LONTAR KOMPUTER VOL. 15, NO. 2 AUGUST 2024
DOI : 10.24843/LKJITI.2024.v15.i02.p04
Accredited Sinta 2 by RISTEKDIKTI Decree No. 158/E/KPT/2021

p-ISSN 2088-1541
e-ISSN 2541-5832

BERT Uncased and LSTM Multiclass Classification Model for Traffic Violation Text Classification

Komang Ayu Triana Indah[a1], I Ketut Gede Darma Putra[a2], Made Sudarma[b3], Rukmi Sari Hartati[b4], Minho Jo[b5]

Department of Information Technology, Bali State Polytechnic,
Jimbaran Hill, South Kuta, Badung 80361, Indonesia
[1]triana_indah@pnb.ac.id

[b]Department of Information Technology, Udayana University,
Jimbaran Hill, South Kuta, Badung 80361, Indonesia
[2]ikgdarmaputra@unud.ac.id
[3]msudarma@unud.ac.id

[d]Department of Electrical Engineering, Udayana University,
Jimbaran Hill, South Kuta, Badung 80361, Indonesia
[4]rukmisari@unud.ac.id

[e]Department of Computer and Information Science, Korea University,
Sejong Metropolitan City, South Korea 339-700
[5]minholo@korea.ac.kr

Abstract

The increasing amount of internet content makes it difficult for users to find information using the search function. This problem is overcome by classifying news based on its context to avoid material that has many interpretations. This research combines the Uncased model BiDirectional Encoder Representations from Transformer (BERT) with other models to create a text classification model. Long Short-Term Memory (LSTM) architecture trains a model to categorize news articles about traffic violations. Data was collected through the crawling method from the online media application API through unmodified and modified datasets. The BERT Uncased-LSTM model with the best hyperparameter combination scenario of batch size 16, learning rate 2e-5, and average pooling obtained Precision, Recall, and F1 values of 97.25%, 96.90%, and 98.10%, respectively. The research results show that the test value on the unmodified dataset is higher than on the modified dataset because the selection of words that have high information value in the modified dataset makes it difficult for the model to understand the context in text classification.

Keywords: *Multiclass Classification, BERT Uncased, LSTM, Traffic Violations*

1. Introduction

Based on a survey conducted by the Reuters Institute in the Digital News Report 2023, 88% of Indonesians use online media as a news source because of the ease of using the search feature to search for news by typing in keywords or hashtags. However, accuracy is compromised when using terms that have multiple meanings, necessitating news categorization based on context to avoid words that have double meanings. This study concentrates on reporting traffic violations due to increased accidents caused by the public's lack of knowledge in obeying traffic regulations [1]. To reduce the number of accidents in the future, news should inform the public about various situations, including categories and types of traffic violations and related sanctions. Natural language processing (NLP) is a subfield of artificial intelligence that allows computers to understand text and categorize objects automatically depending on their context [2]. The increasing volume of user uploads on various social networks has led to the frequent development of natural language text classification research subjects. The text classification process is divided into two types, namely binary and multiclass classification. Pretrained words are refined models that have been fed into large, general datasets to gain a better understanding of semantics and syntax. Encoder Representation BiDirectional Transformers (BERT) have achieved state-of-the-art performance in many NLP-related studies [3]. This model uses the Transformer mechanism, which learns contextual relationships between words in a text with self-attention tools [4]. The BERT Recurrent Neural Network (RNN) architecture is modified using Long Short-Term Memory (LSTM) to overcome the problem of vanishing gradients when processing long sequential data [5]. The sentiment analysis development model uses Word2Vec and LSTM with an accuracy of 85.96%. Several studies identified news-related articles using search methods categorized into five classes and word refinement from the Keras library with CNN architecture and achieved an F1 score of 90.2% [5]. Previous research focused on classifying Indonesian online news based on four currently popular topics using Word2Vec and K-Nearest Neighbor with an accuracy of 89.2% [6]. Therefore, this research proposes a text classification model that combines the BERT Uncased model, which was previously trained with one of the RNN architectures, namely LSTM, to classify traffic violation news into several categories according to the context [7].

2. Research methods

The study approach consists of carrying out schematic procedures using BERT and LSTM. BERT utilizes self-attention mechanisms to understand contextual relationships between words in a text. This self-attention mechanism allows input elements to interact more naturally and determine which ones require more attention. The word sequence representation of a phrase is generated by connecting words in the same sequence using an encoder and decoder algorithm. BERT Uncased is differentiated from Cased by its training techniques, specifically the use of text cases during WordPiece casenization with **the** addition of an accent mark **and** not using lowercase letters [8]. BERT (Bidirectional Encoder Representation from Transformers) and LSTM (Long Short Term Memory) are two different architectures used in natural language processing (NLP). Below is an explanation of how each works and how they can work together on this research [3]:

2.1. BERT

BERT is a transformer model used to produce excellent text representation. BERT works through stages of Tokenization that change text input into tokens using the BERT tokenizer. Next is the embedding process, namely, change tokens become embedding vectors representing words in lower dimensions. The next step is p encoding position, p . This is done because the transformer in BERT does not have an intrinsic order like RNN; the position information of tokens in a sentence is added through position coding. In the next step, the Transformer layer with process t ex passes through several transformer encoder layers using self-attention mechanisms. The output of the last layer is a vector representation of the input text that can be used for various NLP tasks such as classification, entity recognition, and more [9].

2.2. LSTM

LSTM is a type of RNN designed to overcome the vanishing gradient problem by storing information over a long period. How it works:

a. Input Gateway: Determines which information will be updated.
b. Forget Gate: Determines which information will be removed from the cell.
c. Output Gate: Determines which part of the cell will be output.
d. Cell Status: Stores information over time. This information is updated by the input gate and forget gate [10].

2.3. Combination of BERT and LSTM

A combination of BERT and LSTM can combine the strengths of both models. Here's how it works: Stepj First, BERT as a Feature Extractor input text is first processed by BERT to produce a high-quality vector representation. Next, LSTM for Sequence Modeling does r evector representation of BERT, which is then given as input to LSTM to handle sequence and temporal dependencies in the data. The final LSTM output can be used for various tasks such as classification, text generation, or sequence prediction.

2.4. BERT and LSTM Combination Working Scheme:

On the scheme, the Work combination of BERT and LSTM, i.e., step first enters t ex, then continues with the tokenization process and BER embedding. Step This converts t ex into a token and an embedding vector. Next is the BERT encoder layer, which processes embedding tokens to produce a deep vector representation and form an LSTM layer for BERT vector representation to establish sequence and temporal context. The output layer of LSTM results is used for specific tasks, such as sentiment classification. The first testing step is to determine the hyperparameters, which consist of two, namely no need tuning and need tuning [11].

2.5. Data Set Creation

The crawling technique produces news articles about traffic violations, which are then included in the dataset. To use this process online, use the API key listed on the detik.com website, then use the Tweepy library, the Python programming language, and the API Search technique. The data is labeled according to the subject determined through the crawling procedure. The information collected includes around 200 news stories divided into 14 categories, namely Staying in the Stop Lane, Stopping at Zebra Crossings, Turning on Headlights, Pedestrian Lanes, No Passing, No U-Turns, Traffic Signs Ahead, Using Cell Phones While Driving, Driving Against the Lane, Violating Traffic Signs, Not Using a Helmet, Driving Exceeding the Speed Limit, Underage Driver, Not Using a Seat Belt. Duplicate and inappropriate content data is removed during the human flagging procedure [3]. The scikit-learn library was used to divide the data set into training, validation, and test sets with proportions of 70%, 20%, and 10%, respectively. Two scenarios were included in the data processing [9] d unmodified assets (collection data words that contain certain essential information are appropriate categories) and modified data sets (deleting and changing data for each category) [12].

2.6. BERT Uncased-LSTM Text Classification Model Design

The initial step is to input the text/sentences resulting from the API news crawl, removing special characters, stopping words, and lowercase characters. The next stage is the Tokenizing process via BERT Tokenizer, which consists of the Token Filter, Stopwords, Stemming, and Token process Weighting. Next, the BERT Base Model with CLS (Classification Level Sentences) in the form of a Hidden State determines the classification level of a sentence. The proposed model is BERT Uncased -LSTM, and the design flow of the text classification model is shown in Figure 1 [11].

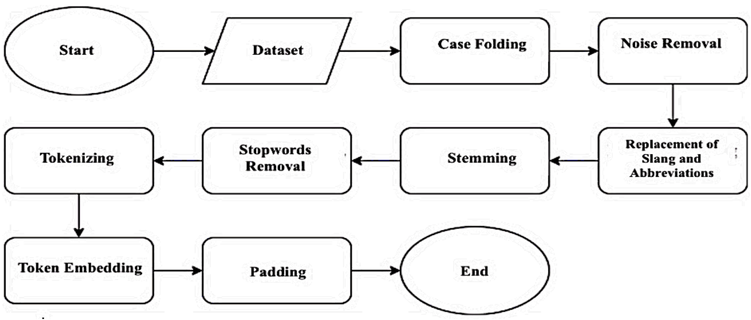

Figure 1. Text Classification Model Diagram with BERT Uncased-LSTM

After crawling the data via the www.detik.com media API, enter the Transform process case / cleaning for processing. A preliminary text is the next step after creating a data set. It aims to eliminate useless characteristics and distractions to organize the text material efficiently and prepare it for the next stage. This procedure is also essential because of the tokenizer approach WordPiece is used; the text preprocessing for the BERT Uncased model is slightly different from other word refinement models. An input representation that the BERT model can accept must be created to do this. Figure 2 illustrates the text preprocessing design in the BERT Uncased -LSTM model [7].

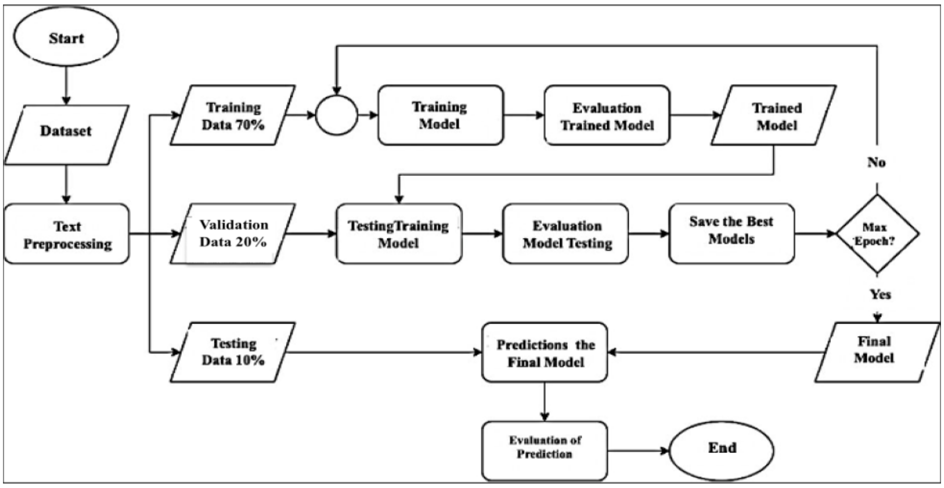

Figure 2. System Process Diagram with Training Data

One of the crucial stages in creating a text classification model is text preparation. The dataset is shared into training data (70%), validation data (20%) and testing data (10%) (3]Training data will form the next training model through an evaluation process. Validation data will test training models for evaluation to produce the best model. Meanwhile, testing data works for final model predictions and evaluating results predictions. The process was previously carried out with several iterations at the maximum specified epoch.

2.7. Casingless BERT Model Designs-LSTM

A BERT Uncased model pre-trained with the BERTBASE measure is used in this study to enable the embedding procedure 1 4 class. The encoder feeds a previously created input representation into the BERT Uncased model. Using a feed-forward network to generate output, each encoder uses independent attention, and this iterative process is carried out with successive encoders. Determining the hyperparameters is essential to achieving the best model performance. These hyperparameters are divided into default parameters that do not need to be adjusted and variables that have been adjusted to improve overall model performance. Table 1 displays the complete list of hyperparameters for use with the BERT Uncased-LSTM model. [11].

Table 1 . Database Characteristics[11]

Group	Hyperparameters	Mark
No need Setup	Period	50
	Max Sequence	256
	Possibility of Dropping Out of School	40%
	Activation Function	Softmax
	Loss Function	Categorical cross-entropy
	Batch File	32 or 64
Requires Setup	Learning Speed	2e-5 or 5e-5
	Combination Method	Average pooling or max pooling

Several libraries from different NLP applications, including Hugging Face, TensorFlow, and Scikit-learn, were used in this study. Learning methods used for Need Tuning are Hyperparameter, Epoch, Batch Size, Learning Rate, Probability of Dropout, Pooling Technique, Loss and Activation Functions, and purpose For optimizing the computing process. Hyperbolic Tangent (Tanh), Linear

Activation Function, and Softmax are the activation functions used. Hugging Face is an open-source package with several NLP applications as its focus. Compared to optimization, the testing process is compared with models that don't require tuning. These two models can be directly used for various NLP modeling needs [1]. This library's front-end API is implemented in Python, using the Scikit-learn algorithm, a freely accessible machine learning library for the Python programming language. It has many features, such as data processing, classification, regression, and clustering algorithms, including model evaluation designed to work with Python NumPy and other numerical and scientific libraries [12].

2.8. Loss Calculation

A multiclass classification model with a total of 14 classes was developed in this research. The appropriate method for calculating loss is categorical cross entropy because it can measure the difference between two probability distributions. Normalization of the prediction result vector should be prioritized to ensure that the activation function softmax reaches a cumulative probability of 1. In classification scenarios, the widely used cross-entropy loss function, also known as logarithmic, log, or logistic loss, compares predicted class probability values with actual values denoted by 0 or 1. The logarithmic penalty provides a higher score for significant differences close to 1 and lower for minor differences approaching 0 [13]. The mathematical formulation of cross entropy is expressed in the following equation.

$$L = -\sum_{i=1}^{M} t_i \log(p_i) \tag{1}$$

Where M is the number of classes, ti is the actual value of the class, and pi is the predicted value of the probability of the ith class. Cross-entropy calculations vary based on the classification problem, distinguishing between binary and multiclass scenarios. Specifically, Binary Cross-Entropy is a loss function designed for binary classification tasks, handling scenarios with only two choices. [14].

2.8.1. Model Evaluation Design

The Confusion Matrix method is used to measure the text classification performance model. This matrix is specifically designed for the proposed model, accommodating 14 classes consisting of the same number of columns and rows [15], with categories Staying in the Bus Stop Lane, Stopping at Zebra Crossing, Turning on Headlights, Pedestrian Lane, No Passing, No U-Turns, Traffic Signs Ahead, Using Cell Phone While Driving, Driving Against the Lane, Violating Traffic Signs, Not Wearing a Helmet, Driving Exceeding the Speed Limit, Underage Driver, Not Using a Seat Belt. The category class depicted in the confusion matrix for 14 classes is shown in Table 2.

Table 2. Confusion Matrix Design for 14 Classes[16]

Confusion Matrix		CORRECT Class													
		A	B	C	D	E	F	G	H	I	J	K	L	m	N
Predictable Class	A	F.A	F.A	F.A	F.A	F.A	F.A	F.A	F.A	F.A	F.A	F.A	F.A	F.A	F.A
	B	FB	FB	FB	FB	FB	FB	FB	FB	FB	FB	FB	FB	FB	FB
	C	FC	FC	FC	FC	FC	FC	FC	FC	FC	FC	FC	FC	FC	FC
	D	FD	FD	FD	FD	FD	FD	FD	FD	FD	FD	FD	FD	FD	FD
	E	Fe	Fe	Fe	Fe	Fe	Fe	Fe	Fe	Fe	Fe	Fe	Fe	Fe	Fe
	F	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
	G	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
	H	F.H	F.H	F.H	F.H	F.H	F.H	F.H	F.H	F.H	F.H	F.H	F.H	F.H	F.H
	I	FI	FI	FI	FI	FI	FI	FI	FI	FI	FI	FI	FI	FI	FI
	J	F.J	F.J	F.J	F.J	F.J	F.J	F.J	F.J	F.J	F.J	F.J	F.J	F.J	F.J
	K	FK	FK	FK	FK	FK	FK	FK	FK	FK	FK	FK	FK	FK	FK
	L	FL	FL	FL	FL	FL	FL	FL	FL	FL	FL	FL	FL	FL	FL
	m	FM	FM	FM	FM	FM	FM	FM	FM	FM	FM	FM	FM	FM	FM
	N	FN	FN	FN	FN	FN	FN	FN	FN	FN	FN	FN	FN	FN	FN

Confusion Matrix is a testing method used to evaluate the classification process and also serves as a basis for calculating metrics to assess model performance. This plays a vital role in helping to train the BERT Uncased-LSTM model and determine validation accuracy. Precision, recall, and F1 scores are important metrics in evaluating text classification models. These steps are invaluable in avoiding bias during calculations, especially when considering less balanced data [8].

3. Results and Discussion

The results of this research are categorized into dataset analysis, the performance of the BERT Uncased-LSTM model, and comparison with the perfected BERT Uncased.

3.1. Data Set Creation

Creating a dataset includes using the crawling method on the online media www.detik.com via Tweepy Library. This iterative method is carried out for specific categories, namely Staying in the lane, Low bridge, Using headlights, Crossing pedestrians, Prohibited crossing and turn return, Giving a signal to Then cross in front, Using a cell phone while driving, going against the flow of traffic, and breaking signs, not using a helmet, exceeding the speed limit, driving underage, and not using a seat belt [16]. The sample composition of the 73 datasets from 2000 data that have been created is shown in Figure 4.

NO.	TEXT CODE	DESCRIPTION	CLASS
1	0FCCA6568FCC 21	A car, a pickup and several motorcycles stop at a traffic light	Rider stops at a special stopping space
2	0FCCA6568FCC 22	Several motorcycle riders and four-wheeled vehicles stop at a traffic light	Rider stops at a special stopping space
3	0FCCA6568FCC 23	The riders are not wearing helmets	Rider does not wear a helmet
4	0FCCA6568FCC 24	Three motorcycle riders stop at a traffic light	Rider does not wear a helmet
5	0FCCA6568FCC 25	A motorcycle rider turns around	U-turning against traffic
6	0FCCA6568FCC 26	Four motorcycle riders and one car stop at a traffic light	Rider stops at a special stopping space
7	0FCCA6568FCC 27	Six motorcycle riders and four cars stop at a traffic light	Rider stops at a special stopping space
8	0FCCA6568FCC 28	Three motorcycle riders stop at a traffic light	Rider stops at a special stopping space
9	0FCCA6568FCC 29	Five motorcycle riders stop at a traffic light	Rider stops at a special stopping space
10	0FCCA6568FCC 30	A black motorcycle turns around and the passenger is not wearing a helmet	Rider does not wear a helmet
11	0FCCA6568FCC 31	Four motorcycle riders and three cars stop at a traffic light	Rider stops at a special stopping space
12	0FCCA6568FCC 32	Two pedestrians cross a zebra crossing	Rider stops at a zebra crossing
13	0FCCA6568FCC 33	A white motorcycle rider turns around	U-turning against traffic
14	0FCCA6568FCC 34	Several motorcycle riders and cars stop at a traffic light	Rider stops at a special stopping space
15	0FCCA6568FCC 35	A black motorcycle rider dressed in black turns around	Rider stops at a special stopping space
16	0FCCA6568FCC 36	One black motorcycle rider turns around	Rider stops at a special stopping space
17	0FCCA6568FCC 37	A black motorcycle rider stops right at the right stop line	Rider stops at a special stopping space
18	0FCCA6568FCC 38	Two motorcycle riders and two cars stop at a traffic light	Rider stops at a special stopping space
19	0FCCA6568FCC 39	Several motorcycle riders and two cars stop at traffic light	Rider stops at a special stopping space
20	0FCCA6568FCC 40	Black motorcycle passengers ride behind black car without wearing helmets	Rider stops at a special stopping space special
21	3A6AEAE4A13F 3	One black car and several motorcycles stop at traffic light	The rider stops at a special stopping space
22	3A6AEAE4A13F 4	Three four-wheeled vehicles and several motorcycles stop at traffic light	The rider stops at a special stopping space
23	3A6AEAE4A13F 5	Two motorcycles stop at traffic light	The rider stops at a special stopping space
24	3A6AEAE4A13F 6	Four cars and motorcycles and pickup stop at traffic light	The rider stops at a special stopping space
25	3A6AEAE4A13F 7	Several four-wheeled and two-wheeled vehicles stop at traffic light	The rider stops at a special stopping space
26	3A6AEAE4A13F 8	Traffic conditions are busy and smooth	The rider stops at a special stopping space
27	3A6AEAE4A13F	Four four-wheeled vehicles and several motorcycles stop at traffic light	The rider stops at a special stopping space
28	3A6AEAE4A13F10	Three pedestrians cross zebra crossing	The rider stops at a special stopping space
29	3A6AEAE4A13F	One black car and five motorcycles stop at traffic light	The rider stops at a special stopping space
30	4CA886995DE6 2	Two trucks and several motorcycle riders stop at traffic light	Turning in the opposite direction
31	4CA886995DE6 3	Several two-wheeled and four-wheeled vehicles stop at traffic light	Riding in one-way traffic
32	4CA886995DE6 4	The traffic situation is quite busy	Riding in a special stopping space
33	4CA886995DE6 5	Five motorcycle riders stop at traffic light	Riding not wearing a helmet
34	4CA886995DE6 6	A black car turns around	Riding in one-way traffic
35	4CA886995DE6 7	Three cars and seven motorcycles stop at traffic light	Riding in a special stopping space
36	4CA886995DE6 8	Three cars and several motorcycles stop at traffic light light.	Riding in a special stopping space
37	4CA886995DE6 9	Five motorcycles and one car stop at a traffic light stop	Riding in a special stopping space

Figure 4. Example Dataset Contains Traffic Conditions and Class Categories

System modeling specified two scenarios, the first being the original unmodified dataset. Meanwhile, the second scenario is a modified data set, which is achieved by eliminating words that contain valuable information in text classification, which in turn special target Name every category [17].

3.2. BERT Uncased-LSTM Model Results

3.2.1. Results on Unmodified Dataset

Based on the designed scenarios, the BERT Uncased-LSTM model was trained with ten scenarios. Each scenario combines three tuned hyperparameters: batch sizes of 32 and 64, learning rates of 2e-5 and 5e-5, and applying average and max pooling layers. [18]. This test aims to determine the influence of each hyperparameter in optimizing the BERT Uncased-LSTM model.

Table 3. Evaluation Results of BERT Uncased-LSTM Model Training for Unmodified Dataset

NO.	Group Size	Learning speed	Merger	Validation Accuracy
1	32	2e- 5	Average	98.00%
2	32	2e- 5	Max	98.10%
3	32	2e- 5	Average	97.56%
4	32	2e- 5	Max	97.40%
5	32	2e- 5	Average	98.00%
6	64	2e- 5	Max	97.20%
7	64	2e- 5	Average	97.15%
8	64	2e- 5	Max	96.25%
9	64	2e- 5	Average	97.20%
10	64	2e- 5	Max	97.00%

The validation accuracy results for each training scenario of the BERT Uncased-LSTM model are shown in Table 3. In the initial scenario, the model with batch size 32 and learning rate 2e-5 uses average pooling to achieve the highest value. Validation accuracy of 98.10%. Comparison of model training on scenarios 1 and 5 shows slightly higher validation accuracy for batch sizes of 32 versus 64, although the difference is not significant [8]. As the first scenario shows, smaller batch sizes result in lower generalization errors due to reduced noise and regularization effects. The difference in learning rate also affects model training results; for example, in scenarios 1 and 3, learning rate 2e-5 has higher validation accuracy than 5e-5. The results obtained imply that the output vector generated by the average pooling method effectively captures the entire sequence generated by the previous layer. Meanwhile, the max pooling method only selects the highest vector course, not represent the overall network [7].

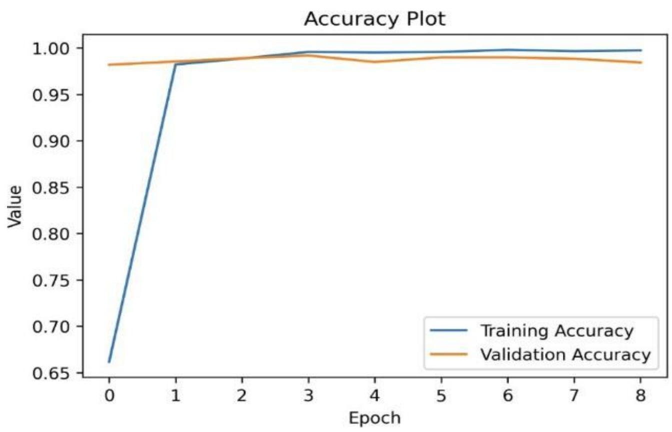

Figure 5. BERT Uncasded - Plot-Unmodified LSTM Model Training Accuracy Dataset

Figure 5 shows the training accuracy plot for the BERT Uncased-LSTM model on the unmodified dataset. This graph shows the optimal scenario, presenting the evolution of accuracy during the model training process. Additionally, the model achieved validation accuracy greater than 95% in the first epoch, peaking in the 4th epoch with a validation accuracy of 99.20%. Subsequent epochs failed to produce further accuracy improvements, leading to the termination of model training at the 9th epoch. The entire training duration for the BERT Uncased-LSTM model on the uncased dataset modified is about 30 minutes [19].

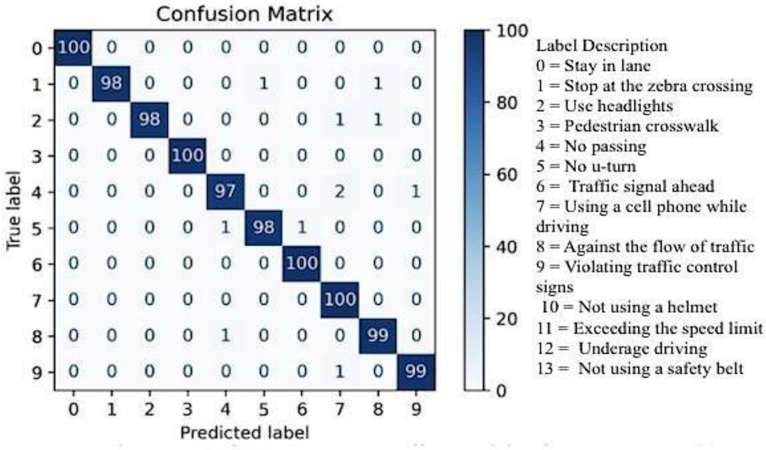

Figure 6. Testing Confusion Matrix BERT Uncased-LSTM Model-Unmodified Dataset

After achieving an impressive validation accuracy of 99.20% in the best-case scenario for the BERT Uncased-LSTM model, further testing was performed using previously unseen data to evaluate predictive performance. The results were used to develop a Confusion Matrix, as shown in Figure 6 shows the model's ability to accurately classify test data in various classes [5]. Meanwhile, four classes were predicted with 100% accuracy, and the rest experienced a slight error. The class with the highest error rate was identified as No-Go, with erroneous predictions assigned to the No-Turn class twice. This prediction error tends to occur because there are words that contextually have the same meaning but are used in several classes. Analyzing the Confusion Matrix, the average values of macro precision, recall, and F1 score were calculated as 98.25%, 97.90%, and 98.10%, respectively. The results demonstrate the commendable performance of the BERT Uncased-LSTM model in classifying unmodified datasets.

3.3. Calculation of Precision, Recall, and F1-Score Values

Precision, Recall, and F1-Score calculation results with Confusion Matrix for text classification with a modified dataset. Confusion Matrix is a performance measurement that is often used in classification problems, and its output consists of two or more classes. It consists of four attributes, a combination of predicted values and actual values: True Positive (TP) and False Positive (FP); this represents data incorrectly identified as positive in the predicted category but negative in the true category. c) True Negative (TN): This indicates data accurately identified as negative in predicted and actual categories. d) False Negative (FN): It indicates data that is incorrectly identified as negative in the predicted category but positive in the true category [24]. True Negative (TN): This indicates data accurately identified as negative in predicted and actual categories. False Negative (FN): It indicates data incorrectly identified as negative in the predicted category but positive in the true category [24]. These four attributes are the basis for calculating several evaluation metrics, namely Accuracy, Precision, Recall, and F1-Score [15].

3.3.1. Accuracy

Accuracy, calculated as the ratio of correct predictions (including positive and negative outcomes) over the entire data, is a commonly used metric because of its simplicity. However, this metric has the disadvantage that it is less reliable for imbalanced data. The accuracy value can be obtained using the following equation.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{9}$$

LONTAR KOMPUTER VOL. 15, NO. 2 AUGUST 2024 **p-ISSN 2088-1541**
DOI : 10.24843/LKJITI.2024.v15.i02.p04 **e-ISSN 2541-5832**
Accredited Sinta 2 by RISTEKDIKTI Decree No. 158/E/KPT/2021

3.3.2. Precision

Precision is the ratio of TP to all data that is predicted to be positive, focusing on minimizing FP. The precision value can be determined using Equation (10):

$$\text{Precision} = \frac{TP}{TP+TN} \tag{10}$$

3.3.3. Remember

Recall, an important metric, determines the number of positive cases accurately predicted by the proposed model. The significance becomes clear when FN is more important than FP. Calculated as the ratio of TP to all true positive cases, recall focuses on minimizing the occurrence of FN. However, the recall value can be obtained using Equation (11).

$$\text{Remember} = \frac{TP}{TP+TN} \tag{10}$$

3.3.4. F1 Score

The F1 Score is the harmonic average of Precision and Recall, providing a unified measure that summarizes both metrics. Moreover, the maximum value is achieved when these two metrics are equal. The F1 score is calculated as the harmonic mean of precision and recall, with mark specific determined by Equation 11[20].

$$\text{F1 Score} = \frac{2}{\frac{1}{Recall}+\frac{1}{Precission}} \tag{11}$$

3.3.5. Results on Modified Dataset

In training experiments using an unmodified dataset, the Caseless BERT-LSTM model demonstrated efficient classification, exploiting the prevalence of words with valuable information in almost every tweet. These informative words correspond to the name of each category, making it easier for the model to carry out the classification process. To evaluate adaptability and performance on more challenging datasets with increased similarity between classes, additional experiments were conducted to modify the dataset by removing such important words.

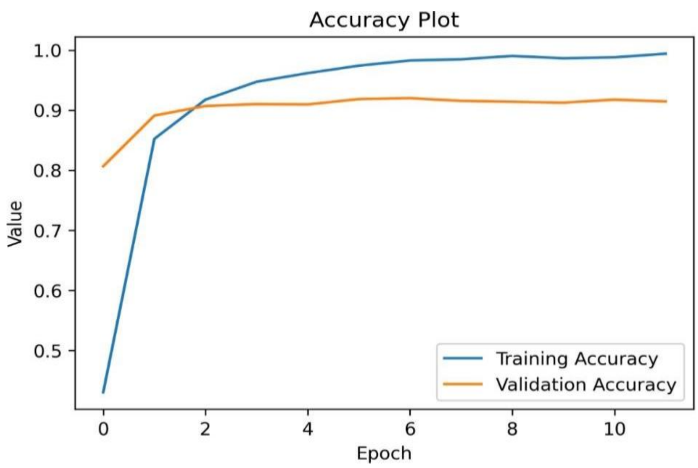

Figure 7. Modified Dataset Plots Training Accuracy of BERT Uncased -Aligned Models

The caseless BERT-LSTM model, trained on a modified dataset with a validation accuracy of 9.35%, is saved for testing by making predictions on the modified, invisible test data. The prediction results developed the confusion matrix shown in Figure 7. Despite specific errors, the model could classify the test data into various classes. It can be seen that the model can classify test data into each class quite well, even though it still experiences some errors. The class with the most errors is class 2, which predicted " The driver stops at *the zebra crossing* " with a score of 8 1 . Another class with quite a lot of errors is class 7, with the *text prediction* " Violating traffic

LONTAR KOMPUTER VOL. 15, NO. 2 AUGUST 2024 **p-ISSN 2088-1541**
DOI : 10.24843/LKJITI.2024.v15.i02.p04 **e-ISSN 2541-5832**
Accredited Sinta 2 by RISTEKDIKTI Decree No. 158/E/KPT/2021

signals " with a score of 8 1 . This is because the dataset is modified so that it has a different context from the reference sentence. The confusion matrix calculated the average macro precision, recall, and F1 scores as 92.45%, 92.70%, and 92.15%, respectively. Regardless of the modifications made to the data set, the refined BERT Uncased-LSTM model obtained values above 90 % for the third metric [16].

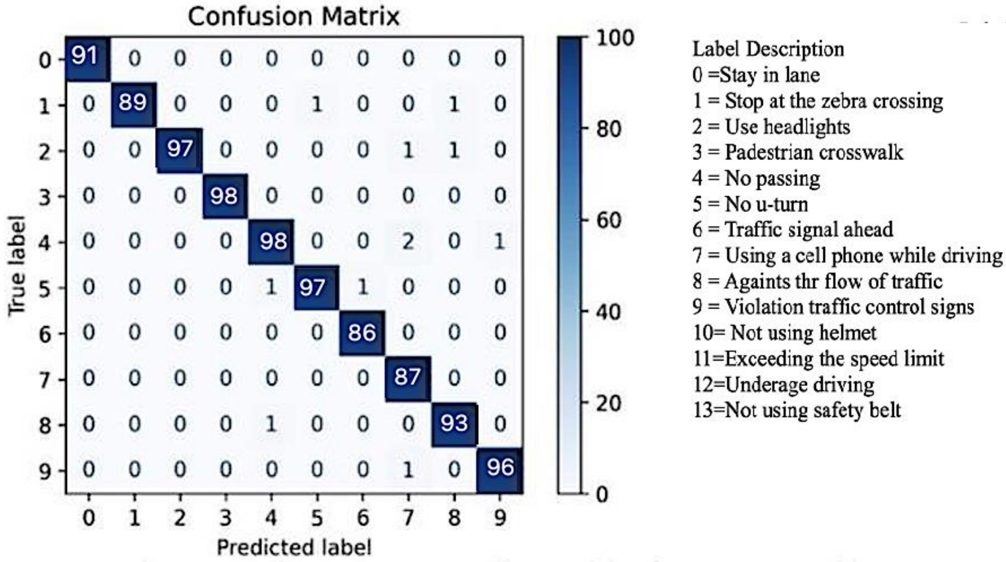

Figure 8 . Confusion Matrix Testing of Modified Data Sets of BERT Models Without Well-Resolved Casings

F1 score computational findings show that the BERT Uncased-LSTM model improves by 0.59% and 0.79% on the adjusted and unmodified datasets. The table shows that the unmodified BERT Uncased-LSTM model produces fewer prediction errors than the modified model. This decrease in inaccuracy is visible in classes that are contextually similar to the actual class but inconsistent with other classes [21].

Table 4. Evaluation Results of Testing Two Text Classification Models

Dataset	Model	Accuracy	Precision	Recall	F1 Score	Period	Time Training
Not modified	BERT Uncased - L	98.10%	97, 25%	96, 9 0 %	98.10%	11	± 30 minutes
Modification	BERT Uncased - LSTM	93.50%	92.45%	92.70%	92.15%	15	± 45 minutes

The BERT Uncased-LSTM model improves by 0.59% and 0.79% on the adjusted and unmodified datasets, respectively, according to the F1 score calculation. Compared with the modified model, Table 4 shows that the unmodified BERT Uncased-LSTM model produces fewer prediction errors in the classes closest to the true class. The pattern of prediction errors shows that both models concentrate on the same classes when comparing the confusion matrix findings on the adjusted. [11] dataset. Table 5 compares the accuracy between BERT Uncased-LSTM and Word2Vec.

Table 5. Evaluation Results Testing BERT Uncased-LSTM and Word2Vec Models

Model	Accuracy	F1 Score	Period	Time Training
BERT Uncased- LSTM	98.10%	98.10%	11	± 30 minutes
Word2Vec	85.96%	90.20 %	15	± 4 0 minutes

From the comparison with a classification development model, existing text done previously using Word2Vec and LSTM produced an accuracy of 85.96%. Several studies identified news-related articles using search methods categorized into 14 classes and word refinement from the Keras library with CNN architecture and achieved an F1 score of 90.2% [5]. Previous research focused on classifying Indonesian online news based on four popular topics using Word2Vec [6]. The research succeeded in developing a text classification model based on binary and multiclass classification. By applying a pre-trained word repair model with state-of-the-art performance, the developed model has the potential to improve accuracy and can be combined with neural network architectures. Therefore, this research proposes a text classification model that combines the BERT Uncased model, which was previously trained with one of the RNN architectures, namely LSTM, to classify traffic violation news into several categories according to the context [7].

4. Conclusion

Using the BERT-LSTM approach to classify traffic violation news content into several classes, this research separates text classification models on modified and unmodified datasets. The comparison shows that combining the pre-trained LSTM model and BERT Uncased results in better text categorization. The original dataset achieved an F1 score of 98.10% with the optimal hyperparameter combination scenario (batch size 16, learning rate 2e-5, use of mean pooling). Additionally, the BERT Uncased-LSTM model had a significantly shorter overall training time (approximately 15 minutes) for both databases when using the unmodified dataset compared to the modified dataset. The research results show that the test value on the unmodified dataset is higher than the modified dataset. The higher volume is caused by selecting words with high information values in the modified dataset, making it difficult for the model to understand the context in text classification.

Reference

[1] V. Singh , V. Unadkat , and P. Kanani , “Intelligent traffic management system” *International Journal of Recent Technology and Engineering*, vol. 8, no. 3, pp. 7592–7597, 2019, doi: 10.35940/ijrte.C6168.098319.

[2] S. Bai *et al.*, “Natural language guided visual relationship detection” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp . 444–453, 2019, doi 10.1109/CVPRW.2019.00058.

[3] D. Guna Mandhasiya , H. Murfi, A. Bustamam , P. Anki , and H. Standard RIS Vancouver Mandasiya , “Evaluation of Machine Performance Based Learning on BERT Data Representation with LSTM Model to Conduct Sentiment Analysis in Indonesian for Predicting Voices of Social Media Users in the 2024 Indonesia Presidential Election” in *2022 5th International Conference on Information and Communications Technology (ICOIACT)*, pp. 441–446, 2022, doi: 10.1109/ICOIACT55506.2022.9972206.

[4] J. Li, P. Yao , L. Guo , and W. Zhang , “Boosted transformer for image captioning” *Applied Sciences (Switzerland)*, vol. 9, no. 16, pp . 1–15, 2019, doi: 10.3390/app9163260.

[5] S. Ren, K. He, R. Girshick , and J. Sun, “Faster R-CNN: Towards Real-Time Objects Detection with Region Proposal Networks” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp . 1137–1149, 2017, doi: 10.1109/TPAMI.2016.2577031.

[6] AK Sharma, S. Chaurasia , and DK Srivastava , “Sentimental Short Sentences Classification by Using CNN Deep Learning Model with Fine Tuned Word2Vec” in *Procedia Computers Science* , Elsevier B.V., 2020, pp . 1139–1147. doi: 10.1016/j.procs.2020.03.416.

[7] QT Nguyen , TL Nguyen , NH Luong , and QH Ngo , “Fine-Tuning BERT for Sentiment Analysis of Vietnamese Reviews” in *2020 7th NAFOSTED Conference on Information and Computer Science (NICS)*, Nov. 2020, doi: 10.1109/NICS51282.2020.9335899, [Online]. Available : http://arxiv.org/abs/2011.10426
[8] Q. Yu , Z. Wang, and K. Jiang , “Research on Text Classification Based on BERT- BiGRU Model” in *Journal of Physics : Conference Series* , IOP Publishing Ltd, Jan. 2021, doi: 10.1088/1742-6596/1746/1/012019.
[9] J. Li, P. Yao , L. Guo , and W. Zhang , “Boosted transformer for image captioning” *Applied Sciences(Switzerland)*, vol. 9, no. 16, pp . 1–15, 2019, doi: 10.3390/app9163260.
[10] M. Sundermeyer , R. Schlüter , and H. Ney , “LSTM neural networks for language modeling” in *13th Annual Conference of the International Speech Communications Association 2012, INTERSPEECH 2012* , 2012.
[11] N. Rai, D. Kumar, N. Kaushik , C. Raj, and A. Ali, “Fake News Classification using transformer based enhanced LSTM and BERT” *International Journal of Cognitive Computing in Engineering*, vol. 3, pp. 98–105, Jun. 2022, doi: 10.1016/j.ijcce.2022.03.003.
[12] A. Géron, “Hands-on Machine Learning with Scikit-Learn , Keras, and TensorFlow Concepts, Tools , and Techniques to Build Intelligent Systems”, Second ed., City: United State of America, O’Reilly Media, Inc., 2019.
[13] S. Minaee , N. Kalchbrenner , E. Cambria , N. Nikzad , M. Chenaghlu , and J. Gao , “Deep Learning Based Text Classification : A Comprehensive Review” *ACM Computing Surveys(CSUR)*, vol. 54, no. 3, pp . 1–43, 2020, doi: https://doi.org/10.1145/3439726, Available: http://arxiv.org/abs/2004.03705.
[14] C. Bircanoğlu, “A Comparison of Losses Functions in Deep embedding”. [Online]. Available : https://www.researchgate.net/publication/318588371
[15] R. He *et al.* , “Confusion Matrices and Rough Set Data Analysis” *Journal of Physics: Conference Series*, vol. 1229, 2019, doi: 10.1088/1742-6596/1229/1/012055.
[16] Z. Karimi , “Confusion Matrix”, 2021.
[17] F. Koto, A. Rahimi, JH Lau, and T. Baldwin , “IndoLEM and IndoBERT : A Benchmark Datasets and Pre-trained Language Model for Indonesian NLP” in *28th International Conference on Computational Linguistics*, Nov. 2020, doi: 10.18653/v1/2020.coling-main.66, [Online]. Available : http://arxiv.org/abs/2011.00677
[18] H. Gholamalinezhad and H. Khosravi ,” Pooling Methods in Deep Neural Networks , a Review”, arXiv.org, https://doi.org/10.48550/arXiv.2009.07485. [Online] Available: https://arxiv.org/pdf/2009.07485
[19] R. Cai *et al.*, “Sentiment analysis about investors and consumers in energy markets based on BERT-BILSTM” *IEEE Access*, vol. 8, pp. 171408–171415, 2020, doi: 10.1109/ACCESS.2020.3024750.
[20] SK Addagarla , “Real Time Multi-Scale Facials Mask Detection and Classification Using Deep Transfer Learning Techniques,” *International Journal of Advanced Trends in Computers Science and Engineering* , vol. 9, no. 4, pp . 4402–4408, Aug . 2020, doi: 10.30534/ ijatcse /2020/33942020.