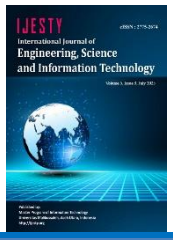
International Journal of Engineering, Science and Information Technology
Volume 5, No. 3 (2025) pp. 409-414
eISSN: 2775-2674
Website: http://ijesty.org/index.php/ijesty
DOI: https://doi.org/10.52088/ijesty.v5i3.1124
Research Paper, Short Communication, Review, Technical Paper



Sentiment Analysis of Vidio Application Based on Reviews on Google Play Store Using Bidirectional Encoder Representations from the Transformers Method

Rina Refianti*, Andrian Senjaya

Faculty of Computer Science and Information Technology, Gunadarma University, Jakarta, Indonesia

**Corresponding author Email: rina@staff.gunadarma.ac.id*

The manuscript was received on 10 January 2025, revised on 26 February 2025, and accepted on 25 May 2025, date of publication 2 July 2025

Abstract

Sentiment analysis is a computational study that aims to process, extract, summarise, and analyse the information contained in the text so it can conclude the emotions and points of view given by the author from the text and share the emotional tendencies in the text through the subjective information contained in it. Vidio is a video streaming site that allows users to watch and enjoy various videos and other services, such as live chat and playing games over the internet, and broadcast them by live streaming and video on demand. The analysis process uses the Bidirectional Encoder Representations from Transformers (BERT) method to classify comments into positive, neutral, and negative sentiments using the Python programming language, and based on the results of the tests that have been carried out from the amount of comment data—as much as 6000 data with training data as much as 4019 data, validation data as many as 1154 data, and test data as many as 569 data—an accuracy result of 76%.

Keywords: *BERT, Encoder, Sentiment Analysis, Transformers, Vidio.*

1. Introduction

Entertainment is a critical human need and cannot be separated from human life. Entertainment is a way for humans to forget all the problems they face in everyday life. One type of entertainment is watching movies and soap operas. Watching movies or soap operas is usually enjoyed on television. Films and soap operas are broadcast on television according to a predetermined schedule.

The rapid development of technology has forced several conventional fields to enter an era of disruption and switch to digital activities to make human life easier. Vidio is an online portal or video streaming website established in 2014 [1]. Vidio allows users to watch and enjoy various videos by streaming (live streaming and video on demand) and other services such as live chat and playing games through the internet. Over time, Vidio can also be accessed via mobile and tablet devices (iOS, Android), personal computers, Chromecast, set-top boxes, smart TVs, and other devices with the Vidio [1] application installed. Movies and soap operas available on the Vidio application can be watched anytime and anywhere.

More than four hundred and eighty thousand Android users have downloaded the Vidio application on the Google Play Store, and there are many opinions regarding the Vidio application on the Google Play Store. A sentiment analysis is needed to determine the user's opinion about the Vidio application. The task of sentiment analysis is to understand, extract opinion data, and process textual data automatically to get a sentiment contained in an opinion [2].

Previous research related to sentiment analysis on the streaming application [3] yielded an accuracy of 92.67 % for the Iflix application, the second Netflix was 82.33%, and the third Disney Hotstar was 69.33 %.

In this study, the method used for sentiment analysis is the bidirectional encoder representations from Transformers (BERT) method, which is used to analyse sentiment through user comments on the Google Play Store. The researcher chose the BERT method because it is new and rarely used. Still, it produces a high percentage of accuracy [4]. Using the BERT method produces an accuracy of 73 % with 2000 reviews, of which 1000 reviews have positive sentiment and 1000 reviews have negative sentiment. The BERT method uses the Python programming language. These sentiments will be categorised into positive, negative, and neutral.

2. Methods

The research steps used in the Vidio application sentiment analysis using the BERT method consist of several steps, as shown in Figure 1.
The research steps below show that sentiment analysis begins by scraping or collecting data through the Google Play website. The results of scraping are then collected and become a dataset. The dataset is then labelled with positive, neutral, and negative labels. The dataset then enters the pre-processing stage. In general, data pre-processing is done by eliminating inappropriate data or converting data into a form that is easier for the system to process. Several pre-processing stages include case folding, data cleaning, tokenising, and normalisation. After going through these processes, the dataset was trained using IndoBERT to be classified into three categories: positive, neutral, and hostile. The classified data is then evaluated to see the accuracy of the predictions.

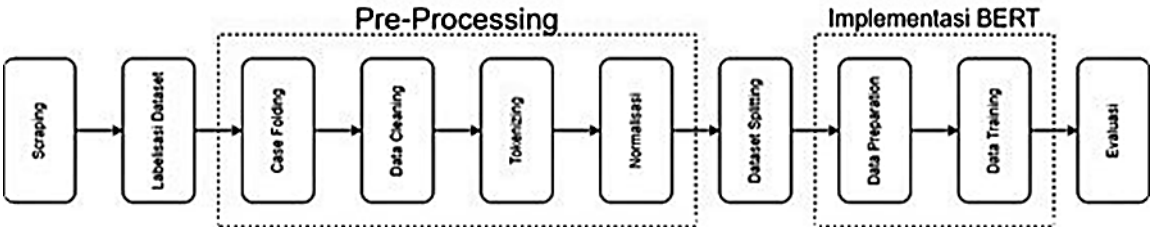


Fig 1. Research steps

2.1. Scraping

This study uses a dataset from user reviews on the Vidio application from the Google Play Store. Scraping is done by utilising the Google Play scraper library provided by pip. Pip is a package management tool that installs, removes, and upgrades libraries in Python. The Google-play-scraper function extracts various attributes on application reviews on the Google Play Store, such as review text, reviewer username, star rating, date the review was created, etc. The dataset successfully extracted from the Google Play Store website was 6000 reviews. The selection feature based on stars on the Google Play Scraper is used to get a balanced dataset between sentiments. Then, feature selection will be performed on the dataset. Feature selection is done by removing the id, username, and date columns because they are deemed unnecessary for the sentiment analysis process. The dataset is then saved in tab-separated value (.tsv) format. The TSV format was chosen because it makes it easier to analyse the scraping results. After all, if you use a more common format, such as comma-separated values (CSV), the scraping results are separated using commas, so there are quite a few commas in the sentence.

2.2. Dataset Labelling

Sentiment analysis using the supervised learning method requires datasets with labels. This labelling must be done because the supervised learning method requires studying examples. The essence of supervised learning is to create a mechanism whereby the model can see examples and generate generalisations so that the model's output is a prediction that matches the desired label [5]. Models can also see and learn how reviews have positive (4 and 5 stars), neutral (3 stars), and negative (1 and 2 stars) sentiments.

2.3. Pre-processing

In general, data pre-processing is done by eliminating inappropriate data or converting data into a form that is easier for the system to process. Several steps in pre-processing, namely case folding, data cleaning, tokenising, and normalisation.

1. **Case Folding**: This step converts the entire text in the document into a standard form (in this case lowercase letters). This step is carried out because not all text documents are consistent in the use of capital letters. Case folding is done using the lower() function which is available in the Python library.
2. **Data Cleaning**: This step removes data attributes that are not needed in the classification process. The data obtained from the dataset collection will be filtered to produce the data that is really needed. Unnecessary data can reduce the quality or accuracy of the classification results. The processes carried out in the data cleaning stage are the removal of numbers, punctuation, emoji, excess spaces, duplicate sentences, stop word sand stemming.
3. **Tokenising**: This step uses a regular expression to find the character to be deleted. This stage is used to break down sentences into word lists. This process uses the word tokenise function provided by the NLTK library.
4. **Normalisation**: This step changes non-standard words to standard words so that the data can be fully recognised by the system. The normalisation process can use the library that has been provided or by uploading a file that contains a list of standard and non-standard words.

2.4. Dataset Splitting

Dataset Splitting is a technique used to see model performance by dividing the data to be processed into several parts, in this case, training, validation, and testing. The training dataset is used to train the model, the validation dataset is used to minimise overfitting that often occurs in artificial neural networks, while the testing dataset itself is used as the final test to see the accuracy of the network that has been trained with the training dataset. This study's proportion of split datasets is 70 % train set, 20 % validation set, and 10 % test set.

2.5. BERT Implementation

This study uses a fine-tuning technique with the IndoBERT base-p1 model, one of the models that uses the BERT-base architecture. This technique uses a model that has been trained previously and only teaches a little more to reach the optimal point on a new task. This model has been trained using 4 billion words with around 250 million formal and colloquial sentences in citeref6 Language. This study uses the Transformers library provided by HuggingFace [7]. This library offers thousands of pre-trained models that can be used to perform classification, information extraction, debriefing, summarisation, translation, text generation, and other tasks in 100 languages. Two leading deep learning libraries, PyTorch and Tensor Flow, support transformers.

1. **BERT Pre-Training**: The BERT pre-trained model used in this study is IndoBERT, an architecture specifically trained using Indonesian corpus data. The dataset used reaches 4 billion words, both informal and formal languages, with 12 different Indonesian corpora. This dataset is then trained with the standard BERT architecture, which has 12 transformer layers [8, 9, 10].
2. **Fine-Tuning BERT**: Training is done by using a model that has been trained previously and then learning a little more to reach the optimal point. This training technique is known as fine-tuning. This study used the indobenchmark/indobert-base-p1 pre-trained model from IndoNLU [6].

Pre-trained models from IndoNLU are accessed via Hugging Face. Fine-tuning for this research uses the BertForSequenceClassification model class from HuggingFace's transformers library.
Training is obtained from the previous data sharing process: training, validation, and test data. The training data is loaded according to the predefined hyperparameter tuning, namely a batch size of 32. The batch size is the number of samples entered into the network before the weight is adjusted. The number of workers is 16, and the maximum sequence length is 512. During training, loss and accuracy of training data and validation data will be monitored and validated for each epoch. The epoch is the number of times the network sees the entire dataset, and the epoch used is 5. The training process will be run using Google Colab. The learning rate determines how much the neural network will change, and trains the data using a learning rate of 3e-6. In the previous step, the optimal parameter search results were obtained. These search results help adjust parameter settings.

2.6. Evaluation

The evaluation step is intended to see the results of predictions against the original dataset. A confusion matrix is used to conduct the evaluation, as shown in Table 1.

Table 1. Confusion Matrix (CM)

True Class	Predicted Class		
	Positive	Neutral	Negative
Positive	True Positive (TF)	False Neutral (FNt)	False Negative (FN)
Neutral	False Positive (FP)	True Neutral (TNt)	False Negative (FN)
Negative	False Positive (FP)	False Neutral (FNt)	True Negative (TN)

The categories in the confusion matrix consist of six categories, namely True Negative (TN), False Negative (FN), True Neutral (TNt), False Neutral (FNt), True Positive (TP) and False Positive (FP). True Negative (TN) are sentences that have negative sentiments, and the predicted results also show negative results. False Negative (FN) is a sentence with negative sentiment, but the prediction results show neutral or positive sentiment. True Neutral (TNt) is a sentence with a neutral sentiment, and the predicted results also show a neutral sentiment. False Neutral (FNt) is a sentence with neutral sentiment, but the prediction results show positive or negative sentiment. True Positive (TP) is a sentence with positive sentiment, and the predicted results also show positive sentiment. False Positive (FP) is a sentence that has a positive sentiment, but the predicted results show a neutral or negative sentiment.
After getting the confusion matrix values, precision, recall, accuracy, and F1-Score values can also be obtained as follows:
Precision is used to measure the frequency of correct answers or predictions from a fact and shows the quality of successful predictions from the applied system. ***Precision*** can be formulated as follows

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

Recall is used to measure the frequency of how many times a specified category is detected and shows the quality to indicate each prediction. ***Recall*** can be formulated as follows:

$$Recall = \frac{TP}{TP+FNt+FN} \quad (2)$$

Accuracy is the division of all correct predictions. ***Accuracy*** can be formulated as follows:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

F1-Score calculates the harmony between precision and recall. ***F1-Score*** can be formulated as follows:

$$F1-Score = 2 * \frac{Recall*Precision}{Recall+Precision} \quad (4)$$

3. Results and Discussion

3.1. Data Scraping Implementation

Scraping is done to get reviews in the Ratings and Reviews column of the Vidio application on the Google Play Store. These reviews will become the dataset used to analyse sentiment towards the Vidio application. Reviews are taken using the Google Play Scraper library, and the scraping process is done through Google Collab. This research only takes the contents of the review text and score columns. The username and at columns are removed because they are not needed in the sentiment analysis process. The star-based selection feature is used to obtain a balanced dataset.

3.2. Dataset Labelling

The dataset in this study must be labeled first. Labeling is done by giving a negative label to data with stars 1 and 2, a neutral label to data with 3 stars, and a positive label to data with 4 and 5 stars. Labeling is done using the vlookup function in Microsoft Excel

3.3. Embedding Model

The pre-processing stage in this study consists of several processes, namely case folding, data cleaning, tokenising, and normalisation. The pre-processing results can be seen in Table 2.

Table 2. Pre-processing results

Index	Review_text	Category
2829	payment method using credit for a subscription	neutral
1021	pay platinum package via XL credit, try again, failed, complicated	negative
1365	payment process	negative
316	Good	negative
1581	wrong live	negative
443	watching movies	negative
552	annoying ads	negative
706	bug	negative
3438	trial	neutral
4713	nice watch video, watch ball, give star, give free watch ball	positive

3.4. Dataset Splitting

This study's proportion of split datasets is 70 % train set, 20 % validation set, and 10 % test set. The dataset splitting process can be seen in Figure 3.

```
[45] # train val split
     train_set, val_set = train_test_split(df_v2, test_size=0.3, stratify=df_v2.category, random_state=1)
     val_set, test_set = train_test_split(val_set, test_size=0.33, stratify=val_set.category, random_state=1)

[46] print(f'Train shape: {train_set.shape}')
     print(f'Val shape: {val_set.shape}')
     print(f'Test shape: {test_set.shape}')

     Train shape: (4019, 2)
     Val shape: (1154, 2)
     Test shape: (569, 2)
```

Fig 2. Dataset Splitting Process

3.5. BERT Implementation

Load Models. Training is not done from scratch, but by using a model that has been previously trained and only learns a little more to reach the optimal point on the new task. This training technique is known as fine-tuning. In this way, there is no need to do training from scratch, but we can download pre-trained models. This research will use the pre-trained model indobenchmark/indobert-base-p1 from IndoNLU.

IndoNLU's pre-trained models are accessed through the Hugging Face platform. Fine-tuned this sentiment analysis by using the BertForSequenceClassification model class from HuggingFace's transformers library.

Initialize DocumentSentimentDataset and DocumentSentimentDataLoader Classes. In the DocumentSentimentDataLoader class instance parameter, there are several other parameters besides max seq len. The dataset parameter is required for the DataLoader class and will be used as the data source. Batch size indicates the amount of data to be retrieved in each batch, num workers suggests the number of processes that will be used to retrieve data in parallel, and shuffle indicates how to retrieve data sequentially or randomly.

Training. This study uses fine-tuning techniques. The fine-tuning process is carried out in several stages, namely defining the Adam optimiser with a small learning rate, usually below 1e-3. Call model.train() to enable the dropout regularisation layer on the model. Call torch.set grad enabled(True) to enable autograd compute. Iterate over the train loader data loader object to fetch batch data and pass it to the forward sequence classification function on the model. Calculate the gradient automatically by calling the loss.backward() function. This function is derived from the autograd package. Perform gradient calculations by calling the optimiser.step().

```
(Epoch 1) TRAIN LOSS:0.8265 LR:0.00000300: 100%|██████████| 126/126 [00:25<00:00,  4.88it/s]
(Epoch 1) TRAIN LOSS:0.8265 ACC:0.64 F1:0.64 REC:0.64 PRE:0.64 LR:0.00000300
  0%|          | 0/37 [00:00<?, ?it/s]/usr/local/lib/python3.7/dist-packages/torch/utils/data/dataloa
  cpuset_checked))
VALID LOSS:0.6734 ACC:0.72 F1:0.72 REC:0.73 PRE:0.73: 100%|██████████| 37/37 [00:03<00:00, 11.25it/s]
(Epoch 1) VALID LOSS:0.6734 ACC:0.72 F1:0.72 REC:0.73 PRE:0.73
  0%|          | 0/126 [00:00<?, ?it/s]/usr/local/lib/python3.7/dist-packages/torch/utils/data/datalo
  cpuset_checked))
(Epoch 2) TRAIN LOSS:0.6544 LR:0.00000300: 100%|██████████| 126/126 [00:23<00:00,  5.25it/s]
(Epoch 2) TRAIN LOSS:0.6544 ACC:0.73 F1:0.73 REC:0.73 PRE:0.74 LR:0.00000300
  0%|          | 0/37 [00:00<?, ?it/s]/usr/local/lib/python3.7/dist-packages/torch/utils/data/dataloa
  cpuset_checked))
VALID LOSS:0.6224 ACC:0.75 F1:0.75 REC:0.75 PRE:0.75: 100%|██████████| 37/37 [00:03<00:00, 10.92it/s]
(Epoch 2) VALID LOSS:0.6224 ACC:0.75 F1:0.75 REC:0.75 PRE:0.75
  0%|          | 0/126 [00:00<?, ?it/s]/usr/local/lib/python3.7/dist-packages/torch/utils/data/datalo
  cpuset_checked))
(Epoch 3) TRAIN LOSS:0.5971 LR:0.00000300: 100%|██████████| 126/126 [00:24<00:00,  5.13it/s]
(Epoch 3) TRAIN LOSS:0.5971 ACC:0.77 F1:0.77 REC:0.77 PRE:0.77 LR:0.00000300
  0%|          | 0/37 [00:00<?, ?it/s]/usr/local/lib/python3.7/dist-packages/torch/utils/data/dataloa
  cpuset_checked))
VALID LOSS:0.6247 ACC:0.74 F1:0.74 REC:0.74 PRE:0.74: 100%|██████████| 37/37 [00:03<00:00, 10.69it/s]
(Epoch 3) VALID LOSS:0.6247 ACC:0.74 F1:0.74 REC:0.74 PRE:0.74
  0%|          | 0/126 [00:00<?, ?it/s]/usr/local/lib/python3.7/dist-packages/torch/utils/data/datalo
  cpuset_checked))
(Epoch 4) TRAIN LOSS:0.5544 LR:0.00000300: 100%|██████████| 126/126 [00:25<00:00,  4.91it/s]
(Epoch 4) TRAIN LOSS:0.5544 ACC:0.78 F1:0.78 REC:0.78 PRE:0.78 LR:0.00000300
  0%|          | 0/37 [00:00<?, ?it/s]/usr/local/lib/python3.7/dist-packages/torch/utils/data/dataloa
  cpuset_checked))
VALID LOSS:0.6147 ACC:0.75 F1:0.75 REC:0.75 PRE:0.75: 100%|██████████| 37/37 [00:03<00:00, 10.39it/s]
(Epoch 4) VALID LOSS:0.6147 ACC:0.75 F1:0.75 REC:0.75 PRE:0.75
  0%|          | 0/126 [00:00<?, ?it/s]/usr/local/lib/python3.7/dist-packages/torch/utils/data/datalo
  cpuset_checked))
(Epoch 5) TRAIN LOSS:0.5188 LR:0.00000300: 100%|██████████| 126/126 [00:25<00:00,  4.94it/s]
(Epoch 5) TRAIN LOSS:0.5188 ACC:0.80 F1:0.80 REC:0.80 PRE:0.81 LR:0.00000300
  0%|          | 0/37 [00:00<?, ?it/s]/usr/local/lib/python3.7/dist-packages/torch/utils/data/dataloa
  cpuset_checked))
VALID LOSS:0.6167 ACC:0.75 F1:0.75 REC:0.75 PRE:0.75: 100%|██████████| 37/37 [00:03<00:00, 10.68it/s]
```

Fig 3. Training Process

From the results of observations of these experiments, it was found that the accuracy during training was better than the results during validation. The curve shows an increase in accurate results obtained during training. However, for validation, the results are lower.

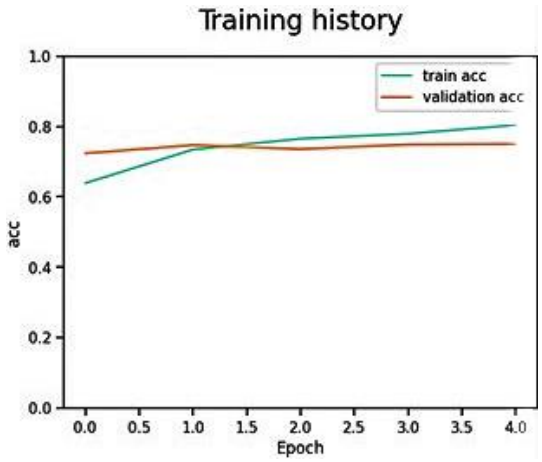


Fig 4. Training Result Curve

3.6. Evaluation

The final step is evaluation. This stage will discuss the results of the training that has been carried out. Results are shown with a confusion matrix to measure how much the model succeeds in predicting sentiment.

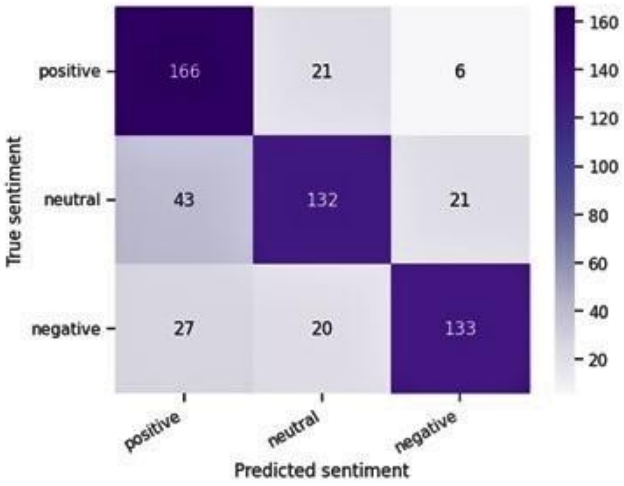


Fig 5. Confusion matrix

From the confusion matrix above, the model is quite good at predicting positive, neutral, and negative sentiments. After getting the results in a confusion matrix, accuracy, precision, recall, and F1-Score calculations can be performed using the classification report() function from sklearn.

	precision	recall	f1-score	support
positive	0.70	0.86	0.77	193
neutral	0.76	0.67	0.72	196
negative	0.83	0.74	0.78	180
accuracy			0.76	569
macro avg	0.77	0.76	0.76	569
weighted avg	0.76	0.76	0.76	569

Fig 6. Classification Report

From the results of the Classification Report, the accuracy of the BERT method in predicting data testing is 76 %. The precision level for the positive class is 70 %, the precision level for the neutral class is 76%, and the negative class's precision level is 83 %. The recall value obtained from the positive class is 86 %, the recall value obtained from the neutral class is 67%, while the recall value obtained from the negative class is 74 %. The F1-Score value obtained from the positive class is 77%, the F1-Score value obtained from the neutral class is 72%, while the F1-Score value obtained from the negative class is 78 %.

4. Conclusion

Google Play Store sentiment analysis research on the Vidio application has been successfully conducted. The analysis process was carried out using the Bidirectional Encoder Representations from Transformers (BERT) method to classify comments into positive, neutral, and negative sentiments using the Python programming language. Based on the results of the tests carried out from the total comment data of 6000 data with training data of 4019 data, validation data of 1154 data, and test data of 569 data, an accuracy of 76 % was obtained.

Acknowledgement

The authors would like to thank the Gunadarma Education Foundation for financial support.

References

[1] A. Nurdin, B.A.S. Aji, A. Bustamin, Z. Abidin, "Perbandingan Kinerja Word Embedding Word2vec, Glove, Dan Fasttext Pada Klasifikasi Teks," Jurnal Teknokompak, vol. 14, no.2, pp. 74-79, 2020.
[2] M.F. Al-Shufi, dan A. Erfina, "Sentimen Analisis Mengenai Aplikasi Streaming Film Menggunakan Algoritma Support Vector Machine Di Play Store," Sismatik, vol. 1, pp. 156-162, 2021.
[3] Y. Goldberg, "Neural network methods for natural language processing," Synthesis lectures on human language technologies, vol. 10, pp. 1-309, 2017.
[4] D. Fimoza, "Analisis Sentimen Terhadap Film Indonesia dengan Pendekatan BERT," Undergraduate Papers, Universitas Sumatera Utara, 2021
[5] C.A. Putri, "Analisis Sentimen Review Film Berbahasa Inggris Dengan Pendekatan Bidirectional Encoder Representations from Transformers," Jurnal Teknik Informatika dan Sistem Informasi, vol. 6, pp. 181-193, 2020.
[6] H.K. Putra, "Deteksi Penggunaan Kalimat Abusive Pada Teks Bahasa Indonesia Menggunakan Metode IndoBERT, Universitas Telkom, 2021.
[7] E.A. Putra, "Implementasi BERT untuk Analisis Sentimen Terhadap Ulasan Aplikasi FLIP Berbahasa Indonesia,", 2021.
[8] F.V. Sari dan A. Wibowo, "Analisis Sentimen Pelanggan Toko Online Jd. Id Menggunakan Metode Naive Bayes Classifier Berbasis Konversi Ikon Emosi. Simetris, vol. 10, pp. 681-686, 2019.
[9] B. Wilie, et.al., "IndoNLU: Benchmark and Resources for Evaluating Indonesian Natural Language Understanding," https://arxiv.org/abs/2009.05387 , 2020.
[10] T. Wolf, et.a.l., "Transformers: State-of-the-art Natural Language Processing," In Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations, pp. 38-45, 2020.
[11] R. Refianti and N. Anggraeni, "Sentiment Analysis Using Convolutional Neural Network Method to Classify Reviews on Zoom Cloud Meetings Application Based on Reviews on Google Playstore," International Journal of Engineering, Science & Information Technology (IJESTY), Volume 3, No. 3, pp. 7-16, 2023.
[12] R. Refianti, A.B. Mutiara, and R.A. Putra, "A Lexicon-Based Long Short-Term Memory (LSTM) Model for Sentiment Analysis to Classify Halodoc Application Reviews on Google Playstore," Journal of Applied Data Sciences, Vol. 5, No. 1, pp. 146-157, Januari 2024.
[13] M.R. Ramli, H. Sulastri, R. Rianto, "Sentiment Analysis Of Student Opinion Related To Online Learning Using Naïve Bayes Classifier Algorithm And SVM With Adaboost On Twitter Social Media," Telematika: Jurnal Informatika dan Teknologi Informasi 20 (2), pp. 187-201, 2023.
[14] Z.F. Ramadhani and A.B. Mutiara, "Sentiment Analysis of Honkai: Star Rail Indonesian Language Reviews on Google Play Store Using Bidirectional Encoder Representations from Transformers Method," International Journal of Engineering, Science & Information Technology (IJESTY), Volume 3, No. 3, pp. 1-6, 2023.
[15] R. Yunanto, E.P. Wibowo, Rianto, "A BERT Model to Detect Provocative Hoax," Journal of Engineering Science and Technology 18 (5), pp. 2281-2297, 2023.