Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 179 (2021) 728–735

Procedia Computer Science

www.elsevier.com/locate/procedia

5th International Conference on Computer Science and Computational Intelligence 2020

# Sentiment Analysis Using Word2vec And Long Short-Term Memory (LSTM) For Indonesian Hotel Reviews

Putra Fissabil Muhammad[a], Retno Kusumaningrum[a*], Adi Wibowo[a]

[a]*Department of Informatics Universitas Diponegoro, Jl. Prof. Soedarto SH Tembalang, Semarang 50275, Indonesia*

**Abstract**

Generally, Online Travel Agent (OTA) has a review element where clients can give reviews of the facilities they have used. Availability of a huge volume of reviews makes it troublesome for service executives to know the percent of reviews that have an effect on their services. Thus, it is essential to develop a sentiment assessment technique with respect to hotel reviews, particularly in Indonesian language. This research makes use of Long-Short Term Memory (LSTM) model as well as the Word2Vec model. The integration of Word2Vec and LSTM variables used in this research are Word2Vec architecture, Word2Vec vector dimension, Word2Vec evaluation method, pooling technique, dropout value, and learning rate. On the basis of an experimental research performed through 2500 review texts as dataset, the best performance was obtained that had accuracy of 85.96%. The parameter combinations for Word2Vec are Skip-gram as architecture, Hierarchical Softmax as evaluation method, and 300 as vector dimension. Whereas the parameter combinations for LSTM are dropout value is 0.2, pooling type is average pooling, and learning rate is 0.001.



*Keywords:* Hotel Reviews; sentiment analysis; Word2Vec; Long-short term memory.

* Corresponding author. Tel.: +62-24-7474754; fax: +0-000-000-0000 .
*E-mail address:* retno@live.undip.ac.id


10.1016/j.procs.2021.01.061

## 1. Introduction

The rapid progress of IT (information technology) cannot be ignored anymore since it is required to solve several problems. One of the examples is using IT for business like an e-commerce (electronic-commerce) business. Customers can easily get goods, services and assured security through electronic transactions due to e-commerce convenience. An instance of an e-commerce business is the OTA (Online Travel Agents) which enables bookings of train and plane tickets, hotels and other such facilities.

Moreover, OTA usually provides a feature of providing reviews for these services, so users can easily select trains, planes, hotels, and other such facilities with good suggestions. Such reviews can also enable service managers to assess consumer reactions to services provided in order that they can evaluate their services. Undoubtedly, positive reviews increase popularity and ratings of the services, whereas, on the contrary, negative reviews can serve as self-evaluation mechanism. Nonetheless, the volume of data to be assessed is so huge that it causes the managers problems in assessing the data. Thus, sentiment assessment is required to process the data and assess the existing reviews.

Sentiment assessment is used to gather and review viewpoints about products and services expressed in Tweets, reviews, comments, or blog posts [1]. Review data that is gathered is then processed to determine the response of each review, whether it is positive or negative. Sentiment assessment research considers the hotel reviews for the purpose of the study. It classifies them into 2 categories, i.e. positive and negative responses, based on various factors such as services, prices, location, food, facilities, etc. Sentiment assessments in Indonesian language texts have been carried out by applying different classical machine learning techniques such as Naïve Bayes [1-7], Support Vector Machines, Logistic Regression [1,5,6], and Latent Dirichlet Allocation [7]. Sentiment analysis research for reviews in Indonesian language has also been carried out using deep learning technique, i.e. the CNN (Convolutional Neural Network) [8].

The classical machine learning methods have a problem wherein it is difficult to determine in the feature extraction what to be included in the given model. If features are missing or incomplete, the model will generate imperfect outcomes. If there are too many features and if all of them do not contribute to the output of the model, this model will not give optimal performance. Nonetheless, it has been found that this problem can be overcome by using CNN (Convolutional Neural Network) since it is a deep learning technique that uses the neural network framework where every layer obtains input from the preceding layer and is delivered to the subsequent layer. Nonetheless, CNN also has the drawback that it cannot work with long sequential data. It is because CNN does not possess a memory, so it cannot retain information about the word meaning. This drawback can be overcome by using the LSTM (Long-Short Term Memory) model. LSTM is a kind of RNN (Recurrent Neural Network) architecture which is designed to "retain" values that have been obtained before for a specific period. LSTM consists of 3 gates that regulate the flow, which are the input gate, forget gate, as well as output gate. The input gate regulates the input of new data into memory, the forget gate regulates how long particular values get stored in memory, and the output gate regulates how much the value retained in memory influences the output block activation [9].

In learning the traits of an object, it is given unique attributes or can be known as features. Both classical and deep learning methods are unable to perform the process of assessment on input data if it is in the form of text or string, so it needs numbers as input. Thus, if the object that is received is in the form of string or text, then the process of feature extraction is required to convert the text into a numeric vector each of which represents a word. This process is called the word embedding process. One of the models used to produce word embedding is the Word2Vec model. Word2Vec generates a vector space obtained from the corpus, which consists of words that are similar in the corpus and are adjacent to one another in the Word2Vec space. Thus, the purpose of this research is to analyse sentiments present in the hotel reviews using LSTM model and word embedding by using the Word2Vec model. The primary contribution of this research is that it combines the best parameters of the LSTM and the Word2Vec models for categorising the sentiments, i.e. negative or positive sentiments, particularly for Indonesian language hotel reviews.

## 2. Methods

There are 2 major phases in this research, i.e. (i) preparation of dataset, and (ii) creation of the sentiment division model. The first phase, dataset preparation, involves 3 processes, which are data gathering, pre-processing, and training using the Word2Vec model. The next stage aims to create a sentiment division model based on the LSTM

model. It involves 2 processes which are dataset partition and training-testing of LSTM model. Fig. 1 depicts the entire process of this study.

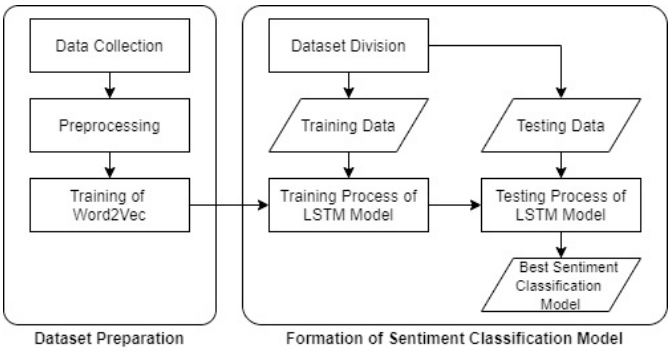


Fig. 1. Research overview

### *2.1. Data Collection*

In the process of data collection, the dataset considered is the set of hotel reviews found on the Traveloka website, which is gathered using Selenium and Scrapy libraries by automatic crawling. This crawling process begins from the initial link that is used to find the city/location of the hotels present on the website. Moreover, from the city/location of the hotel, we can get the hotel name which is situated in a renowned city/location. From every hotel name in the individual city/location, we can find a comment section which is implemented using pagination feature. Only the first page was crawled for this research. All processes are conducted repeatedly for various hotels. Then, this crawled data is divided based on its response, i.e. negative or positive sentiment. Overall, there are 2,500 sentiment texts consisting of 1250 positive and 1250 negative classes.

### *2.2. Pre-processing*

The objective behind data pre-processing is to transform it into a set of word arrays to allow additional pre-processing in the subsequent step. Pre-processing comprises multiple sub-processes, which are case folding, tokenisation, stemming, stopword removal, and padding. During case folding, all words are converted to lowercase and it is beneficial process because words initially in different forms are transformed to become uniform, thereby facilitating the same representation in the Word2Vec technique. Tokenisation is the following process, where the text is broken into tokens, which may be in the form of symbols, words, or distinct elements having a definite meaning [10]. The objective of stopword removal is to reduce words conveying little information so that the subsequent process in focused only upon the essential words. Stemming refers to the process where inflexion words are reduced to the root. The study employs the Sastrawi stemmer library, which may be accessed from http://sastrawi.github.io/. Padding comprises the final process during pre-processing. This is an essential step since all input documents concerning the LST training process should bear the same length. Every document in the dataset falls short of words compared to the maximum allowed number. In such cases, the token "<pad>" is appended until the document length reaches the maximum limit.

### *2.3. Training of word2vec*

Word2Vec was recommended by Mikolov, Corrado, Chen, & Dean [11]. The Word2Vec training process helps make the system learn vector representations of words by using the framework of neural networks. The input data fed into the Word2Vec system during training is the pre-processed hotel review data. The pre-processed hotel review data acts as an input to the Word2Vec system, whereas the output is in the form of a vector representation of every word. The initial process during the formulation of a Word2Vec model comprises the building of vocabulary using input data. Next, the vector representation of the word is learned. The Word2Vec model used in this study employs 12

different combinations obtained by experimenting with two Word2Vec architectures (Continuous Bag of Words – CBOW and the skip-gram), two evaluation techniques (Negative Sampling and Hierarchical Softmax), and three different vector dimensions (100, 200, 300). The detail information about Word2Vec architectures can be seen in [9]. The Hierarchical Softmax initially proposed by [12], where the output layer comprising the vocabulary set having a wordcount (W) is represented using a binary tree where each word is placed on a leaf node, while each node has a defined probability for its child nodes [11]. A random walk designates probabilities to the words. Negative sampling was introduced by [11], which is based on the idea of estimation of noise contrast, as is the case in Generative Adversarial Networks, i.e. a good model should be able to differentiate the real signal and the fake signal through logistic regression techniques. While the selection of word2vec dimension is based on the typical interval between 100-300 that is widely implemented in many previous studies. In addition, if we select the lower than 100 dimension then will lose properties of high dimensional spaces.

### *2.4. Dataset division*

For this study, the dataset division is performed using a 10-fold cross-validation technique. The dataset has a document count of 2500 in a balanced class scenario; hence, every fold comprises 250 documents, where 125 each represents positive and negative sentiment, respectively. The learning and testing process is performed in a pair, and there are ten repetitions, where every repetition uses nine folds for training and the one remaining fold for testing.

### *2.5. Training and testing process*

The objective of the LSTM training process is to formulate the sentiment classification model. In contrast, the objective of the testing process is to assess the performance of the classification model created during training. The training process has many steps, as depicted in Fig. 2.

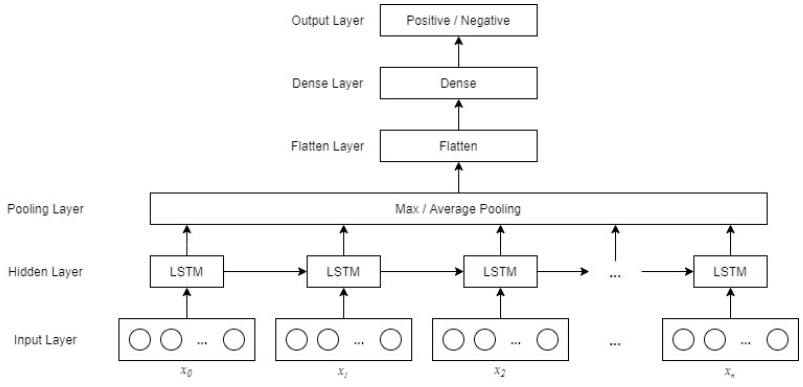


Fig. 2. LSTM model architecture

#### *2.5.1. Embedding layer formation*

Word embedding refers to using vectors to represent document vocabulary. The embedding layer is employed to facilitate the learning of word embedding. Some examples of the techniques employed for the learning word embedding are Glove and Word2Vec. This study uses the Word2Vec technique for word embedding.

#### *2.5.2. Encoding layer based on LSTM model*

LSTM is a specific variant belonging to the Recurrent Neural Network (RNN) method, which is meant to process data models having long sequences. The RNN technique extends the concept of typical feed-forward neural networks. Nevertheless, the RNN models suffer from the exploding or gradient vanishing challenges. The LSTM model was

formulated to address these challenges and is indeed able to do so. This capability can be attributed to the LSTM model possessing a cell memory, which can help the system retain its condition [13].

The LSTM model has three gates, namely input gate, forget gate, output gate along with the cell memory. The information to be updated and fed to the memory cell is determined by the gate input. The forget gate is responsible for determining if the input/output information if suitable for passage. If the result at the forget gate is close to zero, the information is forgotten, while if the result is close to one, the information is retained. This operation at the forget fate is what makes LSTM capable of handling the exploding problem and the vanishing gradient issues. The cell state is unaffected by the gate output; however, the date makes the distinction between the real information and the cell state.

#### *2.5.3. Pooling layer*

The pooling layers are responsible for reducing the input from a spatial perspective, as well as they facilitate a reduction in the network parameter count, thereby increasing computation speed also regulating overfitting. The pooling layer typically employs two types of pooling techniques, namely max pooling and average pooling. The values used in the two techniques are as specified in their name – maximum value for the former, and the average value for the latter.

#### *2.5.4. Fully connected layer*

The pooling layer outputs a multi-dimensional array, hence there is a requirement for a transformation to a vector for use as an input to the fully connected layer.

## 3. Results and Discussion

### *3.1. Experimental setting*

The specific parameters utilised in this study comprise Word2Vec and LSTM parameter sets. The specific Word2Vec parameters evaluated in this research comprise architectures (Skip-gram and CBOW), evaluation techniques (Negative Sampling and Hierarchical Softmax), and also vector dimensions (100, 200, and 300). Conversely, LSTM parameters comprise dropout values (0.2, 0.5, and 0.7), pooling methods (max and average pooling), and also learning rate values (0.0001 and 0.001). Consequently, 144 parameter combinations are evaluated in this research.

This study analyses how changes in Word2vec parameters can affect the accurateness of the sentiment-classification model. This analytical approach is delineated in the initial experimental scenario. The objective of the second experiment is to evaluate how changes in LSTM parameters can affect the accurateness of the sentiment-classification model. Consequently, the best sentiment classification scheme is determined from the results of both scenarios.

### *3.2. Scenario 1 – result and analysis*

As described before, Scenario 1 is focused on understanding the effects of changes in a range of Word2Vec parametric values. For every observed Word2Vec parameter, it is computed as the mean value for other combinations of parameters. In the example of vector dimension, the mean accuracy of the 48 other parametric combinations is computed for each. This calculation is then applied to observations of vector dimension parameters.

As shown in Fig. 3a, the Skip-gram architecture offers greater accuracy in comparison with Word2Vec CBOW architecture. The accuracy of the Skip-gram architecture attains 83.4%, whereas the CBOW architecture attains only 77.2%. This can be explained in terms of the Skip-gram scheme working to predict the context given from a single word, whereas the CBOW scheme supplies context words in order to predict a target. Sentiment analysis mostly develops more unique words, such that more such words will arise infrequently. The Skip-gram approach is

determinably more efficient at predicting words that arise infrequently than its rival CBOW. Consequently, Skip-gram is established as the best Word2Vec architecture for the LSTM model used in this research.

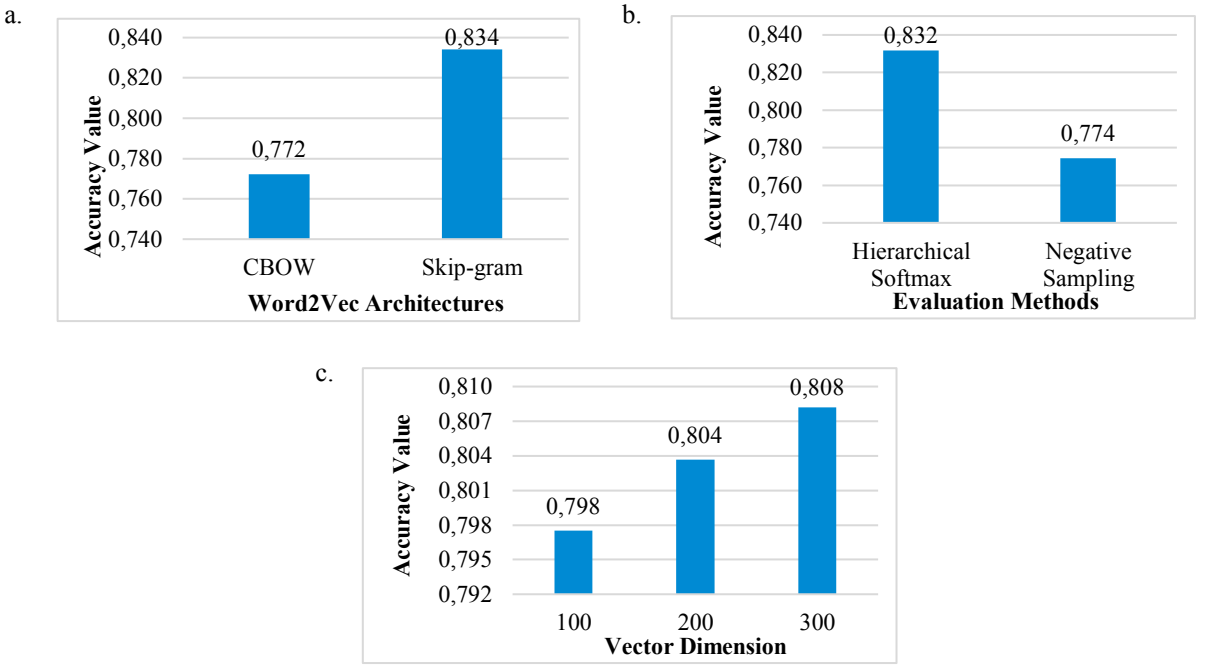


Fig. 3. Effect of word2vec parameter on accuracy value (a). architectures; (b). evaluation methods; (c). vector dimension

The Word2Vec evaluation method used with Hierarchical Softmax offers greater accuracy when compared to the Word2Vec evaluation method used with Negative Sampling, as shown in Fig. 3b. The Word2Vec model with Hierarchical Softmax attains a mean accuracy of 83.2%, whereas the Word2Vec evaluation technique with Negative Sampling attains a mean accuracy of 77.4%. This can be explained in terms of the Hierarchical Softmax evaluation technique using the binary tree scheme during training so as to represent every word in vocabulary and using leaf nodes to represent rare words, such that words which arise infrequently will inherit vector representations on them [15]. Reviews written by numerous users would feature differing writing styles, so various reviewed documents will necessarily feature dissimilar vocabularies. This results in numerous unique words being found in every document, which Hierarchical Softmax can manage.

Word2Vec vector dimensions with a value set to 300 offer greater accuracy in comparison with sizes set to 100 and 200. The vector dimension when set to 300 attains a mean accuracy of 80.8%, as shown in Fig. 3c. In any case, it can be concluded that increases in the size of vector dimensions are directly proportional to increases in accuracy.

### *3.3. Scenario 2 – result and analysis*

Three LSTM parameters are evaluated in Scenario 2, as described previously. The first comprises dropout values (0.2, 0.5, and 0.7), the second comprises pooling methods (mean and max), and the third and final parameter comprises learning rate values (0.0001 and 0.001). Scenario 2 is focused on evaluating how changes in these parametric values affect the accurateness of the sentiment-classification model. Similar to scenario 1, for every LSTM parameter observed, the measure is computed as the mean value for other combinations of parameters.

Dropout with a value set to 0.2 offers greater accuracy in comparison with dropout set to 0.5 and 0.7, as shown in the graph of Fig. 4a. Accuracy of training for dropout with 0.2 attains a mean accuracy of 81, 6%. A dropout with 0.5 and 0.7 attains a mean accuracy of 80.4% and 78.9%, respectively. It can be therefore concluded that mean accuracy value is inversely proportional to dropout, which implies that increases in dropout lead to decreases in accuracy values.

Consequently, when dropout values increase, the numbers of neurons that are eliminated become far more significant. With more and more neurons eliminated, the more challenged the system will be at learning. The lower the dropout value, the more efficient the LSTM model. Thus, the best dropout for the LSTM model in this study is 0.2.

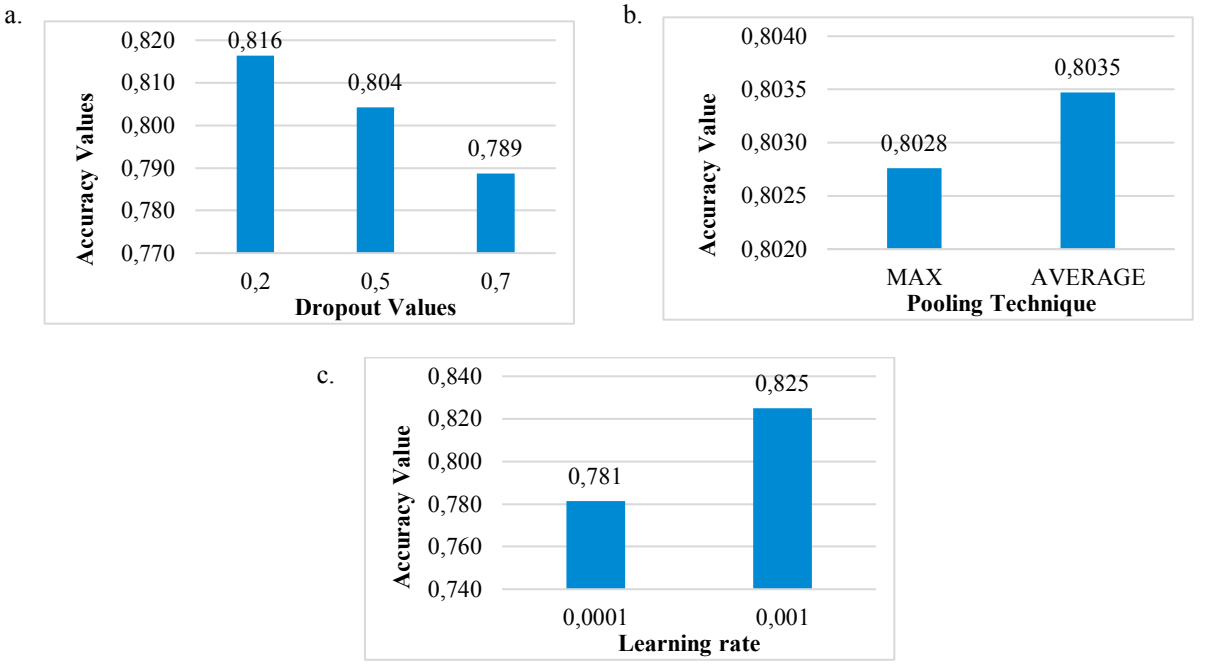


Fig. 4. Effect of LSTM parameter on accuracy value (a). dropout values; (b). pooling techniques; (c). learning rate values

Accuracy of training for all pooling layers that use averages attains a mean accuracy of 80.35% as shown in Fig. 4b. The pooling layer used with max value attains a mean accuracy of 80.28%. This is because extracting features with average results in LSTM operations is more appropriate than extracting the most substantial features. Thus, the best pooling layer for the LSTM model in this study is average.

Learning rate with the value set to 0.001 offers greater accuracy in comparison with that where the learning rate is set to 0.0001, as shown in the graph of Fig. 4c. Training with the learning rate value set to 0.001 attains a mean accuracy of 82.5%. Learning rate with the value set to 0.0001 attains a mean accuracy of 78.1%. It therefore can be concluded that the greater the learning rate, the greater the accuracy. Thus, the best learning rate for the LSTM model in this study is 0.001.

Based on the results of both experimental scenarios, the best scheme for sentiment classification using Word2Vec and LSTM models would be the classification scheme that uses Word2Vec architecture Skip-Gram, with Hierarchical Softmax used as a method of evaluation, and with vector dimension set to 300. For LSTM, a dropout value set to 0.2, with learning rate value set to 0.001, and with average pooling as the pooling method results in an accuracy value of 85.96%.

## 4. Conclusion

In this study, we have shown an implementation of Word2Vec and LSTM for classifying sentiment in hotel reviews. Both models are applied according to a number of observed parameters. Word2Vec model parameters comprise architectures (Skip-gram and CBOW), evaluation techniques (Negative Sampling and Hierarchical Softmax), and vector dimensions (100, 200, and 300). Conversely, LSTM parameters comprise dropout values (0.2, 0.5, and 0.7), learning rate values (0.0001 and 0.001), and pooling methods (max and average pooling). Consequently, a total of 144 parameter combinations are evaluated in this research. The best scheme attains a mean accuracy of

85.96%, which was generated using certain parameters, with Skip-Gram used as the Word2Vec architecture, with Hierarchical Softmax as a method of evaluation, and with vector dimension value set to 300. The scheme is also joined by an LSTM model with the dropout value set to 0.2, with learning rate set to 0.001, and with average pooling as the pooling technique.

**Acknowledgements**

This work was supported by the grant of Basic Research Number 257-20/UN7.6.1/PP/2020, Fiscal Year 2020 from Directorat Research and Development, Ministry of Research and Technology/National Agency for Research and Innovation, Indonesia.

**References**

1. Vinodhini G, Chandrasekaran R. Sentiment Analysis and Opinion Mining: A Survey. Int J Adv Res Comput Sci Softw Eng. 2012; 2(6).
2. Franky MR. Machine Learning-based Sentiment Analysis of Automatic Indonesian Translations of English Movie Reviews. In International Conference on Advanced Computational Intelligence and Its Application (ICACIA); 2008.
3. Kurniawan S, Kusumaningrum R. Hierarchical Sentence Sentiment Analysis of Hotel Reviews using The Naive Bayes Classifier. In Proc of the 2nd International Conference on Informatics and Computational Sciences; 2018.
4. Lutfi AA, Permanasari AE, Fauziati S. Sentiment Analysis in the Sales Review of Indonesian Marketplace by Utilizing Support Vector Machine. J Inf Syst Eng Bus Intell. 2018; 4(2).
5. Prahasiwi TG, Kusumaningrum R. Implementation of Negation Handling Techniques using Modified Syntactic Rule in Indonesian Sentiment Analysis. In J Phys Conf Ser; 2019.
6. Satriaji W, Kusumaningrum R. Effect of Synthetic Minority Oversampling Technique (SMOTE), Feature Representation, and Classification Algorithm on Imbalanced Sentiment Analysis. In Proc of the 2nd International Conference on Informatics and Computational Sciences; 2018. p. 99-103.
7. Wicaksono AF, Vania C, Distiawan TB, Adriani M. Automatically Building a Corpus for Sentiment Analysis on Indonesian Tweets. In The 28th Pacific Asia Conference on Language, Information and Computing; 2014. p. 185-194.
8. Bashri MFA, Kusumaningrum R. Sentiment analysis using Latent Dirichlet Allocation and topic polarity wordcloud visualization. In Proc of 5th International Conference on Information and Communication Technology, ICoIC7 2017; 2017.
9. Nawangsari RP, Kusumaningrum R, Wibowo A. Word2Vec for Indonesian Sentiment Analysis towards Hotel Reviews: An Evaluation Study. In Procedia Comput Sci.; 2019.
10. Sosa PM. Twitter Sentiment Analysis using combined LSTM-CNN Models. [Online].; 2017. Available from: HYPERLINK "http://www.academia.edu/download/55829451/sosa_sentiment_analysis.pdf".
11. Kannan S, Gurusamy V. Preprocessing Techniques for Text Mining. Int J Comput Sci Commun Networks. 2014; 5(1).
12. Mikolov T, Chen K, Corrado G, Dean J. Efficient Estimation of Word Representations in Vector Space. 2013.
13. Morin F, Bengjo Y. Hierarchical probabilistic neural network language model. In AISTATS 2005 - Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics; 2005. p. 246–52.
14. Sepp H, Jurgen S. Long Short-Term Memory. Neural Comput. 1997; 9(8).
15. Junker M, Hoch R, Dengel A. On the evaluation of document analysis components by recall, precision, and accuracy. In Proceedings of the International Conference on Document Analysis and Recognition, ICDAR; 1999. p. 717-20.