A Deep Learning Approach to Abusive Language and Hate Speech Detection for the Javanese Language

Kevin Wu
kevinywu@uchicago.edu
University of Chicago
Chicago, IL, USA

Bayard Walsh
bkwalsh@uchicago.edu
University of Chicago
Chicago, IL, USA

Oscar Dorr
oscarmdorr@uchicago.edu
University of Chicago
Chicago, IL, USA

ABSTRACT

This paper develops a deep learning approach to abusive language and hate speech detection using Javanese and Indonesian large language models (LLMs). We experiment on a Javanese Twitter dataset created by Putri et al. [14], aiming to beat their best F-measure of **0.780**. Using a fine-tuned Javanese GPT-2 as a feature extractor for our classifier, the model achieves an F-measure of **0.811**. Surprisingly, utilizing an Indonesian GPT-2 as the feature extractor yields a superior F-measure **0.854**, potentially attributable to code-mixing in Javanese Twitter data or the model's training on colloquial language. This study further explores the nuances of hate speech detection in Javanese, emphasizing language and model choice. All code is available on GitHub: https://github.com/KevinyWu/javanese-hate-speech.

KEYWORDS

Large Language Models, BERT, RoBERTa, GPT-2, Sentiment Analysis, Hate Speech, Javanese, Computational Linguistics

ACM Reference Format:
Kevin Wu, Bayard Walsh, and Oscar Dorr. . A Deep Learning Approach to Abusive Language and Hate Speech Detection for the Javanese Language. In *Proceedings of University of Chicago (Computational Linguistics).* ACM, New York, NY, USA, 6 pages.

1 INTRODUCTION

The challenge of automatic hate speech detection has been applied numerous times in natural language processing; for example, Najafi [8] introduced a model for Turkish social media analysis. There are many negative byproducts caused by online hate speech; from radicalization to coordinated violence countless issues stem from hate speech. Therefore, having a model to screen the massive amount of text posted every day online is essential, and can provide safer online environments. [2].

We implement a classifier that extracts features from the text using a fine-tuned, pretrained BERT, RoBERTa, or GPT-2 model [3, 20, 21], and tested it on a Javanese Twitter dataset created by Putri et al. [14]. We improved the performance on this dataset, from a previous best of F-measure of **0.780** to an F-measure of **0.811** using a pretrained Javanese GPT-2. While this is an improvement over previous paper, we anticipated a higher metric of performance given the computational strength of LLMs. However, achieved an F-measure **0.854** using a pretrained Indonesian GPT-2.

Computational Linguistics, December 6, 2023, Chicago, IL
.

2 BACKGROUND

Javanese is an Austronesian language in the Malayo-Polynesian subgroup spoken by around 82 million native speakers, primarily on the island of Java in Indonesia. Despite being the most widely spoken indigenous language in Indonesia, Javanese is not the official language of the country, which is instead a standardized form of Malay locally known as Bahasa Indonesia. While Bahasa Indonesia, also referred to as Indonesian, is the most commonly spoken language in Indonesia, it is spoken primarily as a *lingua franca* for communication between speakers of different native languages. Most Indonesians, especially those living in metropolitan areas, use Indonesian for professional and political tasks, while speaking native languages like Javanese and Sundanese primarily at home and in colloquial contexts [9].

One of Javanese's most prominent features that differs from many non-Austronesian languages is its rigid register system, with three distinct registers, or "tones," of formality, each containing many grammatical and lexical differences. The words and structure one would use when speaking to a superior or in a formal setting would differ dramatically than those used in informal and colloquial contexts, which adds another layer of difficulty for anyone trying to loosely acquaint themselves with the language for the purpose of content moderation [10].

These are important features of a language to take into account when considering methods for content moderation and hate speech analysis, as there exists a far sharper divide between the language used in upper- and lower-register contexts than more monolingual societies like that of the US, meaning that it is more difficult to make inferences on one based on the other. Since hate speech exists both in-person and online as a primarily lower-register phenomenon, it would be difficult to keep an accurate and up-to-date registry of trends and keywords associated with specifically Javanese hate speech at a national level since the majority of Indonesians don't speak or understand it.

A further caveat that makes manual moderation of Javanese hate speech difficult is the fact that at the colloquial level it is often mixed with Sundanese, a neighboring language that is somewhat mutually intelligible with, but not identical to Javanese. Sundanese is spoken in the Western portion of Java, with many contact zones with Javanese, most notably around the capital city of Jakarta. These contact effects are especially pertinent in Western Javanese dialects, in which the Sundanese influence permeates past mere code-mixing and has significant influence on the vocabulary and grammar of even the standardized form of the dialect. The two languages are often spoken on a gradient continuum, with many socioeconomic factors determining to what extent the two are mixed [4]. Given the informal tone in which most online hate speech appears, it is quite common to see Tweets that may be classified as hate-speech using code-mixing to one degree or another, meaning that any model that hopes to perform accurate sentiment analysis on Javanese must take this phenomenon into account.

These factors illustrate the complexity of content moderation and slang identification in a society as plurilinguistic as Indonesia, which is unfortunately a very pressing task given the current state of internal politics and attitudes in the country. Alexandra and Satria [1] identify a trend of increasing Islamic conservatism and nationalism as responsible for an uptick of hate speech against several minority groups in Indonesia over the last two decades, specifically in the heavily populated areas where Javanese is primarily spoken. Specific examples include widespread hate speech and violence against the minority Shi'a Muslim community in Sampang, East Java in the early 2010s, against the Ahmadiyya Muslim community in Cikeusik, Java during the same period, and against Ahok, the Chinese Christian mayor of Jakarta in the mid 2010s. In addition to these more coordinated hate speech efforts, much of which took place via online communication, the authors also point to general rising trends of online hate speech in Indonesia, usually against religious and ethnic minority communities in the country. Javanese is a language that is hard to moderate given it is considered a low-resource language in NLP, yet it is a language potentially in need of online content moderation since it is spoken in a political climate in which hate speech is actively contributing to acts of violence and the continued oppression of minorities.

3 RELATED WORK

This paper builds on the findings of a previous paper by Putri et al. [14], who developed a human-annotated Javanese hate speech dataset. The previous paper analyzed the use of machine learning approaches such as Naïve Bayes (NB), Support Vector Machine (SVM), and Random Forest Decision Tree (RFDT) in detecting hate speech and abusive language, using word and character N-grams as features.

Another significant paper on low-resource hate speech detection space was published by Abhishek Velankar et al. [19]. This paper focused on applying multilingual and monolingual BERT models for sentiment analysis in Marathi. The authors concluded that a monolingual model performs better, however, the researchers found that these results were limited outside of simple text classification through social media data sets. Additionally, Velankar et al. tested freezing and non-freezing BERT parameters while training the model and classifier, which this paper implements in its model configurations.

Because of the multilingual nature of Indonesia, Javanese is often mixed with other languages, such as Sundanese and Indonesian, when used colloquially on social media. Therefore, much of the relevant research in Javanese Computational Linguistics is analyzing code-mixed data, such as the paper published by Tho et al. [17] on lexicon-based code-mixed sentiment analysis, which focused on using the formal dictionaries between the two languages to define code-mixing cases.

A paper following a similar structure to this one was published by Faisal et al. [6], focused on Indonesian COVID-19 misinformation detection through fine-tuning a pretrained BERT model to a classification model. The researchers focused on COVID-19 misinformation rather than hate speech detection, meaning the performance on detecting different topics could differ based on each topic's relevance on the trained corpus of text. Their model achieved an accuracy of 87.02%, on binary classification of COVID misinformation.

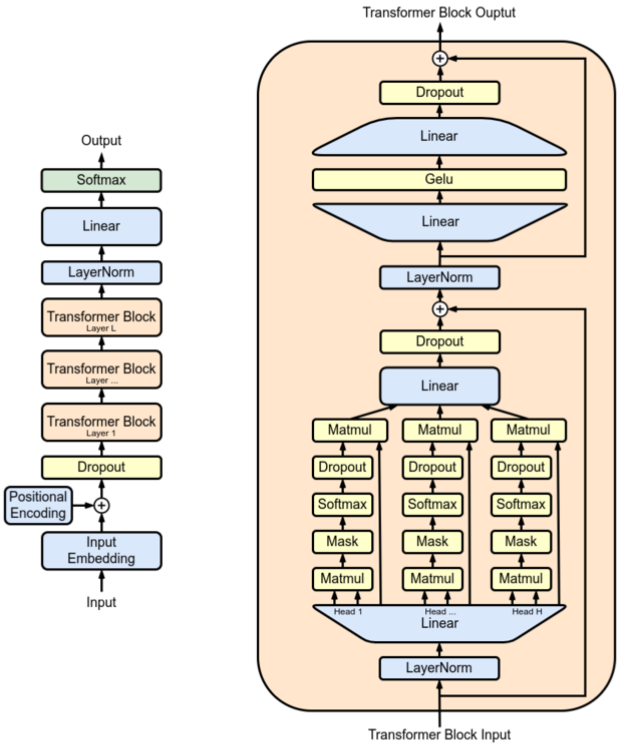

Figure 1: GPT-2 and transformer block architecture [15]

4 METHODOLOGY

Our classifier relies on features extracted from pretrained BERT, RoBERTa, and GPT-2 models. These LLMs rely on the transformer, introduced by Vaswani et al. [18], which consists of an encoder and decoder structure, each comprising multiple layers of self-attention and feed-forward neural networks. Unlike traditional models that process data in a linear sequence, transformers utilize an attention mechanism to weigh the influence of different parts of the input data. This allows for parallel processing of sequences, enhancing efficiency and enabling the model to capture complex relationships in the data.

4.1 BERT, RoBERTa, GPT-2

BERT (Bidirectional Encoder Representations from Transformers) [5] uses the transformer's encoder architecture to understand the context of a word in relation to all other words in a sentence, rather than just the ones immediately adjacent to it. This bidirectional context understanding enables more accurate predictions and interpretations of language. BERT is pretrained on a large corpus using two unsupervised tasks: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). In MLM, some percentage of the input tokens are masked randomly, and the model is trained to predict these masked tokens. NSP involves feeding the model pairs of sentences to learn the relationship between consecutive sentences.

RoBERTa (A Robustly Optimized BERT Pretraining Approach) [7] modifies BERT by removing the NSP task, training on a larger dataset, and using dynamic masking, where the masked tokens change during the training epochs. It also adjusts key hyperparameters like batch size, learning rate, and the number of training steps, enhancing the model's performance on various language understanding benchmarks.

GPT-2 (Generative Pretrained Transformer 2) [15] is another transformer based model which uses only the decoder part of the transformer architecture. It is trained using a variant of the language modeling objective, where it predicts the next word in a sentence given all the previous words. This auto-regressive language modeling process, enables GPT-2 to generate coherent and contextually relevant text continuations from a given prompt.

AttentionMLP

Layer	Dimension	Activation	Params
Input Embed	[B,L,768]	-	-
Hidden Layer	[B,L,64]	ReLU	49,216
Output Weights	[B,1,1]	Softmax	65
Weight-Averaged Embed	**[B,768]**	Total	49,281

Classifier

Layer	Dimension	Activation	Params
Input Embed	[B,L,768]	-	-
Pooling(CLS or Attention)	[B,768]	-	-
Hidden Layer1	[B,64]	ReLU	49,216
Hidden Layer2	[B,32]	ReLU	2,080
Output Classification	[B,3]	Softmax	99
		Total	51,395

Figure 2: AttentionMLP and Classifier architectures; input dimension is the output of BERT/GPT2: [B (batch size), L (sequence length), 768 (token embedding dimension)]

4.2 Our Model

We experiment with Javanese versions of BERT, RoBERTa, and GPT-2 [21]. Each model is trained pretrained on Javanese Wikipedia articles (319 MB of text) through late December 2020. Due to the formal nature of Wikipedia language, we also experiment with Indonesian versions of RoBERTa [20] and GPT-2 [3], which are pretrained on OSCAR (Open Super-large Crawled Aggregated coRpus) [11, 12] and MC4 (Multilingual Colossal, Cleaned version of Common Crawl) [16]. These datasets contain more informal and colloquial language that may extrapolate better to our task. Though Javanese and Indonesian are separate languages, we think this experimentation is reasonable due to code-mixing in both the Javanese Twitter dataset and the pretraining data. Because these LLMs provide more context in terms of both direction and length than N-gram models, we believe that we can improve upon Putri et al. [14].

After obtaining token embeddings from BERT or RoBERTa, we test two pooling methods. The first is simply taking the [CLS] token as our feature, a special token added at the beginning of each input sequence whose final hidden state is used as the aggregate sequence representation for classification tasks [5]. The second method is training a 2-layer feed-forward neural network (which we call AttentionMLP) that learns "attentions" for each token embedding in the sequence. We then use these attentions to weight-average each token embedding in the sequence, using the result to represent the entire sequence. Since GPT-2 lacks a [CLS], we only use the AttentionMLP pooling method.

Lastly, we pass our pooled sequence representation into a 3-layer feed-forward neural network which outputs the class. The AttentionMLP and Classifier architectures is detailed in Fig. 2.

Table 1: Original paper's F-measure on Javanese data [14]

Model	**NB**	**SVM**	**RFDT**
Word Unigram	.752	.778	.762
Word Bigram	.627	.709	.680
Word Trigram	.627	.641	.628
Word Unigram + Bigram	.750	**.780**	.771
Word Bigrams + Trigram	.624	.675	.665
Word Unigram + Bigram + Trigram	.750	**.780**	.755
Char Trigram	.726	.709	.711
Char Quadgram	.743	.752	.758
Char Trigram + Quadgram	.708	.660	.702

5 EXPERIMENTS

In this section, we will describe the dataset introduced by Putri et al. [14] and present our training and evaluation methodology. A strong baseline evaluation metric we aim to exceed is the *Word Unigram + Bigram with SVM* classifier which achieved an F-measure of 0.780 on the target dataset, which was the highest performance achieved in the original paper. Their full results are reported in Tab. 1.

5.1 Data

The dataset consists of 3477 Tweets in Javanese (limited to 280 characters), where each Tweet is accompanied by two binary labels: one indicating whether it contains abusive language, and one indicating whether it is hateful. These labels were determined by two native Javanese speakers, achieving a Cohen's Kappa of 0.44. Many Tweets contain emojis, which are represented as Unicode, and user tags are universally replaced with the USER token.

Following [14], we convert the multi-label data to multi-class with the label power-set (LP) method. There are no instances of Tweets which are hateful but do not contain abusive language, so we have three classes, seen in Fig. 3. The data is imbalanced, with only 173 hate-speech examples.

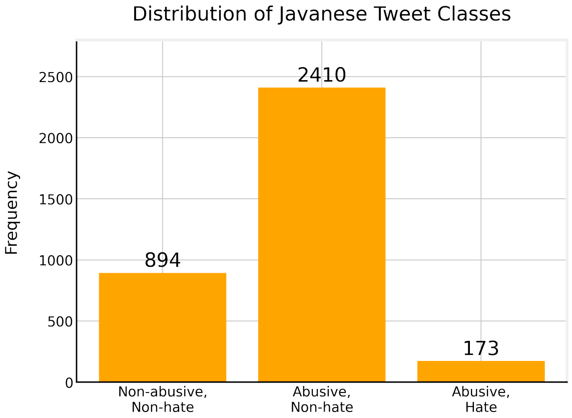

Figure 3: Not all abusive language is hate speech

5.2 Training and Evaluation

We perform five-fold cross validation, with 80-10-10 splits between train, dev, and test sets. We train for 10 epochs, stopping early if the F-measure on the dev set does not improve for two epochs. We use the Adam Optimizer with 0.0005 learning rate and batch size of 16. We report our results as the mean accuracy on the test set. All training was done on Google Colaboratory with an NVIDIA T4 GPU. Training configurations are summarized in Tab. 2.

As described in Sec. 4.2, we experiment with two pooling methods, [CLS] and attention. In addition, we test different levels of fine tuning, unfreezing the last zero, one, two, or three layers of the pretrained model (zero means no fine-tuning).

Table 2: Training configurations

Configuration	Value
Learning Rate	.0005
Batch Size	16
Max Epochs	10
Patience (by F-Measure)	2
Optimizer	Adam
Loss Function	Cross-Entropy
Train-Dev-Test	80-10-10

6 RESULTS

We observed notable variations in performance across different language models and pooling methods when experimenting with Javanese and Indonesian feature extractors. For Javanese, GPT-2 with attention pooling emerged as the most effective, particularly when two layers were unfrozen, achieving a peak score of 0.811 ± 0.022. GPT-2 consistently outperformed BERT and RoBERTa, which peaked at 0.775 ± 0.006 (RoBERTa, attention pooling, one unfrozen layer). Somewhat surprisingly, the Indonesian RoBERTa and GPT-2 models actually outperformed their Javanese counterparts. GPT-2 again demonstrated superior performance, reaching a score of 0.854 ± 0.014. Several configurations of RoBERTa also beat the benchmark F-measure of 0.780.

Table 3: F-measure results for various model configurations

		Unfrozen Layers		
Model	**Pooling**	0	1	2
Javanese				
BERT	[CLS]	.691±.008	.757±.025	.670±.088
BERT	Attention	.755±.020	.759±.042	.635±.085
RoBERTa	[CLS]	.684±.021	.763±.021	.760±.014
RoBERTa	Attention	.748±.021	.775±.006	.658±.076
GPT-2	Attention	.800±.022	.797±.030	**.811±.022**
Indonesian				
RoBERTa	[CLS]	.692±.028	.796±.018	.736±.039
RoBERTa	Attention	.772±.014	.803±.019	.725±.078
GPT-2	Attention	.826±.021	**.854±.014**	.853±.025

With BERT and RoBERTa, attention pooling consistently outperformed the [CLS] token, and one unfrozen layer layer performed by far the best. Unfreezing a single layer may provide just enough flexibility for the model to adapt to the specificities of the new task or language without losing the general understanding it has already acquired. Unfreezing more layers can may have caused the models to overfit and lose crucial pretrained knowledge.

Full results are reported in Tab. 3. Unreported are the results for unfreezing the final three layers, but in every instance the F-measure is worse than unfreezing the final two layers. A confusion matrix of our best model's (Indonesian GPT-2 with one unfrozen layer) predictions on the test set is shown in Fig. 4. From the confusion matrix, we see that the most commonly misclassified category is hate speech. 37% of hate speech examples are misclassified as abusive language but not hate speech, though only 3% (one example in the test set) is classified as non-abusive and non-hate. On the other hand, very few instances of non-hate speech are misclassified as hate speech.

6.1 Misclassified Tweets

Here, we take a look at some Tweets misclassified by our best model.

Model classifies abusive language as hate speech:

> **Javanese:** *Prosedur bikin Somasi buat tetangga pekok yg cm ngeliatin anaknya gk punya etika jam 9 malem gedabrukan maen bola gimana sih?'*
>
> **Translation:** *What's the procedure for making a subpoena for a destitute neighbor who sees that his child has no etiquette at 9 pm and is banging around playing football?'*

A reason for misclassification is the negative tone, created by words like *pekok* (or *destitute*), which could be commonly used in hate speech. Furthermore, the idea of making a subpoena involves an act of persecution of the neighbor, and this structure of threatening to punish a person could be common in hate speech cases. However, the comment is not directed at a group, which is the defining aspect of hate speech that the model misinterprets.

Model classifies hate speech as non-abusive, non-hateful:

> **Javanese:** *USER Mereka itu spt tissue bro.. Tissue toilet buat bersihin eek majikannya yg kesandung RUU HIP'*
> **Translation:** *USER They are like tissue, bro... Toilet tissue is used to clean up the bosses who are caught in the HIP Bill'*

In this instance, there is also code mixing with English, which partially explains the lower performance because of the unfamiliar language. Additionally, *RUU HIP'* is referencing a specific bill on Pancasila [13], or the religious laws incorporated in the Indonesian government. Therefore because of the specificity of the law and the English mixing, the hate speech directed at the religious group was too specific for the model to interpret the comment. This could be solved by fine-tuning on a larger dataset which encompasses more specific laws, groups, and events.

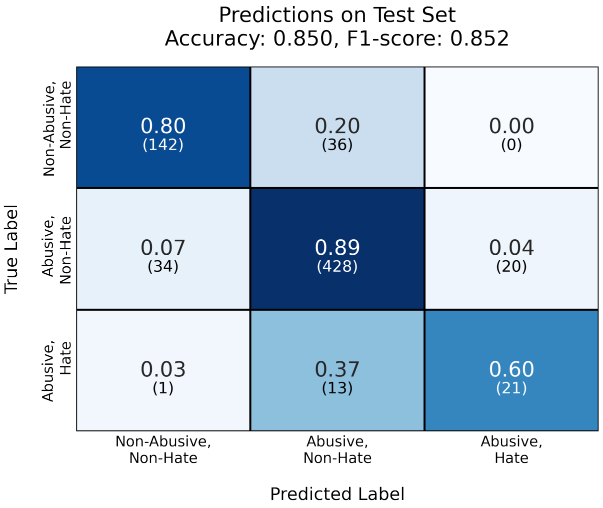

Figure 4: Confusion matrix of our best model's (Indonesian GPT-2 with 1 unfrozen layer) predictions on the test set

7 DISCUSSION

Our models outperform the 0.780 F-measure benchmark, with the Javanese model scoring 0.811, 0.031 higher than the benchmark, and the Indonesian model scoring 0.854 around 0.074 higher. This result also points to the most surprising aspect of this test, which is that the models trained on the Indonesian language far outperformed the models trained on Javanese, despite the fact that the dataset consisted of Javanese Tweets.

The reason for these results comes down to a few main factors. First, whereas the Indonesian LLMs we use in this paper were pretrained on Common Crawl and Wikipedia, which includes hundreds of websites and social media platforms (but not Twitter) [3], the Javanese LLMs were pretrained exclusively on Wikipedia articles [21]. This means that while the Javanese models have the advantage of recognizing more specific words that might appear in the test set, it will be comparatively unfamiliar with many of the slang words and non-linguistic elements like emojis, which are used similarly between Indonesian and Javanese.

Secondly, as mentioned by Putri et al. [14], Javanese and Indonesian are often code mixed quite extensively at informal and colloquial levels, like Tweets. This means that even though the Javanese model was trained extensively on Javanese web data, it may not have encountered as much of this code mixing as may have appeared in the data set, especially since Wikipedia articles are less likely to be code-mixed. While we did not explore this explicitly, we hypothesize that the Common Crawl data from OSCAR and MC4 contain code-mixed text which boosts the Indonesian model's ability to interpret Javanese.

8 CONCLUSION AND FUTURE WORK

This paper demonstrates the fact that LLMs like BERT and GPT can be used as effective tools for online content moderation in low resource languages where other methods may not suffice. Our results also demonstrate the importance of tailoring the type of speech upon which a model is trained to that which it will eventually be tested upon. While there are some languages where a model trained exclusively on formal or academic writing would be able to effectively analyze informal content like Tweets, the same cannot be said of Javanese, where it has been shown that even an Indonesian model that is trained on low-register written content will outperform a Javanese model trained only on formal content like Wikipedia. We believe that it is imperative to train a model on a dataset that includes many low-register and code-mixed examples before implementing it for any sentiment analysis tasks.

Because of the increased word context given by LLMs, a future study exploring the capabilities of our model employed in this paper would benefit from testing on data with more text. Therefore, it would be beneficial to replicate the sentiment analysis experiment with data instances longer than Tweets, such as articles, to see if the increased word context leads to improved detection of abusive language and hate speech. Additionally, we could fine-tune our model on various existing Javanese and Indonesian datasets for further benchmarks.

Training a Javanese LLM from scratch would likely improve our results. We saw in Sec. 6.1 that some Tweets could be not detected by the model if it lacks understanding of specific laws or news, so the model could be improved by training on a larger, more recent corpus of text which includes specific information about local and national news, laws, and events. In the fine-tuning process, gathering more data from Javanese Twitter users would make the size of the data set more representative of actual users, although native speakers would be needed to manually label the data with a larger data set. Lastly, there is more opportunities with exploring the specific effects of code-mixing, which could be done by controlling for the amount of code-mixed text used during pre-training.

ACKNOWLEDGMENTS

To Daniel Lam, for his mentorship and teaching, to Wilson Wongso, for his Javanese BERT models, and to Putri et. al for the high quality Javanese Twitter dataset.

REFERENCES

[1] Lina Alexandra and Alif Satria. 2023. Identifying Hate Speech Trends and Prevention in Indonesia: a Cross-Case Comparison.

[2] Ali Alshehri, El Moatez Billah Nagoudi, and Muhammad Abdul-Mageed. 2020. Understanding and Detecting Dangerous Speech in Social Media. arXiv:2005.06608 https://arxiv.org/abs/2005.06608

[3] Akmal alvinwatner, Cahya Wirawan, Galuh Sahid, Muhammad Agung Hambali, Muhammad Fhadli, and Samsul Rahmadani. 2023. gpt2-small-indonesian.

[4] Luthfiatul Azizah Nuril Anwar. 2021. The Role of the Surabaya Javanese Dialect (Suroboyoan Dialect).

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://api.semanticscholar.org/CorpusID:52967399

[6] Douglas Raevan Faisal and Rahmad Mahendra. 2022. Two-Stage Classifier for COVID-19 Misinformation Detection Using BERT: a Study on Indonesian Tweets. arXiv:2206.15359 [cs.CL]

[7] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. https://api.semanticscholar.org/CorpusID:198953378

[8] Ali Najafi and Onur Varol. 2023. TurkishBERTweet: Fast and Reliable Large Language Model for Social Media Analysis. arXiv:2311.18063 [cs.CL]

[9] Berndt Nothofer. 2009. Javanese.

[10] The Editors of Encyclopaedia Britannica. 2023. Javanese Language.

[11] Pedro Javier Ortiz Su'arez, Laurent Romary, and Benoit Sagot. 2020. A Monolingual Approach to Contextualized Word Embeddings for Mid-Resource Languages. , 1703–1714 pages. https://www.aclweb.org/anthology/2020.acl-main.156

[12] Pedro Javier Ortiz Su'arez, Benoit Sagot, and Laurent Romary. 2019. Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures. , 9 – 16 pages. https://doi.org/10.14618/ids-pub-9021

[13] The Jakarta Post. 2020. Govt opts to postpone deliberation of controversial bill on Pancasila amid backlash against House. https://www.thejakartapost.com/news/2020/06/16/govt-opts-to-postpone-deliberation-of-controversial-bill-on-pancasila-amid-backlash-against-house.html

[14] Shofianina Dwi Ananda Putri, Muhammad Okky Ibrohim, and Indra Budi. 2021. Abusive language and hate speech detection for Javanese and Sundanese languages in tweets: Dataset and preliminary study. , 65–69 pages.

[15] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. https://api.semanticscholar.org/CorpusID:160025533

[16] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. arXiv:1910.10683

[17] C Tho, Y Heryadi, L Lukas, and A Wibowo. 2021. Code-mixed sentiment analysis of Indonesian language and Javanese language using Lexicon based approach. , 012084 pages.

[18] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. https://api.semanticscholar.org/CorpusID:13756489

[19] Abhishek Velankar, Hrushikesh Patil, and Raviraj Joshi. 2022. Mono vs Multilingual BERT for Hate Speech Detection and Text Classification: A Case Study in Marathi. , 121–128 pages. https://doi.org/10.1007/978-3-031-20650-4_10

[20] Wilson Wongso, Steven Limcorn, Samsul Rahmadani, and Chew Kok Wah. 2023. gpt2-small-indonesian.

[21] Wilson Wongso, David Samuel Setiawan, and Derwin Suhartono. 2021. Causal and Masked Language Modeling of Javanese Language using Transformer-based Architectures. , 7 pages.

Received 6 December 2023