Ratnawati, "Analisis Sentimen Opini Film Pada Twitter Menggunakan Algortime Dynamic Convolutional Neural Network," Universitas Gadjah Mada, 2017.
[11] A. Hassan and A. Mahmood, "Deep learning for sentence classification," *2017 IEEE Long Isl. Syst. Appl. Technol. Conf. LISAT 2017*, 2017.
[12] A. Rao and N. Spasojevic, "Actionable and Political Text Classification using Word Embeddings and LSTM," 2016.
[13] D. Tomar, S. Singhal, and S. Agarwal, "Weighted Least Square Twin Support Vector Machine for Imbalanced Dataset," *Int. J. Database Theory Appl.*, vol. 7, no. 2, pp. 25–36, 2014.
[14] C. Olah, "Understanding LSTM Networks," 2015. [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/. [Accessed: 17-Jan-2018].