SENTIMENT ANALYSIS USING RECURRENT NEURAL NETWORK AND LSTM IN BAHASA INDONESIA

by Lilis Kurniasari

Journal of Engineering Science and Technology
Vol. XX, No. Y (Year) PPP - QQQ

SENTIMENT ANALYSIS USING RECURRENT NEURAL NETWORK AND LSTM IN BAHASA INDONESIA

Abstract

The opinion is an important consideration towards decision making. In the current competitive business situation understanding customer opinion is the key success factor for a modern company. Automatics opinion polarization mining from online text sources such as social media, user comments and reviews have been done using various machine learning algorithm. It has been done in many languages including the Indonesian language. This research proposes the Recurrent Neural Network (RNN) and Long Short Memory Term (LSTM) to classify the sentiment polarity of Indonesian sentences. We perform an evaluation of the proposed algorithm with a dataset from travelling site reviews consists of twenty-five thousand reviews, in two classes in equal proportion (positive and negative). According to the evaluation results, the model achieves 95% of accuracy.

Keywords: Sentiment Analysis, Neural Network, Deep Learning, RNN, LSTM.

1. Introduction

Natural language processing (NLP) has been being one of the most active research areas for two decades. The popularity of NLP attracts many researchers to learns about data mining, web mining, text mining, and information retrieva [1]. It has even expanded from computer science to management and social sciences. Many related fields such as marketing, finance, politics, communication and even history start to gain benefits from natural language processing. Opinion plays an important role in almost all human activities and affects Our behaviour. Opinion drive Our beliefs, perceptions of reality, the choices We make, and to a certain extent, are conditioned on the way others see and evaluate the world. Some individuals and even organizations need the opinions of others to be taken into consideration in deciding what decisions or steps to take [2–5]. Problems arise when We make and process information from various opinion sites. The number of opinion sites available, makes it a difficult task to identify which sites are relevant, which contain information that We can process. We need a sentiment analysis system that can automatically identify and process information such as learning techniques [2,6].

One area in machine learning is structured learning which is commonly known as deep learning and hierarchical learning. For each problem and learning technology there are various deep learning architectures that can be used [7,8]. Each architecture is built on a network that is different from the function of each. One of the architectures is recurrent neural network (RNN). We used the RNN algorithm to calculate the dependence of each sentence on NLP. NLP has a temporal aspect where the word in the sentence has a dependence on the word before and after it. This research used RNN to reduce the dependence of each word in a sentence. RNN associates each word in the order of input sentences with a certain time step. Each number of time steps will be equal to the maximum sequence length of each word. Along with the time step there is a new component called hidden vector. Hidden vector summarizes all information in the previous time steps. Traditionally, a simple strategy for sequencing modelling is to map the order of inputs to fixed-size vectors using one RNN [9]. We also implemented Long Short-Term module (LSTM). This module is useful for overcoming the problem of growth of vector gradient components that potentially exist in the RNN algorithm during the training period [10]. Gradient vector growth has caused the RNN network to experience difficulties in studying sequences in sequence [11]. In analytical sentiments there are challenges faced when using deep learning that is modelling representations of words or sentences, LSTM has a superior performance in handling these challenges [12].

The purpose of this research is to evaluate and apply machine learning models. Models will be trained using user reviews dataset from Traveloka website. The model can be used in classifying or analysing sentiments in rankings of reviews as positive or negative to help automate and speed up the process of getting a general review of a product or service. This model will be implemented with the Recurrent Neural Network (RNN) architecture and Long Short-Term Module.

2. Literature Review

There are several previous studies regarding sentiment analysis using algorithms such as Naive Bayes. Bingwei [13] examines the scalability of Naive Bayes classifier on big data to achieve fine-grain control from the analysis procedure in the movie review dataset. Alex Go [14] using three algorithms namely Naive

Bayes, maximum entropy, and support vector machine to classify the sentiment of twitter messages using distant supervision. Some research where scientists use deep learning and neural networks to analyse sentiments. But most of these studies use datasets in English [15,16], and some cases in Asian languages [17–20]. There is still very little previous research in the field to analyse sentiments in Indonesian text using learning techniques and Deep Learning.

Merin Thomas [21] apply a recurrent neural network to analyse sentiment from tweeters in southern Malayalan language. The results showed that the accuracy of the model made using the RNN-LSTM method was 80%. This accuracy value can still be improved by using a deeper dataset. What's interesting about this research is that this method can be implemented using other languages. The RNN-LSTM method is two models that are often used to analyse sentiment.

Fenna Miedema [22] in a study entitled Sentiment Analysis with Long Short-Term Memory Networks conducted a study to find out why the RNN-LSTM model works well for analysing sentiments and how these models perform. LSTM is used to classify sentiments with a movie review dataset. The results show that the model can correctly classify 86.74% of the total reviews into the validation set. This model is very sensitive to overfitting but provides good results even without setting parameters.

Xin Wang and Yuanchao Liu [23] state that traditional RNN is not strong enough to handle complex sentiment expressions. Therefore, LSTM is implemented to classify sentiments. Experiment was done with a corpus of tweeters, a dataset containing 800,000 tweets labelled positive and negative. The results show that the LSTM network outperforms all other methods including SVM and Naïve Bayes.

3. Methodology

Figure 1 show Our research framework. Our model uses word2vec to convert the words in the corpus into a word vector. Words vector function as input in the model. Word2vec is a two-layer neural network that processes words in the corpus into a collection of vectors so that the inner network can understand them. Our model uses a deep learning algorithm to classify reviews from users. Deep learning requires input in numerical form. In this case, We use data in the form of text as input into the model, to overcome this We use word2vec to convert text or sentences into vector.

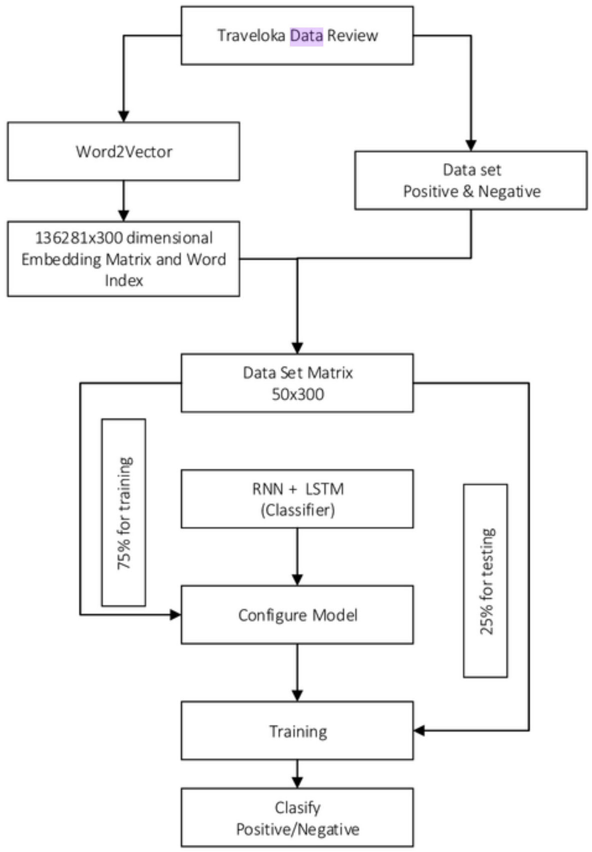

Figure 1. Research Framework

In addition to creating word vectors, word2vec will group vectors from words that have definitions and uses in the same context into vector space figure 2. Word2vec will detect every word vector mathematically. Words that have the same distance will be close to each other in vector space. Representation of word vectors is called embedding words. We can make fairly accurate guesses about the meaning of the word with enough data and context word. this can improve the performance of the recurrent neural network algorithm that We use in the model.

Figure 2. Same Context of Good Keyword

3.1. Word2Vec

Google is a company that develops word2vec. First made two algorithms called skip gram models and bag of word [24]. Word2vec will work optimally in guessing words very accurately if there are sufficient data, the usefulness and context of each word. Each guess can be implemented to build associations between words. For example, Rome, Paris and Beijing have the same distance in the vector space with the country whose capital is Rome - Italy = Beijing - China. When We only know that Rome is the capital of Italy while the capital of China still became question, then the formula above will return Beijing as the capital city of China.

We trained the word2vec model using 1,8 million review data from the Traveloka website with 138 thousand words. Words that are often used in the same context will be in the same group in vector space. Grouping words by context will help our model to reduce error rates. The model will produce a word vector, each word vector represents the word and its context, besides it also represents the word meaning and semantics [25]. Word2vec cluster positive words like *kamar*, *kamarnya*, room, and *ruangan* into one group and negative words into one group table 1. RNN requires this cluster for starting point of training.

Table 1. Words Vector

Positive	**Word**	*dikamar*	*kamarnya*	room	*Ruangan*
	Vector	0.4352757632 7323914	0.399477332830 4291	0.35348874 33052063	0.37247163057 32727
Negative	**Word**	*Kotor*	*Jelek*	*Dekil*	*Kumuh*
	Vector	0.5278632044 792175	0.415385812520 98083	0.40351858 735084534	0.36072847247 12372

We have 1.8 review data with 136280 words used to train the word2vec model. From that process the model produces a matrix that has 136,281 words vectors and each word vector has a dimension of 300. We will create two different data structures for the training process. First, We will make 136,281 word vectors to list in Python and. Second, the structure will make 136,281 x 300 dimensions embedding matrices which are used as a place to load all values of word vectors. The word2vec model produces an embedding matrix, which contains vectors of each word that is trained.

The next step is to make a vector representation of the input sentence. We utilize the function of Tensorflow embedding to get word vectors. Embedding tensor has a function to retrieve embedding matrix or matrix word vector. Besides the matrix word vector function this is also useful for retrieving the index of each word in the sentence. The index of each word also represents the index of the word vector. Index vector is basically a row index of each word in the input sentence. Table 2 show example index word vector for sentence "*pagi pagi sarapan nasi goreng di Kasur sambil nonton renang*".

Table 2. Index words vector

sentence	*Pagi pagi sarapan nasi goreng di kasur sambil nonton renang*									
word	*Pagi*	*Pagi*	*Sarapan*	*Nasi*	*Goreng*	*Di*	*Kasur*	*Sambil*	*Nonton*	*Renang*
index	548	548	130	225	226	19	461	1068	2669	49

3.2. Recurrent Neural Network

This study aims to classify sentiment analysis by applying a recurrent neural network algorithm. Recurrent neural network architecture is able to overcome temporal aspects in NLP. The temporal aspect is where the word in the sentence depends on the word that is before or after it. In the RNN structure, each word in the sentence will be connected to a certain time in sequence according to the input. Therefore, that the number of steps has the same length with the maximum length of the word sequence in the sentence figure 3.

Figure 3. Number of Time Step

RNN apply 2 input sources for the process, current input and past inputs. Figure 4 explain that BTSXPE is the input at the moment and CONTEXT UNIT is the output of the previous moment or called input in the past. The previous output moment in time step or t-1 will have an impact on the current moment's output at the time step or t [26].

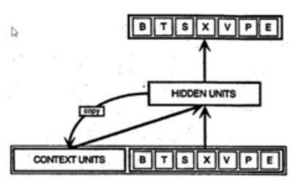

Figure 4. Input RNN

Both inputs on RNN are collected as a new input layer to calculate or determine how to process new data. The new input layer contains all information about all the words that exist in the current input and past money is used to detect the relationship between words in sequence [11]. The RNN maintains this sequential information in a hidden state. This process can achieve many time steps to give effect to the new process. Formulated into the mathematical formula (1), where ht hidden state is a function of the current input, multiplied by the weight matrix w, then added input to the past ht-1 multiplied by the transition matrix u. Weight matrices have an important role in RNN, this matrix will determine adjustments to current input and previous hidden conditions. Errors that occur will be reprocessed by backpropagation until the lowest error rate is reached.

$$h_t = \sigma(w^H\, h_{t-1} + w^x\, x_t) \qquad (1)$$

To update the weight matrix, We use Adam optimization to process backpropagation through time. The next process is to enter the hidden state vector in the last time step into the Binary Softmax classifier which functions to produce values of 0 and 1, or in this research provides the possibility of positive values and negative values figure 5. We implemented Tensorflow with Cuda for training model. Device to support the training process using two GPUs from Invidia 1070 ti. For the deep learning process, We use KERAS.

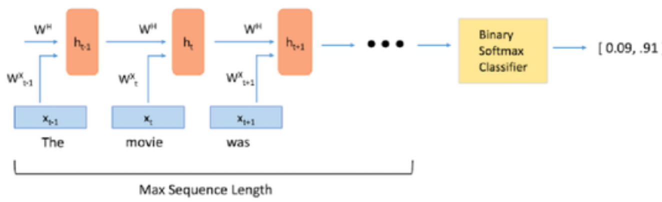

Figure 5. Binary Softmax Classifier

3.3. Long Short-Term Memory Units (LSTM)

Long Short-Term Memory is a module in an RNN network that has a function to overcome missing gradient problems. RNN apply the LSTM network to keep from propagation errors. This enabling the RNN to continue learning through many steps of time. LSTM contains cells that store information outside a recurring network. The cell is like memory in their computer that will decide when that data must be stored, written, read or deleted through the gate figure 6. There are four gates that LSTM use, an input gate, a forget gate, an output gate, and a new memory container [25].

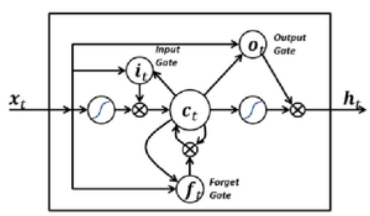

Figure 6. LSTM Gate

Our research uses several hyper tunings that is used to improve the performance of RNN and LSTM effectively. The first is the learning rate in this study starting from 0.001. RNN uses a learning rate to maintain fluctuations and training processes that remain stable. Second is optimizer, We apply Adam optimizer because it has properties with adaptive learning levels. Third is the LSTM unit, We use 64 LSTM units for this model training. the number of units depends on the average length of each review. Fourth is the size of the word vector, from the previous embedding matrix process We get the size of the word vector is 136281 x 300

4. Results and Discussion

4.1. Data Set

We use Traveloka travel web and mobile application review dataset. The dataset consists of twenty-five thousand dataset which is divided into two classes positive and negative at 12,500 for each. The dataset is stored in a CSV file. Table 4 show review dataset.

Table 3. Review Data Set

	Positive review	Negative review
Total File	12.500	12.500
Total words	2.989.896	3.353.227
Average words	28.99	33.53

Figure 7 is a histogram of the sentence length distribution of the review dataset. We use Matplot library to visualize this data. We apply the visualization histogram to know the average number of words each review. We use the average word value as the value of the max sequence length of the model. Figure 7 shows the distribution of sentence length, the length of sentence distributed between 20 to 140 words. In this model We limits the sentence length at 50 words.

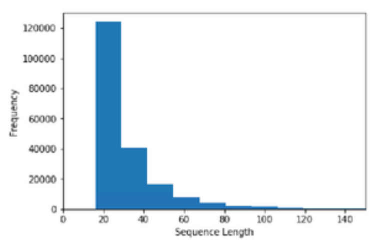

Figure 7. Histogram Review Data Set

4.2. Experimental Data set matrix

Our model used some hyperparameters to improve performance the model table 6. In this work We need to specify placeholders. We create two placeholders, which will be used as input into the network and the others will be used as labels.

Table 4. Hyperparameters

No	Hyperparameter	Total
1	Batch size	24
2	LSTM unit	64
3	Number of classes	2
4	Number of training iteration	10000

Labels placeholders contain a set of 1.0 values representing a positive review dataset and 0.1 representing a negative review dataset. Placeholders have rows that represent the input of each training dataset. Model also uses placeholders as a place to hold input data as input into the system figure 8.

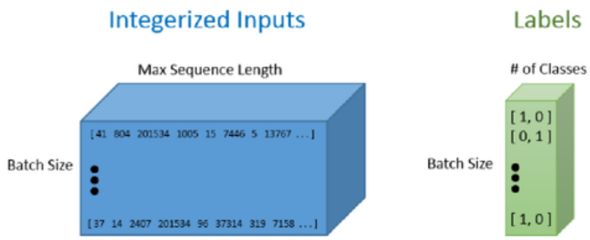

Figure 8. Placeholder on LSTM

With placeholders, the model runs the TensorFlow lookup function to capture word vectors generated by word2vec in the previous process. With the functions, word vectors or so-called embedding matrices together with placeholders will return the 3-D batch tensor dimensions with the maximum dimension sequence length of the embedding matrix figure 9. Using the 3-D test makes it easier for us to see input data in an integer form on TensorFlow.

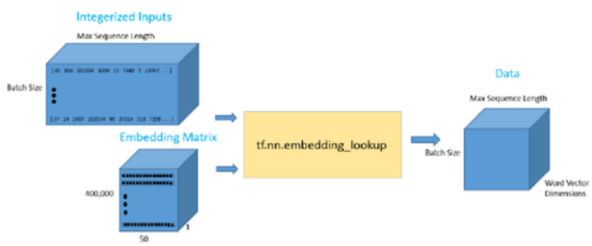

Figure 9. 3-D Dimension

The results of the above process are data in the form of integers, the data as input into the LSTM network. The next process is to take the integer data and enter it into the LSTM unit that We use with a function in Jupiter. After the LSTM is filled with data, then the system will wrap the cell in the dropout layer. This function is to keep the cell from overfitting. The selection of LSTM network architecture is to utilize hidden state vectors. This process is done by stacking lots of LSTM cells. It can help the model maintain information from long-term dependence, otherwise it can be used to provide many parameters to the model. So in the LSTM network it usually takes a little longer for the learning process and more training examples.

The first output of the dynamic RNN function can be thought of as the last hidden state vector. This vector will be reshaped and then multiplied by a final weight matrix and a bias term to obtain the final output values [25]. Then the model will check whether the prediction formulation can work correctly using the maximum value index of all two output values matched with the value of the training data set label.

4.3. Result and analysis

The first step is defining Tensorflow and loading the reviews and labels. Second step is running function in the training session. This function has two arguments namely fetches argument and feed_dict argument. The fetches argument serves to define the value of the computational process and the feed_dict argument functions to enter data. Both argument are places for floating all input from placeholders. Input in the form of data structure serves as input to batch reviews and batch labels. Then the training process will repeat according to the number of iterations that have been set. The iterations in this training We set 10000 iterations. From the training set We get an accuracy of 0.9499% and the loss value is close to zero%.

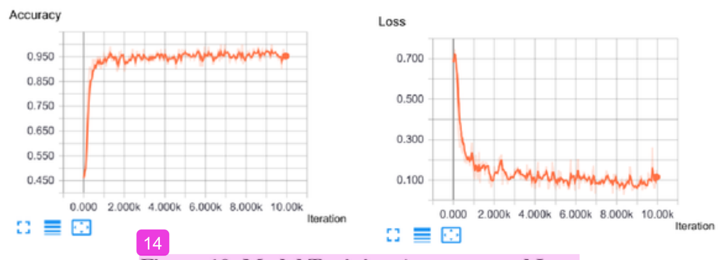

Figure 10. Model Training Accuracy and Loss

Figure 10 show the model works well. With losses decreasing closer to zero value and performance increasing close to 100 percent. However, We must monitor for possible overfitting during training. Overfitting is actually a common phenomenon in machine learning [27]. This phenomenon occurs when the model becomes very compatible with the dataset so that it produces losses that reach zero. Therefore, in the training session We stop training before the loss value reaches zero to anticipate overfitting. We stopped training with intuition techniques. As a comparison, We also conduct training using other machine learning algorithms such as CNN, Naive Bayes, and conventional RNN. Evaluation result of each algorithm can be seen in table 7. These results show that RNN with LSTM achieve better accuracy than the other model.

Table 5. Result Comparison

Model	accuracy
CNN + Word2Vec	0.91323
Naïve Bayes	0.441
RNN Conv	0.8977
RNN + LSTM	0.9499

5. Conclusions

In this research, We proposed a model for sentiment analysis using RNN and LSTM for Indonesian language dataset. According to our experiments, RNN+LSTM achieves 94.99% of accuracy, overperforms than CNN+word2vec, naïve Bayes and RNN Conv. Indonesian corpus datasets consists of twenty-five thousand sentences and has been human labelled. RNN in this model is implemented using TensorFlow framework. We use TensorFlow to assist the model in classifying sentiments and for understanding the natural language of the world. Today, Scientists have proposed grid LSTM with also showed great performance in some application. In the future We could try grid LSTM to improve the performance of sentiment classifier.

References

1. Rezaeinia, S.M.; Rahmani, R.; Ghodsi, A.; and Veisi, H. (2019). Sentiment analysis based on improved pre-trained word embeddings. *Expert Systems with Applications*, 117, 139–147.

2. Zhang, L.; Wang, S.; and Liu, B. (2018). Deep Learning for Sentiment Analysis : {A} Survey. *CoRR*, abs/1801.0, 32.

3. Appel, O.; Chiclana, F.; Carter, J.; and Fujita, H. (2017). A Consensus Approach to the Sentiment Analysis Problem Driven by Support-Based IOWA Majority. *International Journal of Intelligent Systems*, 32, 947–965.

4. Appel, O.; Chiclana, F.; Carter, J.; and Fujita, H. (2017). Cross-ratio uninorms as an effective aggregation mechanism in sentiment analysis. *Knowledge-Based Systems*, 124, 16–22.

5. Appel, O.; Chiclana, F.; Carter, J.; and Fujita, H. (2018). Successes and challenges in developing a hybrid approach to sentiment analysis. *Applied Intelligence*, 48, 1176–1188.

6. Cambria, E. (2016). Affective Computing and Sentiment Analysis. *IEEE Intelligent Systems*, 31, 102–107.

7. Song, M.; Park, H.; and Shin, K. shik. (2019). Attention-based long short-term memory network using sentiment lexicon embedding for aspect-level sentiment analysis in Korean. *Information Processing and Management*, 56, 637–653.

8. Dosoula, N.; Griep, R;. and Den Ridder, R. (2016). Sentiment analysis of multiple implicit features per sentence in consumer review data. *Frontiers in Artificial Intelligence and Applications*, 291, 241–254.

9. Cho, K.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* 1724–1734.

10. Hochreiter, S.; and Frasconi, P. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. *Learning* 401–403.

11. Hochreiter, S.; and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9, 1735–1780.

12. Park, H.; Song, M.; and Shin, K.-S. (2019). Deep learning models and datasets for aspect term sentiment classification: Implementing holistic recurrent attention on target-dependent memories. *Knowledge-Based Systems.*

13. Liu, B.; Blasch, E.; Chen, Y.; Shen, D.; and Chen, G. (2013). Scalable sentiment classification for Big Data analysis using Naïve Bayes Classifier. *Proceedings - 2013 IEEE International Conference on Big Data, Big Data*

2013, 99–104.

14. Go, A.; Bhayani, R.; and Huang, L. (2009). Twitter Sentiment Classification using Distant Supervision. *CS224N Project Report*,

15. Shirani-Mehr, H. (2015). *Applications of Deep Learning to Sentiment Analysis of Movie Reviews*.

16. Timmaraju, A.; and Khanna, V. (2015). Sentiment Analysis on Movie Reviews using Recursive and Recurrent Neural Network Architectures. *Semantic Scholar*, 2–7.

17. Al-smadi, M.; Al-ayyoub, M.; Jararweh, Y.; and Qawasmeh, O. (2019). Enhancing Aspect-Based Sentiment Analysis of Arabic Hotels ' reviews using morphological , syntactic and semantic features. *Information Processing and Management*, 56, 308–319.

18. Zhang, L.; and Chen, C. (IEEE, 2016). Sentiment Classification with Convolutional Neural Networks: An Experimental Study on a Large-Scale Chinese Conversation Corpus. in *2016 12th International Conference on Computational Intelligence and Security (CIS)* 165–169.

19. Heikal, M.; Torki, M.; and El-Makky, N. (in Elsevier 2018). Sentiment Analysis of Arabic Tweets using Deep Learning. in *Procedia Computer Science* 142, 114–122.

20. Pasupa, K.; and Seneewong Na Ayutthaya, T. (2019). Thai sentiment analysis with deep learning techniques: A comparative study based on word embedding, POS-tag, and sentic features. *Sustainable Cities and Society*, 50, 101615.

21. Thomas, M.; and C.A, L. (2018). Sentimental analysis using recurrent neural network. *International Journal of Engineering & Technology*, 7, 88.

22. Miedema, F. (2018). *S*entiment Analysis with Long Short-Term Memory networks. Retrieved April, 2019, from https://www.semanticscholar.org

23. Wang, X.; Liu, Y.; SUN, C.; Wang, B.; and Wang, X. (Association for Computational Linguistics, 2015). Predicting Polarities of Tweets by Composing Word Embeddings with Long Short-Term Memory. in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* 1343–1353.

24. Rong, X. (2014). *Word2Vec Parameter Learning Explained.* arXiv, Cornell University.

25. Deshpande, A. (2018). Perform sentiment analysis with LSTMs. Retrieved 20 April, 2019, from https://www.oreilly.com/learning/perform-sentiment-analysis-with-lstms-using-tensorflow.

SENTIMENT ANALYSIS USING RECURRENT NEURAL NETWORK AND LSTM IN BAHASA INDONESIA

ORIGINALITY REPORT

14%	9%	5%	12%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.oreilly.com Internet Source	3%
2	Submitted to Universiti Teknologi Petronas Student Paper	1%
3	Submitted to Universiti Teknologi MARA Student Paper	1%
4	jestec.taylors.edu.my Internet Source	1%
5	Submitted to University of Melbourne Student Paper	1%
6	Submitted to Universiti Teknikal Malaysia Melaka Student Paper	1%
7	Lei Zhang, Shuai Wang, Bing Liu. "Deep learning for sentiment analysis: A survey", Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2018 Publication	1%

8	Submitted to Trinity College Dublin Student Paper	<1%
9	Bingwei Liu, Erik Blasch, Yu Chen, Dan Shen, Genshe Chen. "Scalable sentiment classification for Big Data analysis using Naïve Bayes Classifier", 2013 IEEE International Conference on Big Data, 2013 Publication	<1%
10	"Natural Language Processing and Chinese Computing", Springer Science and Business Media LLC, 2018 Publication	<1%
11	Submitted to University of Hull Student Paper	<1%
12	"Web, Artificial Intelligence and Network Applications", Springer Science and Business Media LLC, 2019 Publication	<1%
13	link.springer.com Internet Source	<1%
14	Submitted to University of Strathclyde Student Paper	<1%
15	Seyed Mahdi Rezaeinia, Rouhollah Rahmani, Ali Ghodsi, Hadi Veisi. "Sentiment analysis based on improved pre-trained word	<1%

embeddings", Expert Systems with Applications, 2019
Publication

16	Submitted to Universiti Tunku Abdul Rahman Student Paper	<1%
17	sentic.net Internet Source	<1%
18	Submitted to Heriot-Watt University Student Paper	<1%
19	Submitted to City University Student Paper	<1%
20	Submitted to City University of Hong Kong Student Paper	<1%
21	Submitted to Jawaharlal Nehru University (JNU) Student Paper	<1%
22	Submitted to iGroup Student Paper	<1%
23	Wei Xia, Wen Zhu, Bo Liao, Min Chen, Lijun Cai, Lei Huang. "Novel architecture for long short-term memory used in question classification", Neurocomputing, 2018 Publication	<1%
24	Submitted to University of Southampton Student Paper	<1%

25	Submitted to Universidade Nova De Lisboa Student Paper	<1%
26	Submitted to Jawaharlal Nehru Technological University Student Paper	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		