

Effect of Sentence Length in Sentiment Analysis Using Support Vector Machine and Convolutional Neural Network Method

Agung Pambudi*¹, Suprpto²

¹Master Program in Computer Science, FMIPA UGM, Yogyakarta, Indonesia

²Department of Computer Science and Electronics, FMIPA UGM, Yogyakarta, Indonesia

e-mail: *¹agungpambd@mail.ugm.ac.id, ²sprpto@ugm.ac.id

Abstrak

Berdasarkan Pasal 10 ayat 1 UU Nomor 14 tahun 2005, seorang pengajar harus memiliki empat kompetensi: pedagogik, kepribadian, sosial, dan profesional. Training ICT di Universitas Islam Negeri Sunan Kalijaga yang melibatkan instruktur sebagai tenaga pendidik harus memiliki kompetensi tersebut. Kinerja seorang instruktur dinilai melalui sebuah sistem yang dilakukan oleh mahasiswa dengan memberikan komentar terhadap intruktur. Komentar tersebut mengandung sentimen positif dan negatif yang dapat dikaji dengan melakukan analisis sentimen. Penelitian terkait analisis sentimen beberapa tahun terakhir sudah banyak dilakukan, tetapi jarang yang memperhatikan pengaruh panjang kalimat dari dataset terhadap performansi metode yang digunakan. Penelitian ini melakukan analisis sentimen terkait pengaruh panjang kalimat pada komentar mahasiswa training ICT terhadap metode Support Vector Machine dan Convolutional Neural Network.

Dari penelitian ini diperoleh kesimpulan bahwa panjang kalimat pada dataset akan mempengaruhi performansi metode SVM dan CNN jika dikombinasikan dengan Word2vec. Sedangkan untuk metode SVM+TFIDF, performansi tidak begitu dipengaruhi oleh panjang kalimat, meski demikian kombinasi metode ini memiliki waktu proses yang paling cepat jika dibandingkan metode lainnya. Adapun metode CNN+Word2vec menghasilkan performansi terbaik pada penelitian ini dengan nilai akurasi sebesar 0,94%, presisi sebesar 0,95%, recall sebesar 0,96%, dan f1-score sebesar 0,95%.

Kata kunci— Analisis Sentimen, SVM, CNN

Abstract

Based on Article 10 paragraph 1 of Law No. 14 of 2005, a teacher must have four competencies: pedagogical, personality, social, and professional. ICT training at Sunan Kalijaga State Islamic University involves instructors as educators who must have such competencies. An instructor's performance is assessed through students' learning evaluation system by giving comments to the instructions. These comments contain positive and negative sentiments that can be reviewed by conducting sentiment analysis. Research related to sentiment analysis in recent years has been widely done, but researchers rarely pay attention to the effect of sentence length from the dataset on the method's performance. This study tried to analyze sentiment related to sentence length effect on ICT training student comments using Support Vector Machine and Convolutional Neural Network methods.

This study concluded that the sentence length on the dataset would affect the SVM and CNN methods' performance when combined with Word2vec. While the SVM+TFIDF method performance is not affected by sentence length, this method has the fastest process time than other methods. The CNN+Word2vec method produced the best performance in this study with a value of 0.94% accuracy, 0.95% precision, 0.96% recall, and 0.95% f1-score.

Keywords— Sentiment analysis, SVM, CNN

1. INTRODUCTION

Education is one of the most important aspects of every individual also supports the nation's progress. Therefore, every educational institution always makes improvements for a better level and quality, especially in human resources development, related to the educators' quality itself. According to Act of The Republic Indonesian article 10 verse (1) Number 14 of 2005, a teacher must have four competencies: pedagogical competencies, personality competencies, social competencies, and professional competencies. These professional competencies include subject knowledge, pedagogical competencies, also active teaching and learning methods. Meanwhile, personality competencies include a high commitment to learners' duties and success, and then social competencies include a good way of communication with learners.

Information and communication technology (ICT) training, which is compulsory for new students at Sunan Kalijaga State Islamic University, is one example that requires an educator or instructor who must have professional competencies. This instructor's level of professionalism can be assessed from student evaluation results opinions towards the instructor itself. Student assessments of opinions and comments on the instructor's performance are stored in a learning evaluation system. In addition to containing several questions with multiple choice answers, in this learning evaluation system, there are also comments to represent some student complaints about instructors' that are directly related to how the instructors teach and the learning process that occurs. These students' comments contain positive and negative sentiment values that can be used to assess ICT training instructors' performance.

Sentiment analysis is a branch of research in text mining. Text mining is the application of data mining concepts and techniques to look for patterns in text. Sentiment analysis is a process to understand automatically, extract, and process textual data to obtain information in the form of sentiments contained in an opinion sentence. Sentiment analysis is carried out to see the opinion or tendency of a person's opinion on a problem or object, determining whether the opinion is positive or negative [1]. One method often used in research with sentiment analysis is Support Vector Machine (SVM), which is part of the machine learning method. The advantages of the Support Vector Machine method itself are that it can produce higher accuracy values, faster classification speed, and a high tolerance for irrelevant attributes [2].

In addition to these methods, deep learning methods can also solve the sentiment analysis problem nowadays. Deep learning is a relatively new branch of machine learning. The advantage of the deep learning method is that it has higher accuracy and can be trained and consistent when faced with large amounts of data [3]. One example of a method in deep learning is Convolutional Neural Network (CNN). This method works well for image analysis and image classification because it can extract an area of features from global information, and it is able to consider the relationship among these features. Meanwhile, in sentiment analysis, CNN has a convolutional layer to extract information by a larger text piece. CNN also requires fewer connections and parameters making it easier to train. Besides, the CNN method that combines with the word embedding feature generated from Word2vec can improve the classification performance [4].

Research related to sentiment analysis has been done so far. As an example of using the SVM method, Ilmawan [5] developed the application with the Naïve Bayes classification algorithm and compared it with the Linear SVM method to test the accuracy of an application's comment on Google Play. The study results stated that the accuracy rate of Linear SVM is 89.49%, higher than Naïve Bayes' accuracy rate that only 83.87%.

With the same algorithm, Kharisman [6] do researches about sentiment analysis on airline reviews using a combination of lexicon-based classification methods with supervised learning. Lexicon-based classification methods are used to create training data and to classify document sentiments using the Naïve Bayes Classifier and Support Vector Machine methods. The test results showed that the Support Vector Machine method has an accuracy of 91.00%, while the Naïve Bayes Classifier's accuracy is only 89.50%.

In the deep learning method, Kim [7] introduced the CNN method's application in the classification of a sentence. The study introduced a new model to apply to NLP. Sentences and words as input are first created into vector form using Word2vec. In his research, Kim concluded that adjusting the hyperparameter on a simple CNN with a single layer of convolution can produce a good performance.

Ouyang et al. [4] conduct a sentiment analysis using CNN. In his research, CNN has fewer connections and is easier to train than other deep learning methods. Also, CNN can be applied in the field of sentiment analysis. By using CNN, we only need to label sentences artificially. This study applies the CNN network to solve the problem. Technically, by compiling a framework flow consisting of Word2vec + CNN on the dataset. Pre-training + fine-tuning paths are used to train CNN to perform NLP tasks. This framework was chosen because it felt better than other learning models, such as RNN and Matrix-Vector Recursive. The experimental results show that the CNN that has been trained can produce better classifications than conventional algorithms.

Razi [8] do researches to group Indonesian news articles using five classes, namely Entertainment, Health, Sports, Technology, and Economics classes, by implementing the CNN method. Before classification, the words have converted into vector form using Word2vec so that the results of the conversion can be input into CNN. Test results on the built system showed that the combination of CNN and Word2vec methods gave better accuracy results than the Naïve Bayes method, with an accuracy value of 96.70%.

Although research related to sentiment analysis has been widely done so far, most researchers do not pay attention to the effect of sentence length from the dataset used on the process and performance analysis of the classification method used, especially for datasets that use the Indonesian language. In this research, we try to do a sentiment analysis process of the Indonesian language dataset about student opinion for instructors' professionalism by analyzing the effect of sentence length and feature extraction used in the Support Vector Machine and Convolutional Neural Network methods.

2. METHODS

This study uses dataset on learning evaluation system that contains positive and negative sentiment. The stages in this research are preprocessing, feature extraction, classification process using SVM and CNN algorithm, and then evaluate the classifier model using K-fold cross validation. The steps of the study can be seen in Figure 1.

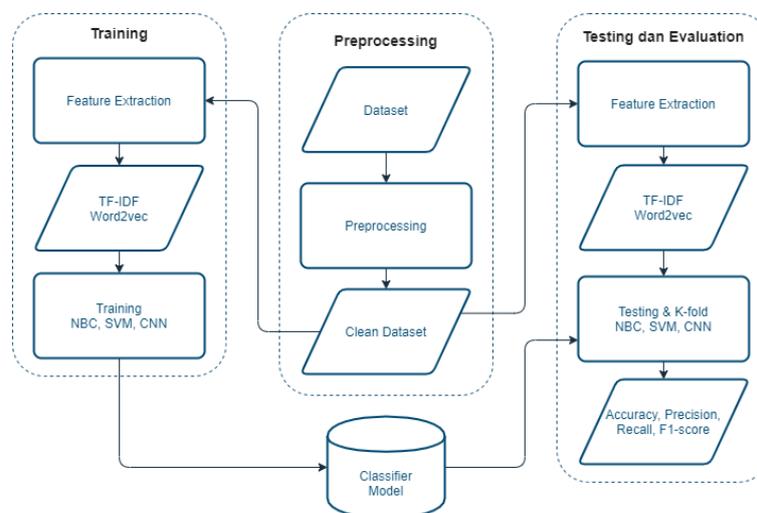


Figure 1 System Architecture

2.1 Preprocessing

Preprocessing is when the text to be classified is cleaned and prepared before the text is analyzed [9]. Preprocessing of data was carried out to avoid imperfect data, disruption of data, and inconsistent data [10]. This step begins by tokenizing the text, removing stop word, and the last step is the stemming process.

2.2 Feature Extraction

Feature extraction is an extraction process to identify the entities in question [11] or the process of extracting a new set of features from the original feature through functional mapping [12]. The features used in this study are TF-IDF (Term Frequency-Inverse Document Frequency) and word embedding using Word2vec. The TF-IDF weight value calculation is obtained from scikit-learn TfidfVectorizer. This python library helps us transform the text into a sparse matrix of n-gram counts and then performs the TF-IDF transformation from a provided matrix of counts.

While for Word2vec, we apply the gensim, which is a python library, to help us implement the Word2Vec. In this process, we first build a vocabulary from the entire training data. To generate the word vectors well, we employ the Skip-gram model because it has a better learning ability than CBOW. After training, each word attaches a vector. Finally, we construct a high dimension matrix. Each row in the matrix represents every training example, and the columns are the generated word vectors. Consequently, the word has multiple degrees of similarity, and it can be computed via a linear calculation.

2.3 Support Vector Machine

Support Vector Machine (SVM) was first introduced by Vapnik [13]. The learning process in SVM aims to obtain a hypothesis in the form of the best separation area that minimizes not only empirical risk, namely the average error in training data, but also has good generalizations. To guarantee this generalization, SVM works based on the principles of Structural Risk Minimization (SRM). SVM is a technique that is relatively new compared to other techniques. However, it has better performance in various application fields such as bioinformatics, handwriting recognition, text classification, and so on.

The best dividing field search on the SVM algorithm can be formulated as follows:

$$\begin{aligned} \max_{\alpha} L_D(\alpha) &= \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \alpha_i \alpha_j y_i y_j x_i x_j \\ \text{s. t. } \sum_{i=1}^n \alpha_i y_i &= 0, \alpha_i \geq 0. \end{aligned} \quad (1)$$

With that formula, it will obtain α_i value that can be used to find w . There is α value for each training data. Training data with a value of $\alpha_i > 0$ is a support vector, while the rest has a value of $\alpha_i = 0$. Therefore, the result decision function is only influenced by the support vector.

The formula for finding the best hyperplane is a quadratic programming problem so that the maximum global value of α_i can always be found. After the quadratic programming problem solution is done (value of α_i), the class of the x test data can be determined based on the value of the decision function:

$$f(x_d) = \sum_{i=1}^{ns} \alpha_i y_i x_i x_d + b \quad (2)$$

Where x_i is a support vector, ns is the total support vector, and x_d is data that will be classified.

Kernel Trick

The SVM algorithm uses a kernel trick to solve data that cannot be separated linearly. The data will be mapped using the mapping (transformation) function $x_k \rightarrow \phi(x_k)$ into the feature space so that there is a separator field that can separate the data according to its class. In practice, feature spaces usually have a higher dimension than the input vector (input space), this will result in the computation of feature spaces being too large because there is a possibility that feature spaces can have an infinite number of features. Also, it is not easy to know the exact transformation function. In the training process, the resulting functions are:

$$f(x_d) = \sum_{i=1}^{ns} \alpha_i y_i k(x_i, x_d) + b \quad (3)$$

$x_i = \text{support vector.}$

In equation 5, ns represents the number of support vectors. SVM is also often referred to as sparse kernel machines. This is because SVM only needs to compute several support vectors in kernel functions; not all of them need to be computed like other kernel methods [14]. Some of the commonly used kernel functions are as follows:

1. Linear Kernel

$$K(x, x_i) = x_i^T x \quad (4)$$

2. Polynomial Kernel

$$K(x, x_i) = [(x, x_i) + C]^d \quad (5)$$

3. Radias Basis Function (RBF)

$$K(x, x_i) = \exp(-\gamma \|x - x_i\|^2) \quad (6)$$

4. Sigmoid Kernel

$$K(x, x_i) = \tanh(x_i^T x + r) \quad (7)$$

2.4 Convolutional Neural Network

In our CNN model, we use a simple CNN architecture described by Kim [7]. It consists of a convolutional layer, a max pooling layer, a dropout layer and a fully connected output layer as shown as in Figure 2. Each of these layers is explained in turn.

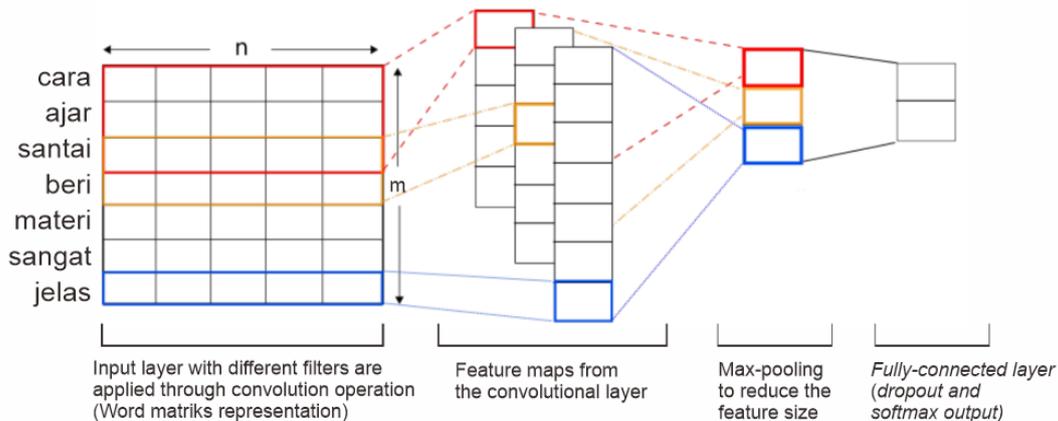


Figure 2 CNN architecture

Input Layer. The inputs of the CNN classifier are preprocessed datasets that consist of a sequence of words. Using Word2vec word embedding, datasets are converted into vector representations in the following way. Assuming $w_i \in \mathbb{R}^n$ to be the n -dimensional word embeddings vector of the i th word in a dataset, a word matrix representation is obtained by looking up the word embeddings and concatenating the corresponding word embeddings vectors of the total m words:

$$DCNN = w_1 \oplus w_2 \oplus \dots \oplus w_k \quad (8)$$

where \oplus denotes the concatenation operation [7]. For training purposes, short datasets are padded to the longest dataset's length using a special token. Hence the total dimension of the vector representation DCNN is always $m \times n$. Afterward, the word matrix representation will feed to the convolutional layer.

Convolutional layer. The convolution operation helps the network learn important words no matter where they appear in a dataset [15]. In this layer, the filter $F_i \in \mathbb{R}^{m \times n}$ with different sizes of m are applied to the word matrix representation DCNN. By varying the stride s [16], we can shift the filters across s word embeddings vectors at each step. By sliding the filters over m word vectors in DCNN using stride s , the convolution operation produces a new feature map c_i for all the possible words in a dataset:

$$c_i = f(F_i \cdot DCNN_{ii+m-1} + b_i) \quad (9)$$

where $i:i+m-1$ denotes the word vectors of the word i to word $i+m-1$ in DCNN. b_i is the corresponding bias term that is initialized to zero and learned for each filter F_i during training. In Equation (x2), f is the activation function. In this CNN architecture, we used a rectified linear function (ReLU) as f . Whether the input x is positive or negative, the ReLU unit ensures its output is always positive as defined by $f = \max(0, x)$.

Max pooling layer. All the feature maps c_i from the convolutional layer are then applied to the max-pooling layer where the maximum value c_i^{max} is extracted from the corresponding feature map. Afterward, the maximum values of all the feature maps c_i are concatenated as the feature vector of a dataset [17].

Dropout layer. Dropout is a regularization technique that only keeps a neuron active with some probability p during training [7]. After training, $p = 1$ is used to keep all the neurons active for predicting unseen tweets. With the L2 regularization, it constraints the learning process of the neural networks by reducing the number of active neurons.

Softmax Layer. The dropout layer outputs are fed into the fully connected softmax layer, transforming the output scores into normalized class probabilities [7]. Using a cross-entropy cost function, the ground truth labels from human assessors are used to train the CNN classifier for our dataset classification task.

3. RESULTS AND DISCUSSION

The number of datasets used in this study was 1.707 comments that divide into four categories based on the sentence length in the datasets, namely Data-1 contain 1-10 words in one comment with 902 dataset, Data-2 contain 5-15 word in one comment with total 1.219 dataset, Data-3 contain 10-20 word in one comment with total 811 dataset, Data-4 contain 1-10 words combined with 30-100 words in one comment with total 950 dataset, and Data-5 contain 1-100 words in one comment with 1.707 dataset. Dataset is divided into 80% of training data and 20% of test data, then it performed on each test process in the five categories of data. The result of the test process is a classifier performance that uses the K-fold cross validation approach with $K=10$.

The first test uses the Support Vector Machine algorithm with a linear kernel and the TF-IDF feature. Tests were carried out on all five dataset categories. Table 1 shows the results of

testing the SVM + TF-IDF method with the K-Fold Cross Validation approach with $K = 10$. Based on the test results on the SVM+TF-IDF method shown in Table 1 of the five dataset categories, the following conclusions are produced:

1. There is no significant difference in the effect of sentence length for testing on the SVM+TF-IDF algorithm, the increase and decrease in performance is only influenced by the number of datasets and variations of the existing terms.
2. The value of the 10-fold cross validation test results on the accuracy, precision, recall, and f1-score on the SVM+TF-IDF method tends to increase when the dataset has more unique term variations even though the test will take longer.

Because there is no impact on the sentence length of the five dataset categories used in the testing process for the SVM+TF-IDF method, from the 1707 datasets used, the SVM+TF-IDF algorithm can produce an accuracy value of 0.92%, with a precision of 0.93%, recall of 0.88%, and f1-score of 0.90%, and processed within 1.27 seconds on the architecture used.

Table 1 SVM+TF-IDF Test Result with 10-Fold Cross Validation

%	Data-1	Data-2	Data-3	Data-4	Data-5
Accuracy	0,91	0,90	0,90	0,91	0,92
Precision	0,91	0,89	0,89	0,93	0,93
Recall	0,85	0,85	0,80	0,87	0,88
F1-score	0,88	0,87	0,84	0,90	0,90
Time (s)	1,09	1,15	1,12	1,13	1,27

The second test was conducted to determine the effect of similar feature weighting that is Word2vec in two different disciplines study, which is machine learning (SVM) and deep learning (CNN) and also to find out whether sentence length on the dataset would have an impact on the test results. At this stage, testing was carried out by experimenting with three different dimensions of Word2vec which is 100, 200, and 300. Hyperparameters that use in this test are the filter size of (3,4,5), dropout value of 0.5, batch size of 50, and epoch of 20. Based on the test results on the SVM and CNN method with Word2vec, as shown in Table 2 of the five dataset categories produced these conclusions:

1. In the testing process, it can be seen that although using a different value of Word2vec dimensions, the SVM method tends to produce the same performance value for each vector dimension. However, it affects only the process time if the vector dimensions are said to be bigger. As for the effect of sentence length, it can be seen that the SVM+Word2vec algorithm will produce better performance if the sentence length has a limit between 10-20 words, both with word vector dimensions of 100, 200, and 300.
2. When compared with the previous test method which is SVM+TF-IDF, the SVM+Word2vec in this research case cannot produce a better performance.
3. For the CNN+Word2vec method, the performance value is affected by the sentence length and also the size of the word vector dimensions itself. This can be seen in the test with Data-2 which is limited to 5-10 words per sentence, and Data-3 which is limited to 10-20 words

per sentence. When tested with datasets that have too short sentence lengths such as Data-1, or datasets with random sentence lengths such as Data-4 and Data-5, the method will not produce better performance even though there are have more datasets.

4. If the CNN+Word2vec method is compared to the SVM+TF-IDF or SVM+Word2vec method, then the CNN+Word2vec algorithm is superior to the two for this research case, it is just that it requires a longer processing time than the other algorithms.

Table 2 SVM and CNN test Result Combined with Word2vec

	%	Data-1		Data-2		Data-3		Data-4		Data-5	
		SVM	CNN								
W2v 100	Accuracy	0,76	0,92	0,84	0,93	0,84	0,92	0,83	0,92	0,78	0,92
	Precision	0,70	0,92	0,86	0,94	0,84	0,92	0,85	0,92	0,70	0,94
	Recall	0,67	0,97	0,64	0,97	0,64	0,97	0,73	0,97	0,62	0,95
	F1-score	0,68	0,95	0,73	0,95	0,73	0,94	0,79	0,94	0,66	0,94
	Time (s)	1,34	4,94	1,43	6,36	1,28	6,69	1,28	6,10	1,59	9,01
W2v 200	Accuracy	0,76	0,93	0,84	0,94	0,84	0,94	0,83	0,92	0,79	0,92
	Precision	0,70	0,94	0,92	0,96	0,84	0,94	0,85	0,94	0,72	0,94
	Recall	0,67	0,97	0,65	0,97	0,64	0,99	0,73	0,96	0,62	0,95
	F1-score	0,68	0,95	0,76	0,96	0,73	0,97	0,79	0,95	0,67	0,95
	Time (s)	1,52	7,98	1,79	12,10	1,42	7,41	1,53	8,43	2,32	19,80
W2v 300	Accuracy	0,76	0,93	0,84	0,94	0,84	0,95	0,83	0,93	0,78	0,92
	Precision	0,70	0,94	0,92	0,96	0,84	0,95	0,85	0,94	0,70	0,94
	Recall	0,67	0,96	0,65	0,97	0,64	0,96	0,73	0,96	0,62	0,95
	F1-score	0,68	0,95	0,76	0,96	0,73	0,95	0,79	0,95	0,66	0,94
	Time (s)	1,86	11,50	2,15	18,50	1,65	11,20	1,70	12,90	3,07	22,30

Thus, for the use of Word2vec features, the CNN algorithm's performance is far superior when compared to SVM. The best performance of the 10-fold cross validation test results on the CNN+Word2vec algorithm is obtained on the hyperparameter with a word representation vector size of 300 in the Data-3 category, with details of the accuracy value of 0.94%, precision of 0.95%, recall of 0.96%, and an f1-score of 0.95%, and processed within 11.20 seconds on the architecture used.

4. CONCLUSIONS

From the research conducted and discussion of previous chapters, it can be concluded that the sentence length in the dataset can impact on the performance results of the tests carried out on the SVM and CNN algorithms if using Word2vec feature weighting. Whereas for the case of weighting using TF-IDF combined with the SVM algorithm, the effect of sentence length is not significant. However, the SVM+TF-IDF algorithm has a faster processing time when compared to other method combinations.

The CNN algorithm combined with Word2vec feature extraction and hyperparameter with a filter size of (3,4,5), a dropout of 0.5, a batch size of 50, an epoch of 20, and a vector dimension size of a word representation of 300 resulted in the best performance in this study. These performance values include accuracy of 0.94%, precision of 0.95%, recall of 0.96%, and f1-score of 0.95%.

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

1. This study is still limited to a small amount of data, which the future research should use large amounts of data.
2. Can be added category classifications based on words that often appear in the comment dataset.

ACKNOWLEDGEMENTS

The author thanked Allah SWT for his mercy so the author can complete this study well. The author also thanked family, lecturers, friends, and the parties who have given help and support for the success of this research.

REFERENCES

- [1] B. Pang and L. Lee, "Opinion Mining and Sentiment Analysis," *Foundations and Trends in Information Retrieval*, vol. 2, no. 1-2, p. 1– 135, 2008.
- [2] S.B. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques," *Informatica*, vol. 31, no. 3, p. 249-268, 2007 [Online]. Available: <http://www.informatica.si/index.php/informatica/article/view/148/140>. [Accessed: 04-Oct-2020].
- [3] Q. T. Ain, M. Ali, A. Riaz, A. Noureen, M. Kamran, B. Hayat, and A. Rehman, "Sentiment Analysis Using Deep Learning Techniques: A Review," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 8, no. 6, p. 424-433, 2017.
- [4] X. Ouyang, P. Zhou, C. H. Li, L. Liu, "Sentiment Analysis Using Convolutional Neural Network," *IEEE (International Conference on Computer and Information Technology)*, 2015.
- [5] L. B. Ilmawan, "Aplikasi Mobile Untuk Analisis Sentimen Pada Google Play," thesis, Jurusan Ilmu Komputer FMIPA UGM, 2014.
- [6] O. Kharisman, "Analisis Sentimen Pada Review Konsumen Maskapai Penerbangan Menggunakan Kombinasi Lexicon Berbasis Sentiwordnet dan Supervised Model," thesis, Jurusan Ilmu Komputer FMIPA UGM, 2017.
- [7] Y. Kim, "Convolutional Neural Networks for Sentence Classification," In *Proc. of EMNLP conference*, p. 1746-1751, 2014 [Online]. Available: <https://arxiv.org/abs/1408.5882>. [Accessed: 04-Oct-2020].

-
- [8] A. Razi, "Klasifikasi Artikel Berita Berbahasa Indonesia Menggunakan Convolutional Neural Network," thesis, Jurusan Ilmu Komputer FMIPA UGM, 2017.
- [9] E. Haddi, X. Liu, and Y. Shi, "The Role of Text Pre-processing in Sentiment Analysis," *Procedia Computer Science*, vol. 17, p. 26–32, 2013 [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1877050913001385>. [Accessed: 04-Oct-2020].
- [10] I. Hemalatha, P.G. Varma, and A. Govardhan, "Preprocessing the Informal Text for Efficient Sentiment Analysis," *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, vol. 1, July – August, 2012.
- [11] H. Siqueira and F. Barros, "A Feature Extraction Process for Sentiment Analysis of Opinions on Services," *Proceedings of International Workshop on Web and Text Intelligence*, 2010 [Online]. Available: http://sites.labicc.icmc.usp.br/wti2012/IIWTI_camera_ready/74769.pdf. [Accessed: 05-Oct-2020].
- [12] N. Wyse, R. Dubes, and A. K. Jain, "A critical evaluation of intrinsic dimensionality, *Pattern Recognition in Practice*", Morgan Kaufmann Publisher, Inc., p. 415-425, 1980.
- [13] V. N. Vapnik, "The Nature of Statistical Learning Theory", Springer-Verlag, New York, 1995.
- [14] C. M. Bishop, "Pattern Recognition and Machine Learning," Springer, New York, 2006.
- [15] A. Severyn and A. Moschitti, "UNITN: Training deep convolutional neural network for Twitter sentiment classification," In *Proc. of the 9th SemEval workshop*, p. 464-469, 2015.
- [16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," In *Advances in neural information processing systems*, p. 1097-1105, 2012 [Online]. Available: <https://arxiv.org/abs/1606.07006>. [Accessed: 08-Oct-2020].
- [17] X. Yang, C. Macdonald, and I. Ounis, "Using Word Embeddings in Twitter Election Classification," *Information Retrieval Journal*, vol. 21, p. 183–207, 2018 [Online]. Available: <https://arxiv.org/abs/1606.07006>. [Accessed: 08-Oct-2020].