Accredited by National Journal Accreditation (ARJUNA) Managed by
Ministry of Research, Technology, and Higher Education, Republic Indonesia with Second Grade (Peringkat 2, Sinta 2)
since year 2017 to 2021 according to the decree No. 10/E/KPT/2019.

Published online on the journal's webpage: **http://jurnal.iaii.or.id**

RESTI JOURNAL
(System Engineering and Information Technology)
Vol. 4 No. 4 (2020) 723 - 729 ISSN Electronic Media: 2580-0760

Emotion Classification of Song Lyrics using Bidirectional LSTM Method with GloVe Word Representation Weighting

Jiddy Abdillah[1], Ibnu Asror[2], Yanuar Firdaus Arie Wibowo[2]
[1]Informatics Department, School of Informatics, Telkom University
[1]jiddyabdillah@student.telkomuniversity.ac.id, [2]iasror@telkomuniversity.ac.id, [3]yanuar@telkomuniversity.ac.id

Abstract

The rapid change of the music market from analog to digital has caused a rapid increase in the amount of music that is spread throughout the world as well because music is easier to make and sell. The amount of music available has changed the way people find music, one of which is based on the emotion of the song. The existence of music emotion recognition and recommendation helps music listeners find songs in accordance with their emotions. Therefore, the classification of emotions is needed to determine the emotions of a song. The emotional classification of a song is largely based on feature extraction and learning from the available data sets. Various learning algorithms have been used to classify song emotions and produce different accuracy. In this study, the Bidirectional Long-short Term Memory (Bi-LSTM) deep learning method with weighting words using GloVe is used to classify the song's emotions using the lyrics of the song. The result shows that the Bi-LSTM model with dropout layer and activity regularization can produce an accuracy of 91.08%. Dropout, activity regularization and learning rate decay parameters can reduce the difference between training loss and validation loss by 0.15.

Keywords: emotion classification, BiLSTM, deep learning, GloVe, song lyrics

1. Introduction

Music Emotion Recognition (MER) use musical features to identify emotions in music. The growing interest in evaluating MER system can provide the emotion of music resource. One of the musical features that are used to identify emotions in music is lyrics. Lyrics are semantically rich and expressive and have profound impact on human perception of music [1]. Classifying music emotion using lyrical feature can be done with text classification method. Algorithm such as machine learning algorithm is used often for text classification.

A model that can be used to classify emotions is the deep learning model. Research that uses a deep learning model to determine the emotion of a song usually uses the lyrics and audio features of the song. In this research, we try to determine the emotion of the song based on the lyrics feature alone and we test how deep learning model and the parameter performance affect the classification process. The deep learning model used in this research is Bidirectional Long-Short Term Memory (Bi-LSTM).

The reason we use Bi-LSTM model is because it considers the context of the text information and can get a better text representation [2]. The accuracy of the Bi-LSTM model for text classification problems also has a good result. In Xu et al. research [2] and in Chen et al. research [3], Bi-LSTM model got 92% and 95% accuracy. For word embeddings, we use GloVe pre-trained word embeddings which have better accuracy compared to other word embeddings models such as CBOW and skip-grams. Overall, GloVe outperformed other models in terms of word analogies, word similarities and named entity recognition tasks [4].

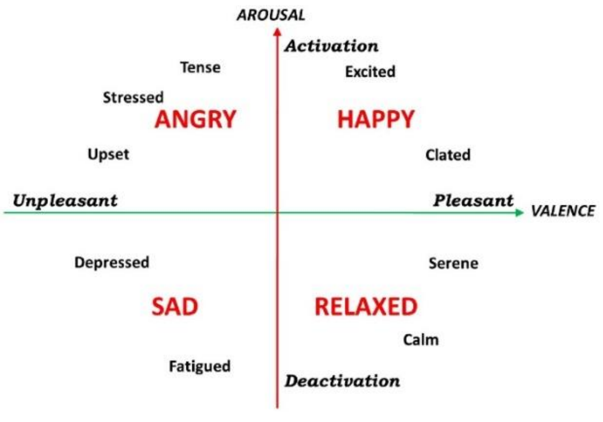

Figure 1. Circumplex Model

In the previous study of emotion classification based on song lyrics, the model used to determine the emotional class is categorical and dimensional models [5, 6, 7]. An emotional evaluation scale was made to evaluate

Accepted by editor: 13-07-2020 | Final Revision: 07-08-2020 | Online Publication : 20-08-2020

emotional state of the listener based on their verbal reports using many emotions and their dimensions (intensity, valence and dominance) [8]. One dimensional model that exists to determine an emotion is the circumplex model in Figure 1 developed by James Russel. This model shows that emotions are distributed in two-dimensional space, namely valence and arousal [9].

Past research related to the classification of emotions based on English song lyrics has been done several times. An et al. conducted a research using Naïve Bayes Classifiers with four classes emotion and got 68% accuracy [10]. Approach using unsupervised learning has also been done before, such as classification using lexicon and clustering. Erion Çano and Maurizio Morisio used ANEW and WordNet Lexicon for word representation and clustering to incorporate emotion values of all the sentences and emotion for the entire song. The accuracy of lexicon and clustering method compared with a lyrics dataset annotated by user tags and human subjects was 74.25%, compared with dataset created by listener and annotated label using user tags [6]. The drawback to this method is the lexicon has limited word and there are some words in song lyrics that are unused and left out in the process of training.

In research conducted by Revanth Akella and Teng-Sheng Moh, several deep learning models are used, such as Convolutional Neural Network (CNN), Bi-LSTM and Convolutional Recurrent Neural Network (CRNN). In that study, the CNN model produced an accuracy of 71%, Bi-LSTM 69.01% and CRNN 67.04% [5]. Although the classification of emotions based on lyrics using the Bi-LSTM method has been done, but the model made by Akella et al. didn't use the regularization layer like the dropout layer and the activity regularization layer. In the deep learning model, hyperparameter such as learning rate and decay scheduling also affect the performance of the model. Therefore, we will try to implement the regularization layer and hyperparameter tuning to the Bi-LSTM model.

2. Research Method

Emotional classification system based on song lyrics in this study was built to create a model that can learn the overall structure of song lyrics from front to back based on the context of the song lyrics. An overview of the system built can be seen from the flowchart of Figure 2.

2.1. Dataset

The song and emotion data used are taken from the MoodyLyrics [6] dataset. The column in the dataset consists of four columns, namely the song index, artist, song title and emotions. In the dataset there are no lyrics from the song. Therefore, we use data artists and song titles to search for lyrics on the Genius [11] and SongLyrics [12] websites by web crawling. The lyrics that have been taken are combined into the MoodyLyrics dataset. All the lyrics in the dataset are from English language. Examples of the dataset can be seen in Table 1.

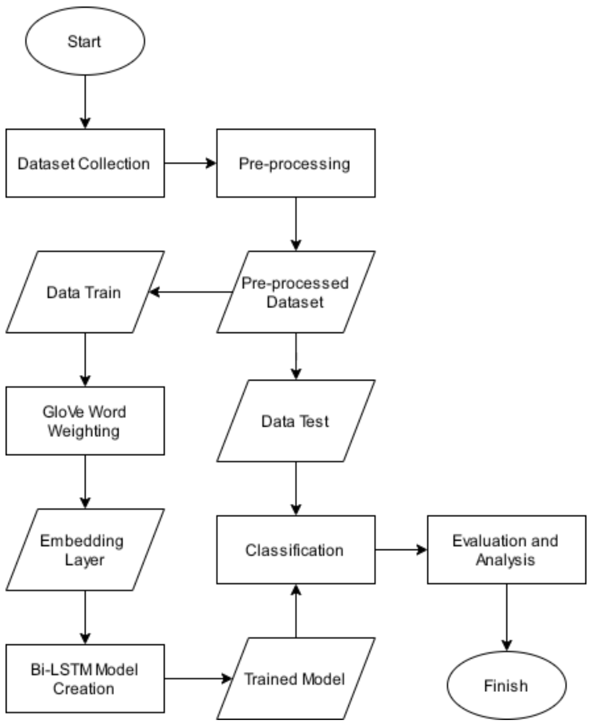

Figure 2. System Flowchart

Table 1. Dataset

Artist	Title	Lyrics	Emotion
Tyrese	One	Let me start by sayin' that you're the first one ….	Relaxed

Collected dataset are divided into two parts, data train and data test. A dataset of 2189 data was then broken down into 80% data train and 20% test data. The data train is used to input the model data created and the test data is used to input the classification and accuracy analysis of the model created.

2.2. Preprocessing

Preprocessing is the process of changing the form of unstructured data into structured data. The preprocessing stage converts textual data into data that is ready to be used as a text mining model [13]. Lyrics from the dataset that have been made are preprocessed to improve the structure and avoid imperfect data. There are several steps that are done at this stage, namely lemmatization. tokenization, stop-word removal and lowercase conversion [14].

2.3. Bidirectional Long-Short Term Memory

Long-Short Term Memory (LSTM) is a recurrent neural network (RNN) architecture designed to overcome gradient fluctuation problems in conventional RNN [15]. LSTM has three types of gates, namely forget gate, input gate and output gate. The structure of the LSTM model can be seen in Figure 3.

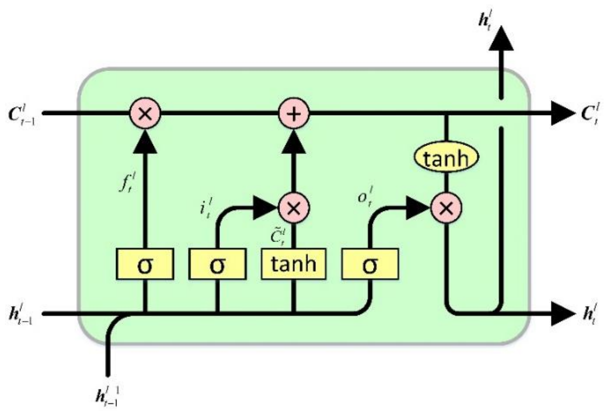

Figure 3. LSTM Structure Unit

The element in input sequence are tracked by cell memory to keep track the dependencies between the elements. The input gate handles new value that get into the cell. To choose which value remains in the cell, LSTM unit uses a forget gate. The remaining value in the cell will go to the output gate, where the computation starts with activation function of the LSTM, which often called logistic sigmoid function. The flow of LSTM is regulated with these equations as follows.

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{1}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1};\ x_t] +\ b_C) \tag{3}$$

$$C_t = f_t\ \times C_{t-1} + i_t * \tilde{C}_t \tag{4}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t \times tanh C_t \tag{6}$$

where:

- W_i, W_f, W_C, W_o: trained weights
- b_i, b_f, b_C, b_o: trained biases
- σ: sigmoid function
- x_t: input at time step t
- C_t: cell state at time step t
- h_t: output at time step t
- f_t: forget gate at time step t
- i_t: input gate at time step t
- o_t: output gate at time step t

One-way LSTMs can only use past contextual information. Bidirectional LSTM can use past and future contextual information, so that it can create two independent sequences of LSTM output vectors [16]. Both LSTM network parameters in the Bi-LSTM model have the same word embeddings of the sentence and both networks are independent. The output of each time step is a combination of two output vectors from both directions, formulated as follows where h_t is forward or backward state.

$$h_t = \overrightarrow{h_t} \oplus \overleftarrow{h_t} \tag{7}$$

2.4. Model Architecture

In this study, the Bi-LSTM model has several layers that are used can be seen in Figure 4, namely embedding layer, dropout layer, Bi-LSTM layer and output layer. All Bi-LSTM modeling and experiments were developed using *Keras* deep learning [17]. The default parameter configuration used in this study can be seen in Table 2.

Table 2. Model Parameter

Parameter	Value
Dimension	100
Number of Bi-LSTM Hidden Units	100
Maximum Sequence	1000
Batch Size	64
Activation Function	Softmax
Optimizer	ADAM

For the embedding layer we used GloVe 6B pre-trained word embeddings to assign values to word input. The GloVe word vector that we use is GloVe 100d which has 100 dimensions and 400000 words. GloVe is used because it is easier to catch word combinations than approaches with n-grams [4].

2.

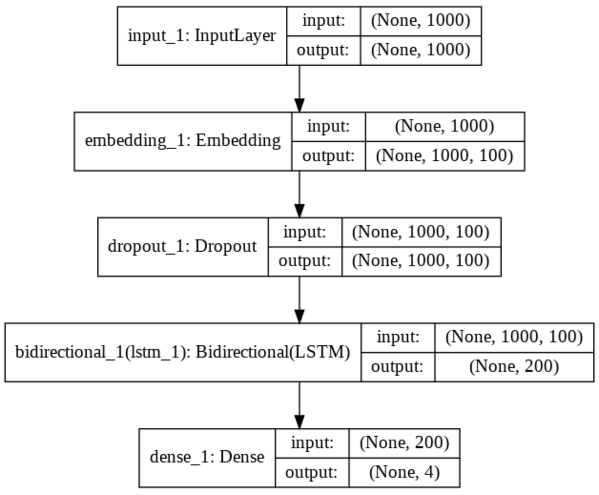

Figure 4. Model Architecture

2.5. Performance Measurement

To evaluate the model, we used the loss, accuracy, precision, recall and f1-score metrics obtained from the confusion matrix table.

In Table 3, True Positive (TP) is the value of the category of classification results and the actual category value is equally positive. False Positive (FP) is the value of the result of the positive classification category and the actual negative category value. True Negative (TN) is the classification value of the result of the classification and the value of the actual category is equally negative. Finally, False Negative (FN) is a negative classification result category and a truly positive category value [18]. To determine accuracy, precision, recall and f1-score, TP, TN, FP, and FN values are needed. The formulation of these metrics are as follows.

Table 3. Confusion Matrix

	Relevant	Not Relevant
Retrieved	True Positive	False Positive
Not Retrieved	False Negative	True Negative

$$Accuracy = \frac{TP + TN}{(TP + FP + TN + FN)} \quad (8)$$

$$Precision = \frac{TP_i}{(TP_i + FP_i)} \quad (9)$$

$$Recall = \frac{TP_i}{(TP_i + FN_i)} \quad (10)$$

$$F_1 = 2 \times \frac{Precision \ \times Recall}{Precision + Recall} \quad (11)$$

To calculate the loss of the model, the writer uses the categorical cross entropy function. This function is used when there are two or more classes. In this study, the four class that are used are Happy, Angry, Sadness and Relaxed. The formulation of categorical cross entropy seen as follows.

$$loss(p,t) = -\sum_{c=1}^{C} t_{o,c} \log(p_{o,c}) \quad (12)$$

where:

- p: prediction vector
- t: target vector
- C: class

3. Result and Discussion

In this section, there are four sub-sections that explain the results of the evaluation and analysis of the research that has been done. These sub-sections are Method Comparison, Hyperparameter Configuration, Parameter Performance Comparison, and Model Performance Analysis.

3.1. Method Comparison

We compare our model with three machine learning method, which was Naïve Bayes, K-Nearest Neighbor (KNN) and Support Vector Machine (SVM). We used two other deep learning method to compare as well, which was Long-short Term Memory (LSTM) and Convolutional Neural Network (CNN). The reason we compare the models is to see which model is the most suitable for this task.

3.1.1. Comparison with other Method

Naïve Bayes

Naïve Bayes classifiers are a simple supervised learning algorithm that uses Bayesian probabilistic with strong (naïve) independence assumption between features [19]. The parameter that we used was $\alpha = 0.05$

K-Nearest Neighbor

K-Nearest Neighbor (KNN) classifier is non-parametric supervised learning algorithm used for classification and regression analysis [20]. KNN doesn't learn about how to categorize the data, but only memorize the data by choosing similar feature vector or closest training example in the vector space. The parameter that we used was n = 29.

Support Vector Machine

Support Vector Machine (SVM) are supervised learning models that uses linear predictors in high dimensional feature spaces to search large margin separators to tackle sample complexity challenge [21]. The parameter that we used was linear kernel.

Convolutional Neural Network

Convolutional Neural Network (CNN) are a specialized kind of neural network for processing data that has known, grid-like topology [22]. Convolutional networks use convolution in place of general matrix multiplication in at least one of their layers. CNN mostly applied to analyze visual imagery, but it's often used in natural language preprocessing as well. The parameter that we used were three one dimensional CNN layer with filters = 128, kernel size = 5 and activation function = ReLU. We used three max pooling layer with pool size = 5 and output layer with activation function = Softmax.

Long-Short Term Memory

In this comparison, we used the same parameters as the Bi-LSTM model. We used epoch = 20 and learning rate = 0.0006.

3.1.2. Comparison Result

Table 4. Comparisons of the Different Methods

Method	Precision	Recall	F1-Score	Accuracy
Naïve Bayes	87%	81%	82%	83%
KNN	75%	74%	74%	76%
SVM	69%	68%	68%	71%
CNN	89%	89%	89%	90%
LSTM	90%	91%	90%	90%
Bi-LSTM	92%	90%	91%	91%

From Table 4, of the three machine learning models Naïve Bayes had the highest accuracy with 83% score. SVM performs the worst out of the machine learning models with 71% accuracy. The three deep learning models had similar accuracy around 90% and the highest accuracy is Bi-LSTM with 91% score. This proves that the Bi-LSTM model is the most suitable for this task.

3.2. Hyperparameter Configuration

The parameters tested for performance in this study are epoch, learning rate, dropout, activity regularization and learning rate decay. We use the number of epochs and the value of learning rate. The value that has the best effect on the model will be used as a base hyperparameter which will be compared and analyzed.

Epoch is a complete presentation of the data set that will be studied learning models. The more epochs are used, the ability of the model to generalize data patterns increases. However, if there are too many epochs, there will be problem with overfitting of the model and if there are too few epochs, there will be underfitting problem.

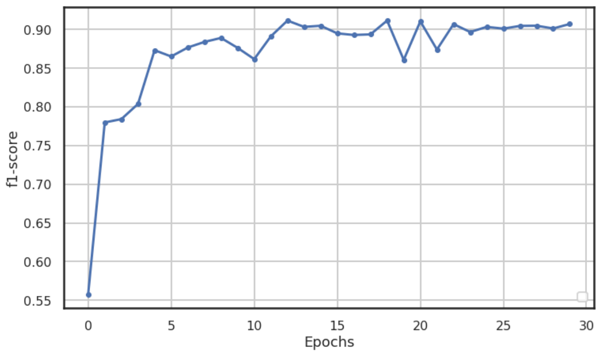

Figure 5. Relationship between Epoch and F1-Score

From Figure 5, the greater the epochs are, the model performance increase as we see in the f1-score graph. The maximum f1-score value occurs at the time of the 18th epoch and the performance of the model starts to look stable at the time of the 22nd epoch. Therefore, the number of epochs used on the model is 20.

To get the most out of our training data, we had to select the right learning rate. If the value of learning rate is too large, the model will easily fluctuate and surpassing the extreme point, thus making the model unstable. If the learning rate is too small, the model will take a long time to train and it can't learn the data optimally.

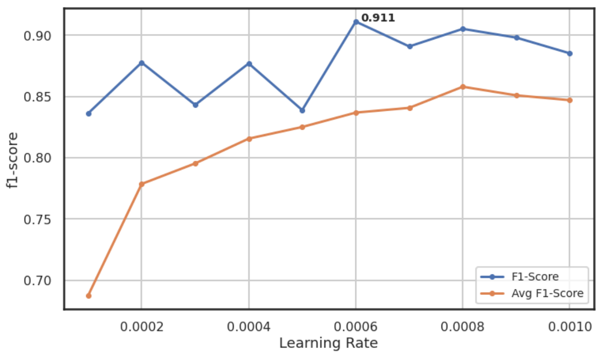

Figure 6. Relationship between Learning Rate and F1-Score

From Figure 6, the average f1-score increases with the increase in the learning rate and starts to decrease when the learning rate is 0.0008. The highest F1-score is 0.911 when the learning rate is 0.0006. Therefore, the value of learning rate that will be used on the model is 0.0006.

3.3. Parameter Performance Comparison

Learning rate decay, activity regularization and dropout are compared using the obtained hyperparameter and the values are 0.0006 for learning rate and 20 for epochs. The results of comparison of the parameter performance can be seen in Table 5.

Based on Table 5, we can see that the parameters can increase and decrease the performance of the model. The combination of dropout and learning rate decay has the worst performance with an f-score of 79.86%, a decrease of 12.08%. Among the three parameters, the learning rate decay had the greatest effect on performance decrease of 3.36% and the dropout had the smallest effect of decline at 0.44%. The parameter with the highest accuracy is dropout plus activity regularization which is 91.08%, an increase of 0.46%.

Table 5. Parameter Performance using Learning Rate = 0.0006 and Epochs = 20 (note. LR = Learning Rate, AR = Activity Regularization)

Parameter Configuration	Precision	Recall	F1-Score	Accuracy
-	91.86	90.39	91.12	90.62
Dropout(0.2)	91.20	90.16	90.68	90.39
LR Decay(0.8)	89.10	87.87	88.48	88.10
AR(0.001)	91.59	87.18	89.33	89.02
Dropout(0.2) + LR Decay(0.8)	81.77	78.03	79.86	80.78
Dropout(0.2) + AR(0.001)	91.83	87.41	89.57	91.08
AR(0.001)+ LR Decay(0.8)	91.89	85.58	88.63	89.24
Dropout(0.2) + AR(0.001)+ LR Decay(0.8)	90.24	86.73	88.45	89.02

3.4. Model Performance Analysis

Performance of the model without parameters and model using parameters are compared to get an overall performance picture. Parameter that we use are 0.2 dropout value, 0.8 learning rate decay value, and 0.001 activity regularization value. We used the 0.0006 learning rate value and 20 epochs for our hyperparameter from our hyperparameter configuration.

From Figure 8 and Figure 10, significant differences can be seen from the loss graph of each model. Models that use parameters have a smaller training loss and validation loss difference of 0.25, compared to models without parameters that have a difference of 0.4, decreasing 0.15.

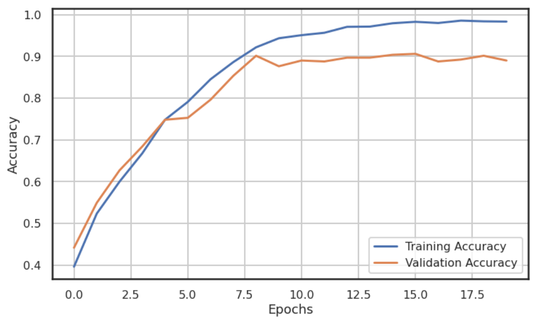

Figure 7. Model Accuracy Performance with Parameter

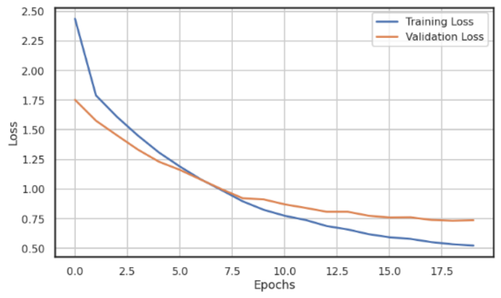

Figure 8. Model Loss Performance with Parameter

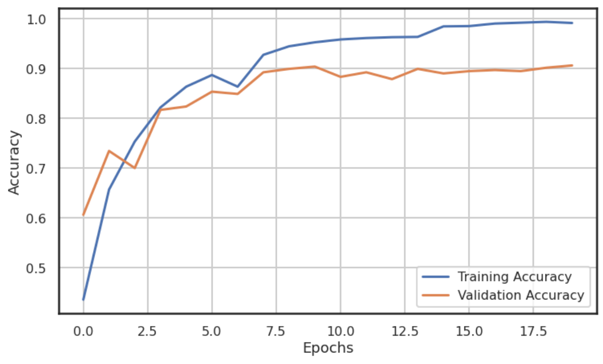

Figure 9. Model Accuracy without Parameter

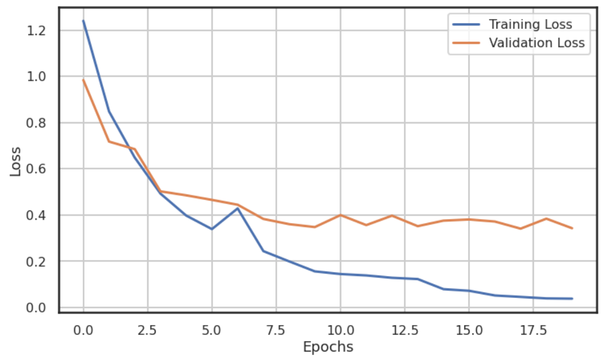

Figure 10. Model Loss without Parameter

A large difference between training loss and validation can cause overfitting on the model. In this case, the parameters can minimize overfitting of the Bi-LSTM model for the emotional classification of song lyrics, despite the final accuracy of the model, as we see from Figure 7 and Figure 9. Models with parameters also reduce the amount of accuracy and loss fluctuations in the model.

4. Conclusion

The Bi-LSTM model using dropout parameters and activity regularization can produce 91.08%, which is quite high compared with machine learning method in emotion classification based on song lyrics task. Dropout parameters, learning rate decay and activity regularization reduce the overfitting problem. Model with parameter had 0.25 difference between training loss with validation loss, smaller than the model without parameters with a difference of 0.4.

For future research, increasing the number of datasets can be useful to reduce the problem of overfitting the model performance. Other parameters such as the attention layer, convolution layer and pooling layer can be used to improve the accuracy and performance of the model. Increasing the number of datasets and layers will take longer to train the model.

References

[1] S. O. Ali and Z. F. Peynircioğlu, "Songs and emotions: are lyrics and melodies equal partners.," *Psychology of Music*, vol. 34, no. 4, p. 511–534, 2006.

[2] G. Xu, Y. Meng, X. Qiu, Z. Yu and X. Wu, "Sentiment Analysis of Comment Texts Based on BiLSTM," *IEEE Access*, vol. 7, pp. 51522-51532, 2019.

[3] C. W. Chen, S. P. Tseng, T. W. Kuan and J. F. Wang, "Outpatient Text Classification using Attention-based Bidirectional LSTM for Robot-assisted Servicing in Hospital," *Information (Switzerland)*, vol. 11, no. 2, p. 106, 2020.

[4] J. Pennington, R. Socher and C. D. Manning, "GloVe: Global Vectors for Word Representation," *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532-1543, 2014.

[5] R. Akella and T. S. Moh, "Mood Classification with Lyrics and ConvNets," *Proceedings - 18th IEEE International Conference on Machine Learning and Applications, ICMLA 2019*, pp. 511-514, 2019.

[6] E. Çano and M. Morisio, "MoodyLyrics: A sentiment annotated lyrics dataset," *ACM International Conference Proceeding Series*, vol. Part F1278, no. 11, pp. 118-124, 2017.

[7] R. Malheiro, R. Panda, P. Gomes and R. P. Paiva, "Emotionally-relevant features for classification and regression of music lyrics," *IEEE Transactions on Affective Computing*, vol. 9, no. 2, pp. 240-254, 2018.

[8] D. Yang and W. S. Lee, "Music Emotion Identification from Lyrics," *ISM 2009 - 11th IEEE International Symposium on Multimedia*, pp. 624-629, 2009.

[9] J. Russel, "A Circumplex Model of Affect," *Journal of Personality and Social Psychology*, vol. 39, no. 6, pp. 1161-1178, 1980.

[10] Y. An, S. Sun and S. Wang, "Naive Bayes Classifiers for Music Emotion Classification Based on Lyrics," *Proceedings - 16th IEEE/ACIS International Conference on Computer and Information Science, ICIS 2017*, no. 1, pp. 635-638, 2017.

[11] T. Lehman, "Genius," August 2009. [Online]. Available: http://www.genius.com.

[12] "SongLyrics," 28 September 1996. [Online].

[13] V. Srividhya and R. Anitha, "Evaluating Preprocessing Techniques in Text Categorization," *International Journal of Computer Science and Application*, pp. 49-51, 2010.

[14] A. K. Uysal and S. Gunal, "The Impact of Preprocessing on Text Classification," *Information Processing and Management*, vol. 50, no. 1, pp. 104-112, 2014.

[15] S. Hochreiter and J. Schmidhuber, "Long Short-term Memory," *Neural Computation*, vol. 9, pp. 1735-80, 1997.

[16] M. Schuster and K. K. Paliwal, "Bidirectional Recurrent Neural Networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, p. 2673–2681, 1997.

[17] F. Chollet, "Keras," 2015. [Online]. Available: https://keras.io.

[18] J. Han, M. Kamber and J. Pei, Data Mining Concepts and Techniques Third Edition, Burlington: Morgan Kauffman Publishers, 2012.

[19] I. Rish, "An Empirical of the Naive Bayes Classifier," *IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence*, vol. 3, pp. 41-46, 2001.

[20] T. M. Cover and P. E. Hart, "Nearest Neighbor Pattern Classification," *IEEE Transaction on Information Theory*, vol. 13, no. 1, pp. 21-27, 1967.

[21] S. Shalev-Shwartz and S. Ben-David, Understanding Machine Learning from Theory to Algorithms, New York: Cambridge University Press, 2014.

[22] I. Goodfellow, Y. Bengio and A. Courville, Deep Learning, Cambridge: MIT Press, 2016.