BERTAnies-Anies, Pre-Election Sentiment Analysis

Justin Martinus[1]

[a] *Program Studi Statistika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Indonesia*
[1] *justin.martinus@ui.ac.id*

ARTICLE INFO

NIM 2106724832

Keywords:
BERT
Anies Baswedan
2024 Indonesia General Election
2024 Indonesia Presidential Election
Sentiment Analysis

ABSTRACT

With 2024 Indonesia presidential election being widely discussed, Anies Baswedan, one of the presidential candidates, raises people's opinion on the topic. One such platform containing the opinions is X (formerly Twitter). Sentiment Analysis may be useful to gain insights regarding those opinions on the platform. However, the sheer number of tweets would pose a challenge if it were to be tackled manually. Thus, fine-tuned BERT models, BERTAnies-Anies, are proposed for the aim of the study. BERTAnies-Anies generally performs well, hovering around 85 to 86% accuracy, with the sole exception of frozen medium BERTAnies, reaching only 65% accuracy. Overall, BERTAnies-Anies has potential to be used to gain more insights.

I. Introduction

The Indonesian general presidential election took place on 14th February 2024. The three candidates vying for the country's executive leadership had already been decided since 13th November 2023. The candidates consisted of Anies Baswedan, Prabowo Subianto, and Ganjar Pranowo with their respective vice president candidate. The eventual elected candidate would then hold the position Indonesia's for the 2024-2029 term.

Anies Baswedan, the candidate representing the National Democratic Party, had amassed significant attention. Various opinions quickly emerged upon the announcement by the related parties. As the former governor of Jakarta, Anies Baswedan had garnered his own share of controversies. Thus, announcing his new standing for the upcoming presidential election at the time sprung up the pros and cons of his candidacy.

X, a social media platform formerly known as Twitter, is now often used as a means for campaigning for each of the candidate's programs. The platform can also be used for exchanging opinions or even attacking the other candidates as shown in the US' 2016 presidential election (Buccoliero et al., 2020). As the interaction can be commented on by everyone on the platform, social media platforms, such as X, are seen as a good way to grasp the overall public sentiment towards a specific candidate.

With its increasingly prominent role in Indonesia's digital interaction, X has found itself as one of the biggest social media platforms commonly used in the largest user of social media in Southeast Asia (Susilo & Putranto, 2018). As such, opinions gathered on the platform can be used to perform in-depth analysis on each candidate. The huge number of tweets open possibilities for various advanced and complex methods to be done for extensive insights on a specific matter. One such case is sentiment analysis (Hakim et al., 2023).

Sentiment analysis is a term used to generalize any research for classifying people's emotions or opinions into a predetermined class based on the text they have written or typed (Hakim et al., 2023). This can be used to gather information in a simpler form, such as positive or negative sentiment (Ramadhan & Gunawan, 2023). In such a case, it is also a form of binary classification, thus there are many adjustments which can be made for machine learning models to be able to do the task more efficiently than a human could. Common machine learning models used include Naïve Bayes, Support Vector Machine, and Decision Trees (Ramadhan & Gunawan, 2023).

DOI:

In performing NLP (Natural Language Processing), BERT (Bidirectional Encoder Representations from Transformers) has proven itself to be superior compared to many other traditional machine learning models (Geni et al., 2023). With Transformers as its base architecture, BERT is capable of capturing contexts of longer dependencies of sequences of tokens. Combined with its bidirectional learning, it can learn context of the text more effectively (Geni et al., 2023). Google offers a pre-trained BERT model, making it more accessible and enabling it to be freely fine-tuned.

This study investigates the effectiveness of fine-tuning BERT model for sentimental analysis of public opinion on X regarding one of the candidates for the presidential election, Anies Baswedan.

II. Literature Review

Research by utilizing sentiment analysis on tweets is commonly done. Politics are no exception to this, with presidential elections as one of the most highlighted research themes. One such example is the research done by (Yaqub et al., 2017) to analyze the behavior of X users and the sentiment of the tweets directed to both presidential candidates for the 2016 US presidential elections. The study found that the sentiments and topics from the tweets can be a good representation of the public's sentiment. It also found that Donald Trump, one of the presidential candidates, generally gave more positive campaign than Hillary Clinton, the other candidate.

In the context of Indonesian presidential election, various studies have also been carried out. (Ningrum & Rahmiyati, 2023) did sentiment analysis using K-Means Algorithm for the 2024 presidential election. (Kristiyanti et al., 2019) tried to predict the winner of the 2019 Indonesia presidential election with SVM (Support Vector Machine), supported by PSO (Particle Swarm Optimization) and GA (Genetic Algorithm) for optimization with N-Gram as the processed initial data after tokenizing. The study predicted Prabowo Subianto – Sandiaga Uno pair as the winner of the 2019 presidential election with the most positive sentiment tweets. (Ramadhan & Gunawan, 2023) approached the problem with logistic regression and descriptive statistics. (Hakim et al., 2023) used stacked ensemble learning with SVM, Random Forest, and Gradient Boosting Classifier with TF-IDF utilization, restating that SVM was good enough for predicting sentiments, but Gradient Boosting Classifier offered higher accuracy. (Setiawan & Dewi, 2023) in particular did sentiment analysis for specifically Anies Baswedan using Naïve Bayes.

The model used in this study, BERT, has also been used in several studies related to sentiment analysis. One such popular theme is COVID tweets sentiment analysis. Both (Lin & Moh, 2021) and (Riaz et al., 2022) investigated the theme using fine-tuned BERT for the specific topic, with the latter primarily focused on the vaccine. While the former did not see significant improvement, the latter did with less time consumed and less data fed to the model. (Geni et al., 2023) even did sentiment analysis for the 2024 Indonesian general election using IndoBERT, a pre-trained BERT model specialized for Bahasa Indonesia, achieving significant improvement in accuracy and F-1 score than the alternatives.

III. Data and Methods

A. *Data*

The data is taken from https://www.kaggle.com/datasets/jocelyndumlao/indonesia-presidential-candidates-dataset-2024, specifically the labeled dataset for Anies Baswedan. The data is from the time frame of January – April 2023. It is specifically mentioned that while it is before the official determination of presidential candidates, the presidential election topic had been widely discussed, thus would still be useful. The data has also been translated to English by the publisher of the dataset. The labels only consist of positive and negative sentiment. It has 4621 tweets after the preprocessing.

B. *Methods*

Artificial Neural Networks

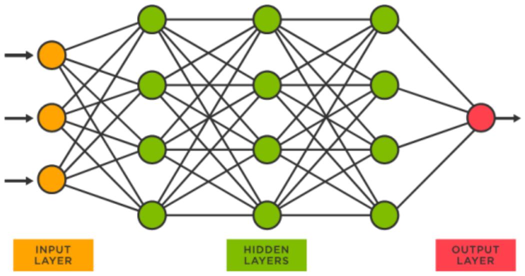

Figure 1 Neural Network Architecture

Artificial Neural Networks (ANN), inspired by study of brain, has an architecture of connected neurons, each having their own weights and biases as parameters (Hwang & Ding, 1997) which are updated each iteration (epoch) using backpropagation. The architecture can easily have and add several layers, allowing the model to learn more complex patterns. Improvement, modifications, and compatibility of newly developed variants of Neural Networks have made it more popular. Generally called deep learning architectures when involving multiple layers, these models are very flexible and since then have been developing rapidly, especially for unstructured data (Goldberg, 2016).

Recurrent Neural Networks

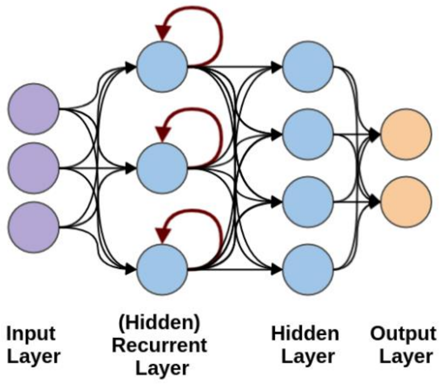

Figure 2 Recurrent Neural Network Architecture

Recurrent Neural Network (RNN) is a deep learning architecture developed for sequential data, such as time series, text, audio, and video data (Yu et al., 2019). It acquires the characteristic by connecting the previous output (from the same recurrent layer) with the current input using a pair of weight and bias for each neuron, recursively doing that for all the inputs without adding new parameters. This means the architecture allows all the previous inputs to influence the current and next inputs, essentially adding the sequential feature to the architecture.

However, Recurrent Neural Network has its primary flaw when handling longer sequences. The weight for the recurrent connections may bring problems by vanishing gradient or exploding gradient problem when the weight's value is less than or more than one respectively. When the value is less than one, the effect of further data points in the sequence has almost no significance in the latter data point. On the other hand, when the value is more than one, the effect of further data points gets exponentially bigger, making the calculation extremely high, thus rendering the training impossible.

Long-Short Term Memory

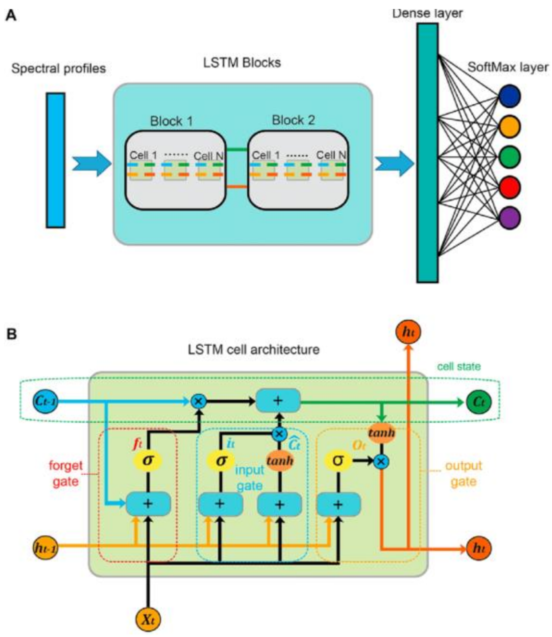

Figure 3 Long- Short Term Memory Architecture

Long-Short Term Memory (LSTM) addresses the issue with RNN by adding gates to the architecture (Yu et al., 2019). The architecture is designed such that each input is affected by the “short-term memory” by the less far separated inputs and by the “long-term memory” by the further separated inputs. This allows the model to capture the more necessary information from the further separated inputs via “long-term memory” while still retaining nearer inputs’ information. More specifically, the input gate adds a candidate value to update the “long-term memory”, while the forget gate decides (outputting 0 or 1) whether the “long-term memory” should be updated.

The number of parameters remains unchanged. Thus, the same set of parameters are used by all the inputs, and updated every epoch, the same as other deep learning architectures that can handle sequences. However, simply adding more cells or layers of LSTM will allow the model to capture more complex sequential pattern in the data.

Word Embedding

Computers, unfortunately, are unable to directly take words as input to be analyzed. Traditionally, non-numerical features can be handled with one-hot encodings. This, however, is inefficient and doesn’t capture any relationships or similarities between words. Word Embedding uses Dense Neural Network to address the issue by capturing semantic and syntactic relationship between words (Lai et al., 2016).

This process is typically done together with the training of the dataset, effectively using backpropagation for parameter optimization, while being part of preprocessing at the same time. The algorithm passes the one-hot encoded inputs to a few Dense Neural Network layers, outputting outputs that are in lower dimension, but effectively representing the distance, which might be interpreted as their similarity, between each word. The training process includes the algorithm trying to predict the surrounding words of each word more accurately. As this process itself doesn’t involve labeled data, Word Embedding is entirely unsupervised.

Sequence-to-Sequence Encoder-Decoder Neural Network and Attention Mechanism

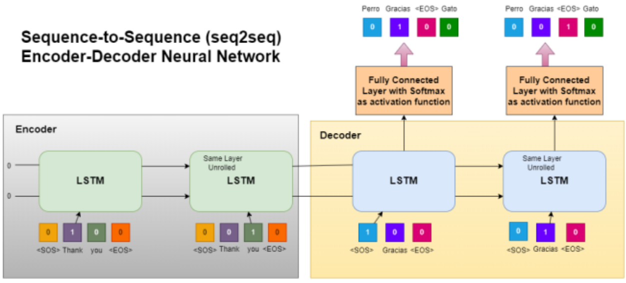

Figure 4 Sequence-to-Sequence Encoder-Decoder Architecture

Encoder-Decoder Architecture, as the name suggests, is divided into two parts, the encoder and decoder. The encoder receives a sequence of symbols (tokens in the case of NLP), then analyzes and compresses it to a single representative vector. The decoder receives the vector and recursively outputs the prediction for tokens until it outputs the “<EOS>” token which stands for “End of Sentence” token (Cho et al., 2014). The first paper introducing the architecture used RNN to process the input sequence and output the representative vector.

Attention mechanism was introduced to highlight specific parts of the representative vector from the encoder. The decoder, by using Attention mechanism, can focus on the more important parts using the context vector as the result from the algorithm. The Attention mechanism calculates the weights for each element in the representative vector based on its relevance to the current step. Especially for language translation, this allows the decoder to capture the context more effectively.

However, as previously explained, RNN is not ideal for longer sequences, thus using LSTM in the architecture instead to handle longer sentences ends with better results (Sutskever et al., 2014). Incorporating the Attention mechanism on top of that gives results that are better at capturing context and handling longer sentences. However, while it is better at handling longer sentences, it is not exactly ideal to handle extremely long sentences due to the limitations of LSTM in the architecture.

Transformers

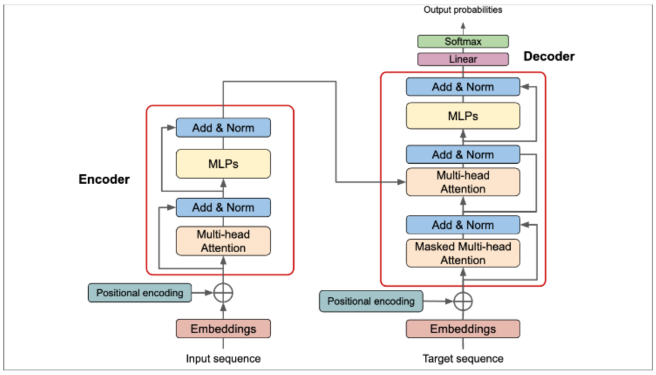

Figure 5 Transformers Architecture

Sequence-to-Sequence Encoder-Decoder Neural Network Architecture has 2 major weaknesses, limited performance for long sequences and slow training speed due to LSTM having to process information sequentially. Thus, (Vaswani et al., 2017) introduced the Transformers architecture, completely abandoning LSTM in the architecture, relying solely on Attention mechanism with Self-Attention mechanism. The new Self-Attention mechanism allows the model to capture the similarities between each word to all the other words in a sentence. On top of abandoning LSTM in the architecture, Transformers is also very optimized for parallelization, enabling more efficient training runtime.

The architecture is still composed of an encoder and decoder but is optimized by implementing the core idea of abandoning LSTM and adopting Self-Attention mechanism in it. The inputs are embedded with Word Embedding, then are assigned unique values by positional encoding commonly with sine and cosine functions with different frequencies to compensate for the lack of positional information by the abandonment of LSTM. The next step is Self-Attention layers for the encoder which involves calculating the similarities to the other elements by creating context vector for each element in the sentence. After the model outputs the first token to signal the beginning of the sentence, it also starts its own Self-Attention layers both within the decoder and the bridge between the encoder and the decoder. The Self-Attention within the decoder (Masked Self-Attention) allows the model to also capture the similarities between the outputs with all the previous outputs, while the Self-Attention bridging the Encoder-Decoder enables the model to align the meaning from the decoder with the encoder, in which the results after all of those are forward-fed to Dense Neural Network with its SoftMax activation, recursively predicts the next token until the “<EOS>” token.

Bidirectional Encoder Representations from Transformers

Bidirectional Encoder Representations from Transformers (BERT) was introduced by (Devlin et al., n.d.) as a pre-trained model with the specific architecture derived from Transformers by using only the decoder and Masked Self-Attention. The encoder handles the encoding and outputs the prediction with Masked Self-Attention, which calculates the similarities between a token with all the previous tokens, throughout the process instead of implementing it only on the outputs of the decoder in the normal Transformers. The model is pre-trained on two unsupervised learning tasks, Masked Language Modelling, which is predicting masked words based on the surrounding context, and Next Sentence Prediction, which is predicting the next sentence and comparing it to the actual next sentence. By training BERT on a massive dataset, it has learned to capture information within sentences effectively. However, as it is not specifically designed for a single task, BERT needs to be fine-tuned to handle a specific task, such as sentiment analysis by adding a single layer for the output with sigmoid or SoftMax activation for binary or multiclass classification respectively.

C. Workflow

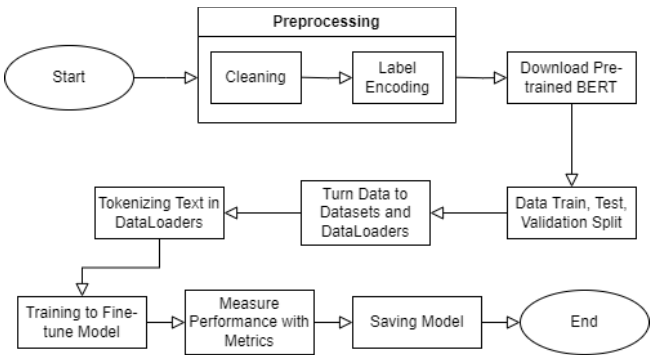

Figure 6 Study's Workflow

This study follows a normal procedure for fine-tuning a model from the huggingface library. The downloaded dataset has other features, e.g. followers and likes, but only the text content of the tweet is used. The cleaning consists of dropping missing values and duplicates, then stripping emojis and other unnecessary symbols. The labels are then applied one-hot encoding. Stop words removal is not included because BERT, pre-trained on a massive dataset, can capture the relevancy of stop words, thus may benefit more from it, instead of interrupting its performance like how more traditional machine learning approaches.

The pre-trained BERT models are “google-bert/bert-base-uncased” and several smaller size BERT by prajjwal1, with one of them will have its BERT layers frozen and having only its classifier layer trained, and are downloaded along with the AutoTokenizer for smoother and optimized tokenizing specifically for the BERT model. To simplify the names, the fine-tuned BERT will be called BERTAnies and BERTAnies-Anies will be used to call the usage of all variants of BERTAnies. The output of the evaluation metric is set up with a function to return the argmax of the logits. The data is then split into train, test, and validation. To accommodate the architecture from the model, the data needs to be changed into Dataset from pandas DataFrame, then into a DataLoaders. The DataLoaders, which contains the train, test, and validation Dataset (60-20-20 percent split), is then tokenized with the AutoTokenizer and is now ready for training and evaluation.

IV. Results and Discussions

A. Descriptive Statistics – Word Cloud

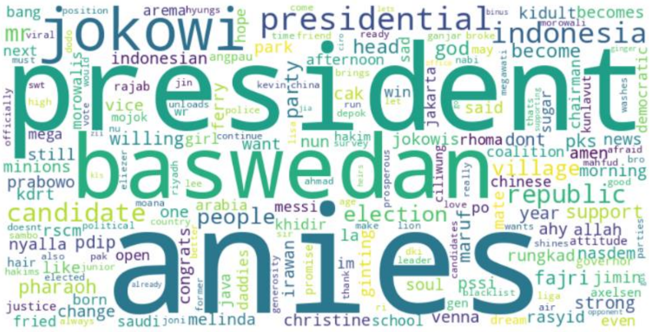

Figure 7 Word Cloud of the Dataset

The Word Cloud is created by calculating the frequency of each term mentioned in the combined text of the dataset. To make sure that only the relevant terms are shown to gain insights, specifically only for the Word Cloud generation, the dataset is preprocessed to remove stop words. Based on the result of the figure, it is obvious that politics terms are the most frequently discussed. There are, however, not a lot of obviously strong sentiments leaning to either positive or negative. This may indicate a challenge to gain sentimental insights only by using Word Cloud visualizations.

B. BERTAnies-Anies

The training used the transformers library and took 109 minutes with CPU or 150 seconds with T4 GPU for BERTAnies, both done in Google Colab. The model is set up to train based on the accuracy metrics. The epoch used is 3 and cross-validation is not used due to the sheer robustness of BERT pre-trained on a massive dataset. The difference between CPU and GPU training time highlights the efficiency of the parallel architecture of Transformers-based models.

Table 1. Performance on Test Data

Model	**Evaluation Metrics**				**Training Runtime**
	Accuracy	***Precision***	***Recall***	***F1-score***	***GPU***
Small BERTAnies	0.8496	0.8566	0.8901	0.8731	50s
Medium BERTAnies	0.8593	0.8812	0.8935	0.8873	54s
Frozen Medium BERTAnies	0.6504	0.6822	0.8168	0.7434	29s
BERTAnies	0.8636	0.9071	0.8691	0.8877	157s

[a.]BERTAnies refers to the Google pre-trained base BERT, while the other variants are smaller pre-trained BERT by prajjwal1.

The small, medium, and regular BERTAnies, as their size increases, their overall performance tend to increase as well. The medium BERTAnies which has its BERT layer frozen underperforms, showing the lack of fine-tuning capabilities by only training the classifier layer in this context. While frozen medium BERTAnies requires the least training runtime, it does not justify the lack of performance with small BERTAnies and Medium BERTAnies require less than double of the runtime but having roughly twenty percent better accuracy. Small BERTAnies and Medium BERTAnies have similar training runtime requirements, while regular BERTAnies roughly has three times the runtime requirement, with some minor improvements in performance compared to the medium BERTAnies model.

V. Conclusions

BERTAnies-Anies generally performs very well on sentiment analysis for the tweet's dataset aimed at the presidential candidate, Anies Baswedan. While regular BERTAnies performs the best, its computational cost, shown by the required training runtime, should be considered based on the aim of further use. Medium BERTAnies may be the most recommended model as it does not require much more computational cost than small BERTAnies to gain improvements in performance, while not requiring much more computational cost like regular BERTAnies with minimal compromises on the performance. On the other hand, frozen medium BERTAnies is not recommended due to its lack of performance. Overall, BERTAnies-Anies shows promising results and may prove to be useful to gain insights regarding sentiments toward Anies Baswedan as a presidential candidate rather than the vague and challenging-to-interpret result from Word Cloud Visualization.

References

[1] Buccoliero, L., Bellio, E., Crestini, G., & Arkoudas, A. (2020). Twitter and politics: Evidence from the US presidential elections 2016. Journal of Marketing Communications, 26(1), 88–114. https://doi.org/10.1080/13527266.2018.1504228

[2] Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1724–1734. https://doi.org/10.3115/v1/D14-1179

[3] Devlin, J., Chang, M.-W., Lee, K., Google, K. T., & Language, A. I. (n.d.). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://github.com/tensorflow/tensor2tensor

[4] Geni, L., Yulianti, E., & Sensuse, D. I. (2023). Sentiment Analysis of Tweets Before the 2024 Elections in Indonesia Using IndoBERT Language Models. Jurnal Ilmiah Teknik Elektro Komputer Dan Informatika (JITEKI), 9(3), 746–757. https://doi.org/10.26555/jiteki.v9i3.26490

[5] Goldberg, Y. (2016). A Primer on Neural Network Models for Natural Language Processing. Journal of Artificial Intelligence Research, 57, 345–420. https://doi.org/10.1613/jair.4992

[6] Hakim, J. N., Sibaroni, Y., & Prasetiyowati, S. S. (2023). Sentiment Analysis of the Indonesia Presidential Election 2024 with Ensemble Learning for Reducing Disinformation on Social Media. 2023 11th International Conference on Information and Communication Technology (ICoICT), 563–568. https://doi.org/10.1109/ICoICT58202.2023.10262584

[7] Hwang, J. T. G., & Ding, A. A. (1997). Prediction Intervals for Artificial Neural Networks. Journal of the American Statistical Association, 92(438), 748–757. https://doi.org/10.1080/01621459.1997.10474027

[8] Kristiyanti, D. A., Normah, & Umam, A. H. (2019). Prediction of Indonesia Presidential Election Results for the 2019-2024 Period Using Twitter Sentiment Analysis. 2019 5th International Conference on New Media Studies (CONMEDIA), 36–42. https://doi.org/10.1109/CONMEDIA46929.2019.8981823

[9] Lai, S., Liu, K., He, S., & Zhao, J. (2016). How to Generate a Good Word Embedding. IEEE Intelligent Systems, 31(6), 5–14. https://doi.org/10.1109/MIS.2016.45

[10] Lin, H. Y., & Moh, T.-S. (2021). Sentiment analysis on COVID tweets using COVID-Twitter-BERT with auxiliary sentence approach. Proceedings of the 2021 ACM Southeast Conference, 234–238. https://doi.org/10.1145/3409334.3452074

[11] Ningrum, L. T., & Rahmiyati, D. (2023). Analysis Sentiment of Twitter User on Indonesia's 2024 Presidential Election Using K-Means Algorithm. JURNAL SISFOTEK GLOBAL, 13(2), 102. https://doi.org/10.38101/sisfotek.v13i2.9609

[12] Ramadhan, F. A., & Gunawan, P. H. (2023). Sentiment Analysis of 2024 Presidential Candidates in Indonesia: Statistical Descriptive and Logistic Regression Approach. 2023 International Conference on Data Science and Its Applications (ICoDSA), 327–332. https://doi.org/10.1109/ICoDSA58501.2023.10276417

[13] Riaz, M. T., Shah Jahan, M., Khawaja, S. G., Shaukat, A., & Zeb, J. (2022). TM-BERT: A Twitter Modified BERT for Sentiment Analysis on Covid-19 Vaccination Tweets. 2022 2nd International Conference on Digital Futures and Transformative Technologies (ICoDT2), 1–6. https://doi.org/10.1109/ICoDT255437.2022.9787395

[14] Setiawan, R., & Dewi, F. (2023). Analysis of Twitter User Sentiment on Presidential Candidate Anies Baswedan Using Naïve Bayes Algorithm. ILKOM Jurnal Ilmiah, 15(3), 473–487. https://doi.org/10.33096/ilkom.v15i3.1775.473-487

[15] Susilo, D., & Putranto, T. D. (2018). Indonesian Youth on Social Media: Study on Content Analysis. Proceedings of the 2017 International Seminar on Social Science and Humanities Research (SSHR 2017). https://doi.org/10.2991/sshr-17.2018.15

[16] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks.

[17] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. http://arxiv.org/abs/1706.03762

[18] Walter, W. [StatQuest: Machine Learning]. (2019, October 29). Sequence-to-Sequence (seq2seq) Encoder-Decoder Neural Networks, Clearly Explained!!! [Video]. YouTube. https://youtu.be/L8HKweZIOmg?si=n42T7sU9_usX8HAU

[19] Walter, W. [StatQuest: Machine Learning]. (2022, January 24). Transformer Neural Networks, ChatGPT's foundation, Clearly Explained!!! [Video]. YouTube. https://youtu.be/zxQyTK8quyY?si=ShhP2RjlpQURugDC

[20] Walter, W. [StatQuest: Machine Learning]. (2023, February 10). Decoder-Only Transformers, ChatGPTs specific Transformer, Clearly Explained!!! [Video]. YouTube. https://youtu.be/bQ5BoolX9Ag?si=eGMp00i5X5v_ygGH

[21] Yaqub, U., Chun, S. A., Atluri, V., & Vaidya, J. (2017). Analysis of political discourse on twitter in the context of the 2016 US presidential elections. Government Information Quarterly, 34(4), 613–626. https://doi.org/10.1016/j.giq.2017.11.001

[22] Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. Neural Computation, 31(7), 1235–1270. https://doi.org/10.1162/neco_a_01199