Sriwijaya Journal of Informatic and Applications
Vol. 01, No. 01, August 2020, pp. 34-44
ISSN xxxx-xxxx, DOI:xxxxx

Multilabel Classification for News Article Using Long Short-Term Memory

Winda Kurnia Sari [a,1], Dian Palupi Rini [a,2,*], Reza Firsandaya Malik [b,3]

[a] Master of Informatics Engineering, Universitas Sriwijaya, Palembang 30128, Indonesia
[b] Communication Network and Information Security Research Lab, Palembang 30128, Indonesia
[1] windakurniasari@unsri.ac.id; [2] dprini@unsri.ac.id*; [3] rezafm@unsri.ac.id
* corresponding author

ARTICLE INFO

Article history
Received
Revised
Accepted

Keywords
Recurrent Neural Network
Long Short-Term Memory
Multilabel Classification
Text Classification

ABSTRACT

Multilabel text classification is a task of categorizing text into one or more categories. Like other machine learning, multilabel classification performance is limited when there is small labeled data and leads to the difficulty of capturing semantic relationships. In this case, it requires a multi-label text classification technique that can group four labels from news articles. Deep Learning is a proposed method for solving problems in multi-label text classification techniques. By comparing the seven proposed Long Short-Term Memory (LSTM) models with large-scale datasets by dividing 4 LSTM models with 1 layer, 2 layer and 3-layer LSTM and Bidirectional LSTM to show that LSTM can achieve good performance in multi-label text classification. The results show that the evaluation of the performance of the 2-layer LSTM model in the training process obtained an accuracy of 96 with the highest testing accuracy of all models at 94.3. The performance results for model 3 with 1-layer LSTM obtained the average value of precision, recall, and f1-score equal to the 94 training process accuracy. This states that model 3 with 1-layer LSTM both training and testing process is better. The comparison among seven proposed LSTM models shows that model 3 with 1 layer LSTM is the best model.

1. Introduction

Multi-label classification [1] is a generalization of multiclass classification, which is the single-label problem in categorizing instances with only one or two classes. Multi-label problems exist in several domains, such as document classification [2], [3], text categorization [4], [5], social network [6], music emotions categorization [7], [8]. Previous research on multi-label text classification has involved traditional machine learning algorithms such as k-Nearest Neighbours [9], [10], Naive Bayes [11], [12], Support Vector Machine [13], [14], Logistic Regression [15]. In addition, compared to the traditional algorithm mentioned, it has certain limitations in terms of large-scale dataset training [17].

Just like other traditional single-label classifications, multi-label classifications have limitations when data labels are small [18]. In this case, a large-scale dataset based on previous research is used [19] to overcome multi-label classification in news articles. While news articles consist of several long sentences and can change their meaning if there are missing sentences. Therefore, Recurrent Neural Network (RNN) was chosen to solve this problem, since the recurrent structure is very suitable for long variable text processing [20]. One of deep learning methods proposed in this study is RNN architecture by applying the Long Short-Term Memory (LSTM) which able to expands the memory [21]. However, during training, traditional RNN has gradient vanishing and exploding problem, which can be solved by LSTM [22]. LSTM has different processing with a common RNN model. Another difference is an additional signal given from a time-step to the next time-step, which called context or memory cell.

A set of comparisons offered in this study has trained a deep learning model by adding seven LSTM variant models i.e 4 models with 1 layer LSTM, 2 layers and 3 layers LSTM, and Bidirectional LSTM (Bi-LSTM).

2. Related Work

a. Multilabel Classification

In many applications, basically text classification is multi-label [23]. For instance, online news articles [24], web classification [25], and information retrieval [26]. In order to overcome multi-label classification problem, two methods are applied. First, Problem Transformation methods and second is Adapted Algorithms [27]. Several problem transformation methods can be roughly broken down into binary classification problem, multi-class classification problem, and ensemble methods. For adapted algorithms, some classification algorithms have been adapted to the multi-label task directly. Boosting, k-nearest neighbors, decision tree, neural networks are proposed in adapted algorithm methods.

b. Deep Learning

Deep learning is an efficient version of neural networks [48] that can perform unsupervised, supervised, and semi-supervised learning [49]. Deep learning has been extensively used for image processing, but many recent studies have applied deep learning in other domains such as text and data mining. The basic architecture in a neural network is a fully connected network of nonlinear processing nodes organized as layers. The first layer is the input layer, the final layer is the output layer, and all other layers are hidden. Deep learning methods have already used for text classification are convolutional neural network [35]-[36], autoencoder [16], deep belief network [11]. One of deep learning architecture used in this paper is the Recurrent Neural Network (RNN). RNNs connect the output of a layer back to its input. This architecture is particularly important for learning time-dependent structures to include words or characters in text [50].

3. Method

a. Recurrent Neural Network

RNN is a type of neural network with a memory status for processing sequence inputs. Traditional RNN has a problem called gradient vanishing and exploding during training [20]. Recurrent node activation consists of feedback for itself from one time-step to the next. RNN is included in the deep learning category because data is processed automatically and without defining features [28]. RNN can use the internal states (memory) to process the input sequence. This makes it applicable to tasks such as Natural Language Processing (NLP) [29], speech recognition [30], music synthesis [31], time-series financial data processing [32]. There are two implementations of RNN i.e Backpropagation Through Time (BPTT) algorithm for calculating gradients and Vanishing Gradients problems that have led to the development of LSTM and GRU, the two most popular and powerful models currently used in NLP.

Basic equation of RNN,

$$s_t = \tanh(U_{x_t} + W_{s_{t-1}}) \quad (1)$$

$$\hat{y}_t = softmax(V_{s_t}) \quad (2)$$

Then define loss or error as cross entropy loss,

$$\begin{aligned} E_t(y_t, \hat{y}_t) &= -y_t log\hat{y}_t \\ E(y, \hat{y}) &= \sum_t E_t(y_t, \hat{y}_t) \\ &= -\sum_t y_t log\hat{y}_t \end{aligned} \quad (3)$$

Add up the gradients at each time-step for one training example,

$$\frac{\partial E}{\partial W} = \sum_t \frac{\partial E_t}{\partial W} \quad (4)$$

To calculate this gradient, chain-rule differentiation is used. That's the backpropagation algorithm when applied backwards starting from error.

$$\frac{\partial E_t}{\partial V} = \frac{\partial E_t}{\partial \hat{y}_t}\frac{\partial \hat{y}_t}{\partial V}$$

$$= \frac{\partial E_t}{\partial \hat{y}_t}\frac{\partial \hat{y}_t}{\partial z_t}\frac{\partial z_t}{\partial V}$$

$$= (\hat{y}_t - y_t) \cdot s_t \quad (5)$$

The calculated gradient,

$$\frac{\partial E_t}{\partial W} = \sum_{k=0}^{t}\frac{\partial E_t}{\partial \hat{y}_t}\frac{\partial \hat{y}_t}{\partial s_t}\frac{\partial s_t}{\partial s_k}\frac{\partial s_k}{\partial W} \quad (6)$$

Note that is the chain rule itself. Because it takes the derivative of a vector function, the result is a matrix,

$$\frac{\partial E_t}{\partial W} = \sum_{k=0}^{t}\frac{\partial E_t}{\partial \hat{y}_t}\frac{\partial \hat{y}_t}{\partial s_t}\left(\prod_{j=k+1}^{t}\frac{\partial s_j}{\partial s_{j-1}}\right)\frac{\partial s_k}{\partial W} \quad (7)$$

b. Long Short-Term Memory

Long short-term memory (LSTM) has recently become a popular tool among NLP researchers for their superior ability to model and learn from sequential data. These models have shown state-of-the-art results on various public benchmarks ranging from the classification of sentences [33] and various tagging problems [34] for language modeling [35]-[36], and sequence-to-sequence predictions [37]. LSTM aims to solve the RNN problem called gradient vanishing and exploding. LSTM replaces hidden vectors from recurrent neural networks with memory blocks equipped with gates. This can maintain long-term memory in principle by practicing appropriate gating weights and has proven to be very useful in achieving state-of-the-art for various problems, including speech recognition [38]. LSTM was proposed by Hochreiter and Schmidhuber, 1997 to specifically address this problem of learning long-term dependency. LSTM stores separate memory cells in it which can update and display their contents only if necessary [39]. The LSTM gates mechanism implements three layers; (1) inputs gate, (2) forget gate, and (3) output gate [40].

Each LSTM unit, can be seen in Fig. 1 has a memory cell, and the states at time t is represented as ct. Reading and modifying are controlled by the sigmoid gate and affect the input gate i_t, forget gate f_t and output gate o_t. LSTM is calculated as follows: At the moment of the moment, the model receives input from two external sources (h_{t-1} and x_t). The hidden states h_t is calculated by the x_t input vector the network received at time t and the previous hidden states h_{t-1}. When calculating the hidden layer node states, input gate, output gate, forget gate and x_t will simultaneously affect the state of the node.

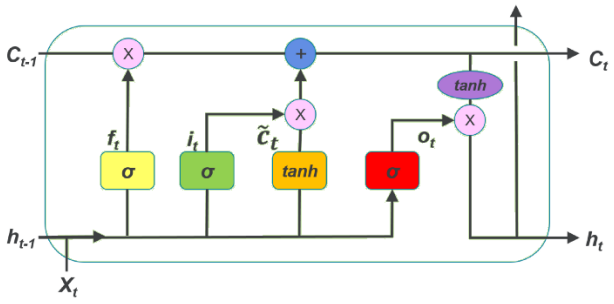

Fig. 1. LSTM Architecture

A step-by-step explanation of the LSTM cell and its gates is provided below:

1) Input Gate:

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \quad (8)$$

$$\check{C}_t = \tanh(W_C.[h_{t-1}, x_t] + b_C \quad (9)$$

2) Forget Gate:

$$f_t = \sigma\big(W_f.[h_{t-1}, x_t + b_f\big) \quad (10)$$

3) Memory State:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (11)$$

4) Output Gate:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (12)$$

$$h_t = o_t * \tanh(C_t) \quad (13)$$

c. Evaluation

The multi-label evaluation steps of the confusion matrix in the following equations:

$$Acc = \frac{\sum_{i=1}^{l} \frac{TP_i + TN_i}{TP_i + FN_i + TN_i + FP_i}}{l} * 100\% \quad (14)$$

$$Presisi = \frac{\sum_{i=1}^{l} TP_i}{\sum_{i=1}^{l} (FP_i + TP_i)} * 100\% \quad (15)$$

$$Recall = \frac{\sum_{i=1}^{l} TP_i}{\sum_{i=1}^{l} (TP_i + FN_i)} * 100\% \quad (16)$$

$$F1\ score = \frac{2 * Presisi * Recall}{Presisi + Recall} \quad (17)$$

where *l* indicates number of classes, *i* describes which class is selected.

d. Optimization

There are some types of optimizer for deep learning models such as SGD, Adam and RMSProp. This paper applied Adam and RMSProp for training the data. Adam Optimizer can control sparse gradient issues [41]. It is an expansion to stochastic gradient descent that has currently seen wider adoption for deep learning applications such as Natural Language Processing.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (18)$$

$$v_t = \beta_2 v_{t-1} + (1 + \beta_2) g_t^2 \quad (19)$$

where m and v refer averages the first two moments of gradient, g indicates gradient on current mini-batch. RMSProp is able to adapt the learning rate for each of the parameters. It aims to divide the learning rate for weight by a running average of the magnitudes of recent gradients for that weight [42].

$$v(w, t) \coloneqq \gamma v(w, t-1) + (1 - \gamma)(\nabla Q_i(w))^2 \quad (20)$$

Where γ is the forgetting factor.

And the parameters are updates as,

$$w := w - \frac{n}{\sqrt{v(w,t)}} \nabla Q_i(w) \quad (21)$$

4. Results and Discussion

a. Dataset

Previous research on Zhang 2015, Wang 2018 has shown work well with large-scale datasets [35], [43]. From eight large-scale datasets, the AGNEWS dataset was taken for training. AGNews is a classification of topics in four categories of Internet news articles consisting of titles and descriptions classified into four classes: World, Entertainment, Sports, and Business. The dataset is shown in Table 1, with the following content specifications:

Table 1. Dataset Specification

Dataset	Class	Contains
AGNews	4	496,835

b. Research Framework

Generally, the steps in the research methodology used to assist in compiling this research, require a clear framework for the stages. The research framework is used as in Fig. 2, which consists of a literature review, data preparation, pre-processing, classification process with LSTM, results analysis, and conclusions. The classification process with LSTM consists of 3 sub-processes, that is training, validation, and testing process.

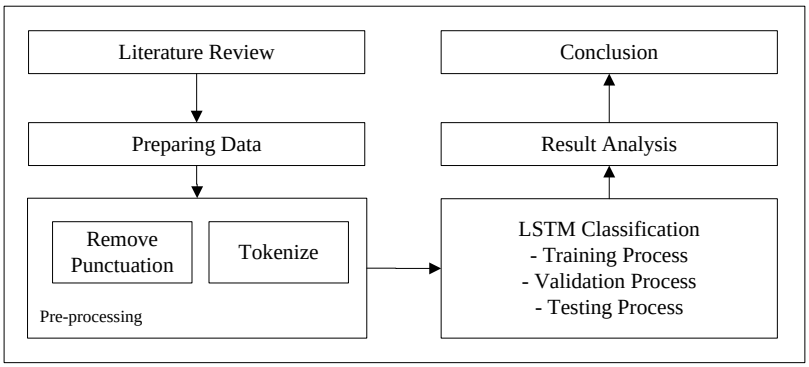

Fig. 2. Research Framework

c. Training Process

AGNews dataset is divided into 80% each for training and 20% for testing. The training dataset used is not used for LSTM testing, and vice versa. From 80% of the training data, 10% is used for the data validation process. The amount of each dataset is randomly split, with an automatic data split.

d. Training Models

Hyper-parameters used with Adam and RMSProp optimizers will be validated with a learning rate of 0.001, and dropout 0.5 is used to minimize errors. The output dimension is 256. The structure and hyper-parameters used in LSTM validation can be shown in Table 2.

Table 2. Training Models

	Optimizer	Loss Function	Activation Function	
			Hidden	Output
Model 1	Adam	Categorical Cross Entropy	Relu	Softmax
Model 2	Adam	Categorical Cross Entropy	Tanh	Softmax
Model 3	RMSProp	Categorical Cross Entropy	Relu	Softmax
Model 4	RMSProp	Categorical Cross Entropy	Tanh	Softmax

e. LSTM 1 Layer Model

The classification model with LSTM architecture is used to correct the classification problems in the standard RNN by modeling the LSTM 1 layer. Hyper-parameters used in this study used four LSTM models with a general structure of the softmax activation function at the output, loss function used categorical cross-entropy. Model results 1, 2, 3, and 4 with 1 layer LSTM in the multi-label text classification process by using different hyper-parameters shown in Table 3. The confusion matrix for each model can be seen in Table 4. The Performance Evaluation Results of the Multi-label Text Classification in the testing process for each model can be seen in Table 5.

Table 3. Performance Evaluation Results of Model LSTM 1 Layer in Training Process

Evaluation Parameters	Model 1	Model 2	Model 3	Model 4
Accuracy	97.41	97.37	94.95	94.93

Table 4. Confusion Matrix For Each Model with 1 Layer LSTM

Model	Confusion Matrix			
Model 1 LSTM	8676	120	151	109
	157	7793	703	202
	108	411	8367	99
	99	208	127	8670
Model 2 LSTM	8523	163	138	94
	106	8189	474	176
	108	612	8270	91
	98	239	116	8603
Model 3 LSTM	8708	73	113	66
	148	8153	521	165
	144	410	8425	81
	100	163	109	8621
Model 4 LSTM	8624	122	150	76
	83	8241	448	131
	74	498	8476	67
	127	205	128	8550

Table 5. Performance Evaluation Results of Model LSTM 1 Layer in Testing Process

Model	Class	Precision	Recall	F1-score	Data
Model 1 LSTM	0	97	96	96	8935
	1	92	90	91	9061
	2	93	92	92	9029
	3	94	96	95	8975
	Avg	**94**	**94**	**94**	**36000**
Model 2 LSTM	0	97	96	96	8970
	1	92	90	91	8970
	2	92	93	92	8982
	3	95	96	96	9078
	Avg	**94**	**94**	**94**	**36000**
Model 3 LSTM	0	96	97	96	8900
	1	93	90	91	9105
	2	91	94	92	9043
	3	96	96	96	8943
	Avg	**94**	**94**	**94**	**36000**

Model 4 LSTM	0	97	96	96	8914
	1	93	88	91	8981
	2	90	94	92	9036
	3	95	96	96	9069
	Avg	**94**	**94**	**94**	**36000**

Based on the results of the performance of the 4 models with 1 layer LSTM, model 3 and 4 have good results for the training and testing process by using the RMSProp optimizer thus the LSTM models 2 and 3 layers are trained with the same hyper-parameters as the relay in the hidden gate. RMSProp has shown excellent adaptation of learning rate in various applications. RMSProp can be seen as a generalization of Rprop and is able to work with mini-batches and only conflict with full-batches [44].

f. LSTM 2 Layers and 3 Layers (Stacked LSTM) Models

In this model, using 2 layers and 3 layers LSTM on top of each other, makes the model able to study higher-level temporal representations. The first two LSTMs return the full output sequence, but the last only returns the last step in the output sequence, so the dimensions dropping temporarily.

The results of the performance evaluation of the training process for models 2 layers and 3 layers LSTM are shown in Table 6, confusion matrix in Table 7 and the performance evaluation result of the testing process in Table 8.

Table 6. Performance Evaluation Results For 2 Layer And 3 Layer LSTM in Training Process

Evaluation Parameters	2 Layer LSTM	3 Layer LSTM
Accuracy	96.28	95.68

Table 7. Confusion Matrix For Each Label In 2 Layer Dan 3 Layer LSTM

Model	Confusion Matrix			
2 Layer LSTM	8597	117	96	98
	121	8193	517	198
	106	403	8424	86
	74	145	92	8733
3 Layer LSTM	8713	109	101	149
	101	8127	554	217
	109	362	8347	124
	91	167	94	8635

Table 8. Performance Evaluation Results For 2 Layer And 3 Layer LSTM in Testing Process

Model	Class	Precision	Recall	F1-score	Data
2 Layers LSTM	0	97	97	97	8908
	1	92	91	92	9029
	2	92	93	93	9019
	3	96	97	96	9044
	Avg	**94**	**94**	**94**	**36000**
3 Layers LSTM	0	97	96	96	9072
	1	93	90	91	8999
	2	92	93	93	8942
	3	95	96	95	8987
	Avg	**94**	**94**	**94**	**36000**

Overall, the 2 layer and 3 layer LSTM are not much different from previous Four models of LSTM with 1 Layer based on the performance evaluation results of precision, recall and f1-score evaluations. In this case, the 2 layer LSTM model gets the highest accuracy for testing accuracy compared to a simple LSTM and four models of 1 layer LSTM, that is 94.3.

g. Bidirectional LSTM (Bi-LSTM) Model

XXBidirectional LSTM (Bi-LSTM) works by connecting two hidden layers from opposite directions to the same output. The output layer can obtain information from the previous conditions (backward) and afterward (forward) simultaneously.

In this research, by trained Bi-LSTM 1 layer with the same hyper-parameters in model 2 and 3 layer LSTM. Previously trained using the Adam optimizer, but the high overfitting of the validation loss reduced the value of accuracy in the testing process, so the retraining process was done using the RMSProp optimizer. The results of the performance evaluation of the Bi-LSTM 1 layer training process model are shown in Table 9 and the results of the evaluation of the LSTM 1 layer testing performance are shown in Table 10.

Table 9. Performance Evaluation Results For Bi-LSTM in Training Process and Confusion Matrix for Each Class

Evaluation Parameters	**Bidirectional LSTM (BiLSTM)**			
Accuracy	95.88			
Confusion Matrix	8745	110	68	64
	142	8174	421	182
	172	466	8362	108
	127	175	52	8632

Table 10. Performance Evaluation Results For Bi-LSTM in Testing Process

Class	**Precision**	**Recall**	**F1-score**	**Data**
0	95	97	96	8987
1	92	92	92	8919
2	94	92	93	9108
3	96	96	96	8986
Avg	**94**	**94**	**94**	**36000**

h. Comparison Results

The performance evaluation result of RNN structure by applying the LSTM architecture in this study was compared with some previous studies for multi-label text classification. The results can be seen in the following Table 11.

The initial purpose of this study was to determine the structure of LSTM as a classification of multi-label text in large-scale datasets. With seven models of LSTM that have been trained and tested above does not subtract the fact that the performance results in this initial stage have not been done much pre-processing before being classified by the LSTM. For example, external word embedding is used, the resulting feature representation is obtained from the input of words that often appear 100,000 words with a sequence of words 130. In addition, the LSTM architecture used is still standard and simple. Even though it is not optimal for a simple LSTM network, the results of the performance evaluation in the training and testing process show good results. This can prove that LSTM is a pretty good method for classifying sequential texts. Table 11 shows a comparison of previous studies.

Table 11 shows comparison with a variety of methods, including (i) the bag-of-words in [36]; (ii) sophisticated deep CNN/RNN models: large/small word CNN, LSTM reported in [36] and deep CNN (29 layer) [45]; (iii) simple compositional methods: fast Text [46] and simple word embedding models (SWEM) [47]; (iv) the Label Embedding Attentive Model (LEAM) [42].

Table 11. Comparison of Previous Research

Model	AGNews
Bag-of-words (Zhang et al.,2015)	88.8
Small word CNN (Zhang et al.,2015)	89.13
Large word CNN (Zhang et al.,2015)	91.45
LSTM (Zhang et al.,2015)	86.06
Deep CNN (29 layer) (Conneau et al.,2017)	91.27
SWEM (Shen et al.,2018)	92.24
fastText (Joulin et al.,2016)	92.5
LEAM (Wang et al., 2018)	92.45
LEAM (linear) (Wang et al., 2018)	91.75
Proposed Model*	**93.9**

Fig. 3 shows also the comparison among training, validation, and testing accuracy for all proposed models and the comparison loss between training and testing can be seen in Fig. 4.

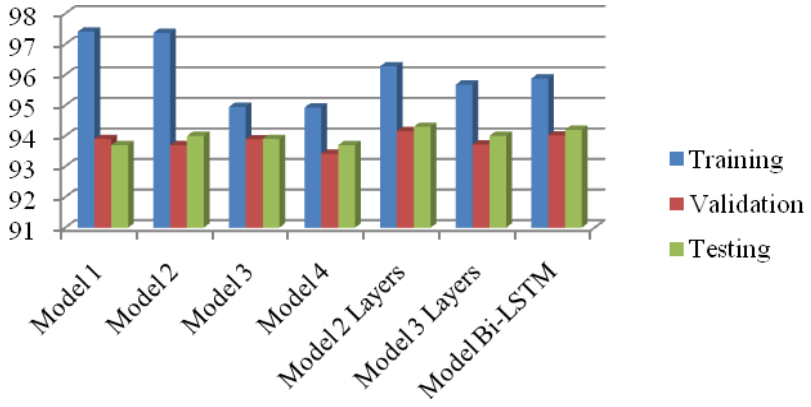

Fig. 3. Comparison Results of Training, Validation, and Testing Accuracy

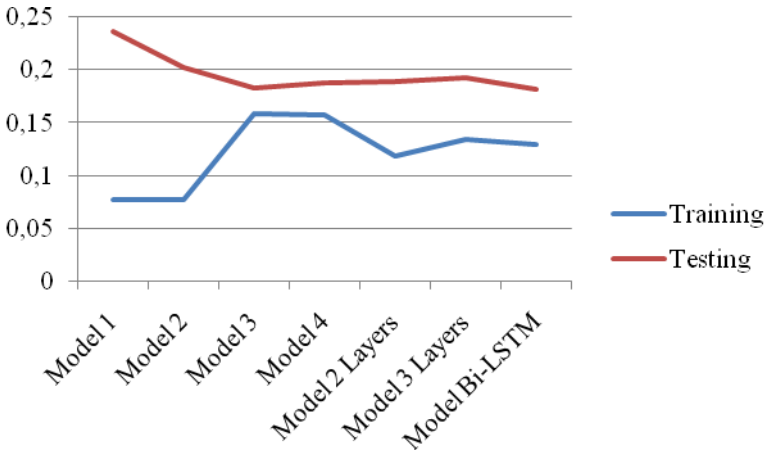

Fig. 4. The Comparison Loss Between Training and Testing

5. Conclusion

The classification of documents is a critical issue to deal with, given the increasing size of the scientific literature and other documents. When documents are arranged in a large-scale dataset, a multi-label approach is difficult to apply using traditional supervised learning methods. n this study comparing seven Long Short-Term Memory (LSTM) models using large-scale datasets. Based on experiments on several LSTM models show good performance results. Based on experiments on several LSTM models show good performance results. Performance evaluation results of the 2 layer LSTM model get the highest testing accuracy results from several models that have been done, is 94.3 but the result of the training process accuracy is 96.28. For model performance results with good performance indicated by model 3 with 1 layer LSTM based on the training accuracy at 94 with the average value of precision, recall, and f1-score of 94 for each label. LSTM can also be implemented with a number of different large-scale datasets by tuning the models.

References

[1] J. Petterson and Caetano, T.S, "Reverse multi-label learning". In Advances in Neural Information Processing Systems (pp. 1912-1920). 2010.

[2] S. Lai, L. Xu, K. Liu, and J. Zhao, 2015, "Recurrent convolutional neural networks for text classification". In Twenty-ninth AAAI conference on artificial intelligence.

[3] D. Zeng, K. Liu, S. Lai, S., G. Zhou, and J. Zhao, 2014, "Relation classification via convolutional deep neural network".

[4] G. Chen, D. Ye, Z. Xing, J. Chen, and E. Cambria, 2017, "Ensemble application of convolutional and recurrent neural networks for multi-label text categorization". In 2017 International Joint Conference on Neural Networks (IJCNN) (pp. 2377-2383). IEEE.

[5] F. Rousseau, E. Kiagias, and M. Vazirgiannis, "Text categorization as a graph classification problem," in ACL, 2015, pp. 1702–1712

[6] X. Wang and G. Sukthankar, "Multi-label relational neighbor classification using social context features," in Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2013, pp. 464–472.

[7] B. Wu, E.-H. Zhong, A. Horner, and Q. Yang, "Music emotion recognition by multi-label multi-layer multi-instance multi-view learning," in Proc. ACM Multimedia, 2014, pp. 117–126

[8] G. Tsoumakas, I. Katakis, and I. Vlahavas, 2008, "Effective and efficient multilabel classification in domains with large number of labels". In Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08) (Vol. 21, pp. 53-59). sn.

[9] B.Y. Pratama, and R. Sarno, 2015, "Personality classification based on Twitter text using Naive Bayes, KNN and SVM". In 2015 International Conference on Data and Software Engineering (ICoDSE) (pp. 170-174). IEEE.

[10] M. Azam, T. Ahmed, F. Sabah, F. and M.I. Hussain, 2018, "Feature Extraction based Text Classification using K-Nearest Neighbor Algorithm". IJCSNS Int. J. Comput. Sci. Netw. Secur, 18, pp.95-101.

[11] L. Jiang, C. Li, S. Wang, and L. Zhang, 2016, "Deep feature weighting for naive Bayes and its application to text classification". Engineering Applications of Artificial Intelligence, 52, pp.26-39.

[12] S. Xu, 2018, "Bayesian Naïve Bayes classifiers to text classification". Journal of Information Science, 44(1), pp.48-59.

[13] M. Fanjin, H. Ling, T. Jing, and W. Xinzheng, 2017, "The Research of Semantic Kernel in SVM for Chinese Text Classification". In Proceedings of the 2nd International Conference on Intelligent Information Processing (p. 8). ACM.

[14] M. Goudjil, M. Koudil, M. Bedda, and N. Ghoggali, 2018, "A novel active learning method using SVM for text classification". International Journal of Automation and Computing, 15(3), pp.290-298.

[15] A. Onan, S. Korukoğlu, and H. Bulut, 2016, "Ensemble of keyword extraction methods and classifiers in text classification". Expert Systems with Applications, 57, pp.232-247.

[16] R. G. F. Soares, 2018, "Effort Estimation via Text Classification and Autoencoders". Proceedings of the International Joint Conference on Neural Networks, 2018-July, 1-8. https://doi.org/10.1109/IJCNN.2018.8489030.

[17] M. Gao, T. Li, and P. Huang, 2018, "Text Classification Research Based on Improved Word2vec and CNN". In International Conference on Service-Oriented Computing (pp. 126-135). Springer, Cham.

[18] L. Li, M. Wang, L. Zhang, and H. Wang, 2014, "Learning semantic similarity for multi-label text categorization". In Workshop on Chinese Lexical Semantics (pp. 260-269). Springer, Cham.

[19] H. Wang, P. Yin, L. Zheng, and J.N. Liu, 2014, "Sentiment classification of online reviews: using sentence-based language model". Journal of Experimental & Theoretical Artificial Intelligence, 26(1), pp.13-31.

[20] Y. Yan, Y. Wang, WC. Gao, BW. Zhang, C. Yang, and XC. Yin, "LSTM^2: Multi-Label Ranking for Document Classification," Neural Processing Letters 47, no. 1, 2018, pp. 117-138

[21] S. Hochreiter, and J. Schmidhuber. "Long short-term memory." Neural computation 9, no. 8, 1997, p. 1735-1780

[22] Y.N. Dauphin, A. Fan, M. Auli, and D. Grangier, 2017, "Language modeling with gated convolutional networks". In Proceedings of the 34th International Conference on Machine Learning-Volume 70 (pp. 933-941). JMLR.

[23] S. Burkhardt, and S. Kramer, 2018, "Online multi-label dependency topic models for text classification". Machine Learning, 107(5), pp.859-886.

[24] D. Rahmawati, and M.L. Khodra, 2016, "Word2vec semantic representation in multilabel classification for Indonesian news article". In 2016 International Conference On Advanced Informatics: Concepts, Theory And Application (ICAICTA) (pp. 1-6). IEEE.

[25] I. Partalas, A. Kosmopoulos, N. Baskiotis, T. Artieres, G. Paliouras, E. Gaussier, I. Androutsopoulos, M.R. Amini, and P. Galinari, 2015, "LSHTC: A benchmark for large-scale text classification". arXiv preprint arXiv:1503.08581.

[26] D. Rahmawati, and M.L. Khodra, 2015, "Automatic multilabel classification for Indonesian news articles". In 2015 2nd International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA) (pp. 1-6). IEEE.

[27] G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, and I. Vlahavas, 2011, "Mulan: A java library for multi-label learning". Journal of Machine Learning Research, 12(Jul), pp.2411-2414.

[28] T. Wiatowski, and H. Bölcskei, 2017, "A mathematical theory of deep convolutional neural networks for feature extraction". IEEE Transactions on Information Theory, 64(3), pp.1845-1866.

[29] K. Kowsari, D.E. Brown, M. Heidarysafa, K.J. Meimandi, M.S. Gerber, and L.E.Barnes, 2017, "Hdltex: Hierarchical deep learning for text classification". In 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA) (pp. 364-371). IEEE.

[30] H. Zen, and H. Sak, 2015, "Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis". In 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 4470-4474). IEEE.

[31] Li, K., Daniels, J., Liu, C., Herrero-Vinas, P. and Georgiou, P., 2019. "Convolutional recurrent neural networks for glucose prediction". IEEE Journal of Biomedical and Health Informatics.

[32] K. Tseng, C. Ou, A. Huang, R.F. Lin, and X. Guo, 2019, "Genetic and Evolutionary Computing ". Vol. 834. https://doi.org/10.1007/978-981-13-5841-8

[33] C. Zhou, C. Sun, Z. Liu, and F. Lau, 2015, "A C-LSTM neural network for text classification". arXiv preprint arXiv:1511.08630.

[34] M. Pota, F. Marulli, M. Esposito, G. De Pietro, and H. Fujita, 2019, "Multilingual POS tagging by a composite deep architecture based on character-level features and on-the-fly enriched Word Embeddings". Knowledge-Based Systems, 164, pp.309-323.

[35] Y. Kim, 2014. "Convolutional neural networks for sentence classification". arXiv preprint arXiv:1408.5882.

[36] X. Zhang, J. Zhao, and Y. LeCun, 2015, "Character-level convolutional networks for text classification". In Advances in neural information processing systems (pp. 649-657).

[37] I. Sutskever, O. Vinyals, and Q.V. Le, 2014, "Sequence to sequence learning with neural networks". In Advances in neural information processing systems (pp. 3104-3112).

[38] C.C. Chiu, T.N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R.J. Weiss, K. Rao, E. Gonina, and N. Jaitly, 2018, "State-of-the-art speech recognition with sequence-to-sequence models". In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 4774-4778). IEEE.

[39] A. Graves, 2013, "Generating sequences with recurrent neural networks". arXiv preprint arXiv:1308.0850.

[40] A. Kumar, and R. Rastogi, 2019, "Attentional Recurrent Neural Networks for Sentence Classification". In Innovations in Infrastructure. pp. 549-559. Springer, Singapore.

[41] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.

[42] T. Tieleman, T. and G. Hinton, 2012, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude". COURSERA: Neural networks for machine learning, 4(2), pp.26-31.

[43] G. Wang, C. Li, W. Wang, Y. Zhang, D. Shen, X. Zhang, R. Henao, and L. Carin, 2018, "Joint embedding of words and labels for text classification". arXiv preprint arXiv:1805.04174.

[44] G. Hinton, N. Srivastava, and K. Swersky, 2012, "Overview of mini-batch gradient descent". Neural Networks for Machine Learning, 575.

[45] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, 2016, "Very deep convolutional networks for text classification". arXiv preprint arXiv:1606.01781.

[46] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, 2016, "Bag of tricks for efficient text classification". arXiv preprint arXiv:1607.01759.

[47] D. Shen, G. Wang, W. Wang, M.R. Min, Q. Su, Y. Zhang, C. Li, R. Henao, and L. Carin, 2018, "Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms". arXiv preprint arXiv:1805.09843.

[48] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," science, vol. 313, no. 5786, pp. 504–507, 2006.

[49] R. Johnson and T. Zhang, "Effective use of word order for text categorization with convolutional neural networks," arXiv preprint arXiv:1412.1058, 2014.

[50] L. Medsker and L. Jain, "Recurrent neural networks," Design and Applications