A Systematic Evaluation of BERT Classifiers for Indonesia-based Text Data

Yogie Oktavianus Sihombing*[1], Khusnul Muchlisin[2], Tri Fidrian Arya[3], Moh. Jabir Mubarok[4], Reza Fuad Rachmadi[5]

[1,2,3,4,5]*Faculty of Intelligent Electrical and Informatics Technology, Electrical Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia*
E-mail : [1,]yogie.sihombing27@gmail.com, [2]shin.kotakpos@gmail.com, [3]tri.fidrian.arya@gmail.com, [4]jabirmubarok@gmail.com, [5]fuad@its.ac.id*
**Corresponding author*
Received 9 March 2026; Revised 23 March 2026; Accepted 4 April 2026

Abstract - *This study presents a systematic evaluation of Indonesian BERT models across multiple natural language processing (NLP) tasks, including named entity recognition (NER), sentiment analysis (SA), emotion classification (EmoT), and hate speech detection (HS). Unlike prior studies that primarily focus on effectiveness metrics, this work incorporates both effectiveness (F1-Macro and accuracy) and efficiency (training time and memory usage) to provide a more comprehensive benchmark. Experimental results show that IndoRoBERTa achieves the highest overall F1-Macro (0.826), indicating strong generalization across tasks, while IndoNLU attains the highest accuracy (0.833), suggesting better performance on dominant classes. IndoLEM demonstrates superior efficiency with the lowest training time (988.68 seconds) and minimal GPU memory usage (4.00 GB), making it suitable for resource-constrained environments. In contrast, the multilingual mBERT model exhibits higher computational cost with comparatively lower efficiency. The findings highlight a trade-off between performance and computational efficiency, where monolingual Indonesian models consistently outperform multilingual models in both effectiveness and resource utilization. These results provide practical insights for selecting appropriate pretrained language models based on task requirements and computational constraints in Indonesian NLP applications.*

Keywords - *BERT; Indonesian NLP; model efficiency; multi-task evaluation*

1. INTRODUCTION

Natural Language Processing (NLP) has experienced rapid development in recent years, driven by advances in artificial intelligence and the increasing availability of large-scale textual data [1], [2]. In the Indonesian context, NLP applications have been widely explored across various tasks, including named entity recognition (NER), sentiment analysis (SA), emotion classification (EmoT), and hate speech detection (HS) [3], [4]. These tasks play an important role in extracting meaningful information from textual data and supporting decision-making processes in multiple domains.

The emergence of pretrained language models, particularly BERT (Bidirectional Encoder Representations from Transformers), has significantly improved performance across NLP tasks [5]. Various Indonesian-specific BERT models, such as IndoNLU and IndoLEM, have been developed to better capture linguistic characteristics of the Indonesian language [6], [7]. In addition, multilingual models such as mBERT have been widely adopted due to their ability to generalize across multiple languages [8]. As a result, evaluating and comparing the performance of these models has become an important research direction in Indonesian NLP.

Several previous studies have comparatively evaluated BERT variants. Most of these studies focus primarily on effectiveness metrics, such as F1-score and accuracy, in specific tasks

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v25i2.15843

such as text classification or sentiment analysis [9], [10]. Some works have also explored efficiency aspects, including reducing training time and memory consumption [11]. However, these studies are generally limited to single-task evaluations or focus on a single dimension of performance. Consequently, a comprehensive understanding of how different BERT models perform across multiple tasks while simultaneously considering both effectiveness and efficiency remains limited.

To address this gap, this study presents a systematic multi-task benchmark of Indonesian BERT-family models, including IndoNLU, IndoLEM, IndoRoBERTa, and mBERT, across four representative NLP tasks: named entity recognition, sentiment analysis, emotion classification, and hate speech detection. Unlike prior studies, this work jointly evaluates both effectiveness (F1-Macro and accuracy) and efficiency (training time, validation time, and memory usage), enabling a more holistic comparison of model performance.

The main contributions of this study are as follows. First, it provides a multi-task benchmark evaluation of Indonesian BERT models across diverse NLP tasks. Second, it integrates effectiveness and efficiency metrics within a unified evaluation framework. Third, it analyzes the trade-offs between performance and computational cost across monolingual and multilingual models. Finally, this study offers practical insights to guide model selection based on task requirements and resource constraints in real-world Indonesian NLP applications.

2. RESEARCH METHOD

The research process begins with the collection of supervised datasets from various Indonesian text-based tasks, including named entity recognition (NER), sentiment analysis (SA), emotion classification (EmoT), and hate speech detection (HS). Furthermore, training is conducted from various Indonesian BERT variants, namely IndoNLU, i.e. IndoBERT-Base-P2, IndoLEM, i.e. IndoBERTBase-Uncased, IndoRoBERTa from Flax Community, and mBERT. Finally, model performance is evaluated using F1-Macro, accuracy, training time, validation time, and memory usage metrics. The entire research phase was conducted experimentally. The research stages are shown in Figure 1.

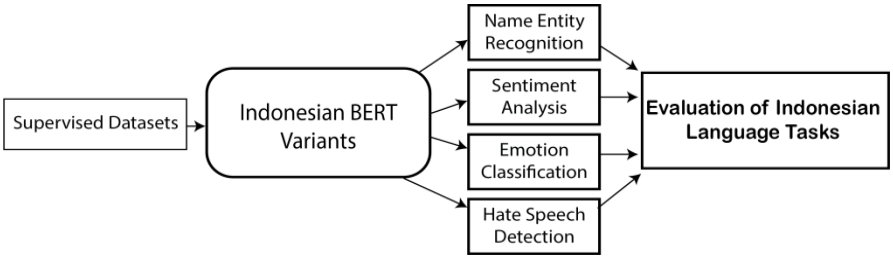

Figure 1: Stages of Model Performance Evaluation on Indonesian Text-Based Tasks

2.1. Dataset

The first process is the collection of datasets from various sources for text-based Indonesian tasks. The datasets used are shown in Table 1.

Table 1. Data Collection from Various Indonesian Text-Based Tasks.

Task	Label	Dataset Total
NER	B-LOC, B-ORG, B-PER, I-LOC, I-ORG, I-PER, O	2340
SA	NEG, NEU, POS	11000
EmoT	SAD, FEA, ANG, HAP, LOV	4401
HS	HS, NO_HS	14129

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v25i2.15843

For the NER task, we obtained a dataset from research [12] totaling about 2340 sentences, with 7 labels: Beginning of Location (B-LOC), Beginning of Organization (B-ORG), Beginning of Person (B- PER), Inside Location (I-LOC), Inside Organization (I-ORG), Inside Person (I-PER), and Outside (O). For the SA task, the dataset obtained from the research [13] consists of 11000 rows of data, with labels: Negative (NEG), Neutral (NEU), and Positive (POS). Furthermore, for the EmoT task, a dataset was taken from research [14] with a total of 4401 rows of data, with labels: Sadness (SAD), Fear (FEA), Anger (ANG), Happy (HAP), and Love (LOV). And the last task is HS, the dataset comes from the combination of the [15] and [16] datasets into 14129 rows of data with two labels: Non-Hate Speech (NO_HS) and Hate Speech (HS).

2.2. Model Architecture

Basically, each task trained on the different variations of BERT Bahasa Indonesia is trained using the BERT architecture itself. More about the tasks that use the BERT architecture is depicted through Figure 2.

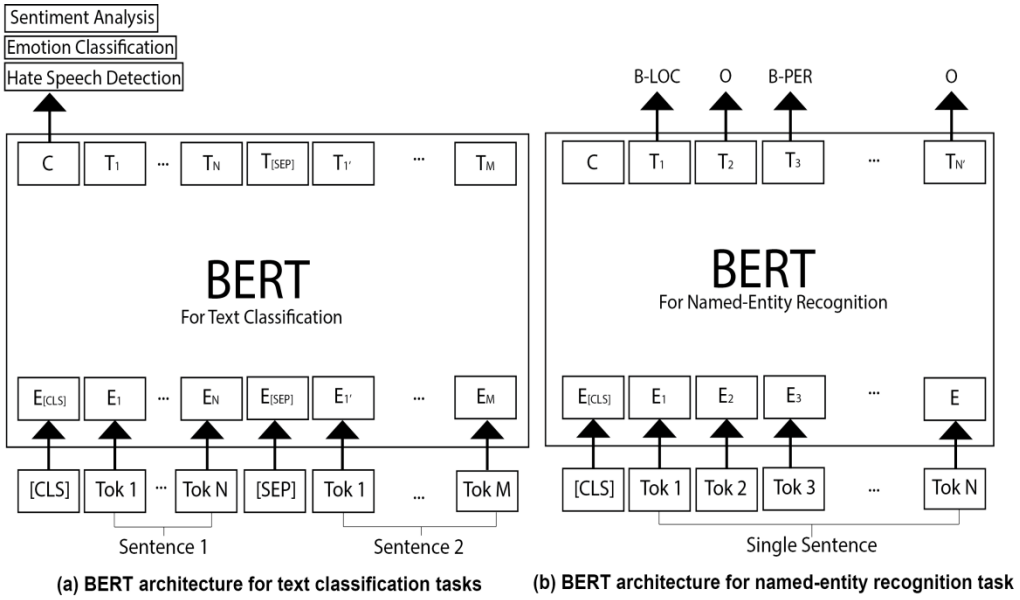

Figure 2: BERT For Various Indonesian Tasks.

As shown in Figure 2, BERT is used in two different configurations based on the task formulation: sequence classification and token classification. Figure 2(a) illustrates sequence classification tasks, including sentiment analysis (SA), emotion classification (EmoT), and hate speech detection (HS), where the input text is encoded as a single sequence supplemented with special tokens [CLS] and [SEP]. The contextual representation of the [CLS] token is then used as a holistic representation of the input, which is fed into a classification layer to generate a single label prediction.

In contrast, Figure 2(b) shows the named entity recognition (NER) task, where BERT is formulated as a token-level classification model, in which predictions are generated at the token level. Each token is assigned a contextual embedding, allowing the model to classify entity categories such as people, places, and organizations for each token individually. Since BERT uses subword tokenization, individual words can be broken down into multiple subword units; thus, label consistency is maintained by propagating the original word-level label to all corresponding subword tokens.

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v25i2.15843

This distinction reflects the fundamental difference between sequence-level and token-level modeling, where sequence classification captures global semantic representations, whereas token classification requires detailed contextual understanding at the token level.

2.3. Experimental Setup

Evaluation of the performance of the Indonesian BERT model for all variants (Table 1) on all tasks is based on F1-Macro, accuracy, time, and memory usage metrics. F1-Macro and accuracy to measure the effectiveness of the model's performance in performing its task. F1-Macro is used as a measure in looking fairly at the unbalanced distribution of the dataset [34]. Meanwhile, time and memory usage to measure the efficiency of model performance in utilizing computing device resources used when training the model. The device specifications used in this experiment are shown in Table 2.

Table 2. Device Specifications.

Operating System	Windows 11
Processor	Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz (12 CPUs) 2.6GHz
RAM Size	16196 MB (~16 GB)
GPU Size	6144 MB (~6 GB)
GPU Version	NVIDIA GeForce RTX 2060
GPU Type	GDDR6

All experiments were conducted under consistent and controlled conditions to ensure reproducibility and fair comparison across models. All models were implemented using the HuggingFace Transformers library with a PyTorch backend. Details of the model used can be found in Table 3. Tokenization was performed using the pretrained tokenizer corresponding to each model. Unless otherwise specified, the maximum sequence length was set to 128 tokens, with post-padding and post-truncation applied. All models were trained using the AdamW optimizer with a learning rate of 5e-5 for 10 epochs. Hyperparameters were kept consistent across all models to ensure fair comparison, except for batch size adjustments required due to computational constraints.

Table 3. BERT Models for Experiment

IndoNLU	indobenchmark/indobert-base-p1
IndoRoBERTa	flax-community/indonesian-roberta-base
IndoLEM	indolem/indobert-base-uncased
mBERT	google-bert/bert-base-multilingual-uncased

The configuration for sequence classification tasks is shown in Table 4, which includes sentiment analysis (SA), emotion classification (EmoT), and hate speech detection (HS). These classification tasks use the AutoModelForSequenceClassification architecture. The dataset is split into training, validation, and test sets in a ratio of approximately 80:10:10 using a fixed random seed. A batch size of 16 is used for IndoNLU, IndoLEM, and IndoRoBERTa. Due to higher computational demands, the mBERT model is trained with a smaller batch size of 8 to prevent GPU memory overflow. Despite these differences, all other hyperparameters were kept consistent to ensure a fair comparison. Evaluation for the sequence classification task was performed using accuracy and F1-Macro, calculated using the scikit-learn library. These metrics were chosen to account for both overall accuracy and class imbalance.

Table 4. Experimental Setup for Sequence Classification Tasks (SA, EmoT, HS)

Model Architecture	AutoModelForSequenceClassification
Library	HuggingFace Transformers (PyTorch)
Tokenizer	Pretrained tokenizer (model-specific)
Max Sequence Length	128 tokens

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v25i2.15843

Padding Strategy	Post-padding
Truncation Strategy	Post-truncation
Attention Mask	Used
Train/Val/Test Split	80:10:10
Random Seed	Fixed
Optimizer	AdamW
Learning Rate	5e-5
Batch Size	16 (IndoNLU, IndoLEM, IndoRoBERTa); 8 (mBERT)
Epochs	10
Evaluation Metrics	Accuracy, F1-Macro (scikit-learn)
Training Time	Python time module
GPU Memory	GPUtil (peak usage)
CPU Memory	psutil (peak usage)

The configuration for the Named Entity Recognition (NER) task is shown in Table 5. The NER task performs token classification using AutoModelForTokenClassification. The dataset is restructured into a sentence-level sequence of token-label pairs. Since the BERT-based model uses subword tokenization, each word may be split into multiple tokens; therefore, the original label is passed to all corresponding subword tokens to maintain consistency. The NER dataset is split into training and validation sets at a 90:10 ratio using a fixed random seed. The data sequences are padded and truncated to a maximum length of 128 tokens. Evaluation for the NER task is performed using the seqeval library to calculate the F1-Macro score, along with accuracy as an additional metric.

Table 5. Experimental Setup for Named Entity Recognition (NER)

Model Architecture	AutoModelForTokenClassification
Library	HuggingFace Transformers (PyTorch)
Tokenizer	Pretrained tokenizer (model-specific)
Tokenization Handling	Subword tokenization with label alignment
Label Strategy	Labels duplicated across subwords
Max Sequence Length	128 tokens
Padding Strategy	Post-padding
Truncation Strategy	Post-truncation
Train/Validation Split	90:10
Random Seed	2018
Optimizer	AdamW
Learning Rate	5e-5
Batch Size	16
Epochs	10
Scheduler	Linear scheduler
Evaluation Metrics	F1-Macro (seqeval), Accuracy
Training Time	Python time module
GPU Memory	GPUtil (peak usage)
CPU Memory	psutil (peak usage)

For all experiments, training and validation times were measured using the Python time module. GPU memory usage was monitored using GPUtil, while CPU memory usage was tracked using the psutil library. Peak memory usage during training was recorded as an efficiency metric.

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v25i2.15843

Each experiment was conducted multiple times during the development process to ensure stability of the results. The reported results correspond to the final consistent outcomes obtained after repeated trials. Due to the lack of systematic logging for each run, statistical significance analysis was not performed. Future work may include controlled repeated experiments to further validate the robustness of the results.

2.5. Evaluation Metrics

For evaluation use metric of F1-Macro, accuracy, memory usage, and training and validation times.

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \quad (1)$$

Based on the equation 1, accuracy is one of the most commonly used evaluation metrics to measure the performance of classification models. TP (True Positive) indicates the number of positive cases that were correctly predicted, while TN (True Negative) is the number of negative cases that were also correctly predicted. FP (False Positive) is the number of negative cases that were incorrectly predicted as positive, and FN (False Negative) is the number of positive cases that were incorrectly predicted as negative. In other words, accuracy calculates the proportion of data that is correctly classified by the model, be it positive or negative. This metric is very useful when the distribution of data between classes is balanced, but can be misleading if the data is very unbalanced. In the case of unbalanced data, the model may achieve high accuracy by only predicting the majority of classes, without really understanding the characteristics of the minority classes. Therefore, while accuracy is easy to understand and calculate, the selection of evaluation metrics should be tailored to the context and characteristics of the data being used [17].

$$F1\text{-}Macro = (2 \times Precision \times Recall) / (Precision + Recall) \quad (2)$$

Next is F1-Macro based on the equation 2, which is an averaged version of the F1-score metric used to evaluate classification models, especially in the case of multiclass classification. This metric strikes a balance between precision (accuracy of positive predictions) and recall (ability to catch all positive cases), making it suitable for use when we want to consider both aspects simultaneously. In the context of F1-Macro, F1 values are calculated separately for each class, then averaged regardless of the proportion of data in each class (each class has equal weight). This makes F1-Macro very useful when facing datasets with an unbalanced class distribution, as it does not favor the majority class. As such, F1-Macro gives a fairer picture of the overall model performance, especially in the context of complex multiclass classification [18].

$$Avg.GPU\ Mem.Used = \sum_{(GPU)} gpu.memoryUsed \quad (3)$$

The third, based on the equation 3, is a formulathat describes how to calculate the average GPU (Graphics Processing Unit) memory usage. In this formula, Avg. GPU Mem. Used refers to the average GPU memory usage, which is calculated by sum- ming up all the GPU memory usage values recorded by each GPU in the system. The notation Σ indicates the sum total of all recorded gpu.memoryUsed values, which represents the amount of memory used by each GPU involved. In other words, this formula accumulates the memory used by all GPUs, and the result is divided to get the average GPU memory usage across the devices. This formula is useful in monitoring and analyzing performance and memory usage in GPU-based computing systems [19].

$$Avg.CPU\ Mem.Used = \sum_{(CPU)} cpu.memoryUsed \quad (4)$$

Fourth, based on the equation 4, is a formula on how to calculate the average CPU (Central Pro- cessing Unit) memory usage. In this formula, Avg. CPU Mem. Used refers to the

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v25i2.15843

average CPU memory usage, which is calculated by summing up all the CPU memory usage values recorded by each CPU in the system. The notation Σ indicates the total summation of all recorded cpu.memoryUsed values, which represents the amount of memory used by each CPU involved. In other words, this formula accumulates the memory used by all CPUs, and the result is divided to get the average CPU memory usage across the devices. This formula is useful in monitoring and analyzing performance and memory usage in CPU-based computing systems [20].

$$Avg.Training\ Time = (\sum^{n}_{(i=1)} T\_train,i) / N \quad (5)$$

Fifth, based on equation 5, is the formula for calculating Avg. Training Time in the context of training machine learning models. In this formula, Avg. Training Time refers to the average time taken to train the model in multiple training trials or iterations. The number of symbols Σ indicates that the calculated training time is the sum of all recorded training times, i.e. T-trains, i, for each trial i, which is N trials. This sum is then divided by the total number of trials N, which gives the average training time per trial. In other words, this formula calculates the average training time by summing the time taken for each training trial and dividing it by the number of trials performed [21].

$$Avg.Validation\ Time = (\sum^{n}_{(i=1)} T\_valid,i) / N \quad (6)$$

Finally, based on equation 6, is the formula for calculating Avg. Validation Time in the context of training machine learning or deep learning models. In this formula, Avg. Validation Time refers to the average time taken to perform the model validation process in multiple iterations or trials. The notation Σ denotes the sum of the total validation time T_valid, i calculated for each trial i, with the total number of trials N. This sum is then divided by the number of trials N, to give the average validation time per trial. In other words, this formula calculates the average time taken for model validation in various trials, which allows assessment of the time efficiency of the validation process [22].

3. RESULTS AND DISCUSSION

In this section, the experimental results will be divided into four experimental results. The first experimental result is to evaluate the NER task. The second experimental result is to evaluate the SA task. The third experimental outcome is to assess the EmoT task. Finally, the experimental results are used to evaluate the HS task.

3.1 NER Task Evaluation Results

The evaluation results of the NER task on various Indonesian BERT variants are shown in Table 6. Based on Table 6, the performance evaluation results are explained as follows:

Table 6. NER Task Evaluation Results

Model	F1-Macro	Accuracy	Training Time (s)	Validation Time (s)	GPU (GB)	RAM (GB)
IndoRoBERTa	**0.878**	0.771	295.55	**10.48**	4.36	3.59
IndoNLU	0.665	0.880	288.17	**10.48**	4.34	**3.41**
IndoLEM	0.627	0.878	**285.24**	10.51	**4.03**	3.59
mBERT	0.847	**0.954**	302.14	10.50	5.36	3.60

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v25i2.15843

IndoRoBERTa achieved the highest F1-Macro score, demonstrating its strong ability to identify entity boundaries and maintain a balance between precision and recall across all entity classes. This suggests that the pre-training strategy it employs yields more robust contextual representations for sequence labeling tasks.

In contrast, mBERT achieved the highest accuracy but had a slightly lower F1-Macro score compared to IndoRoBERTa. This difference indicates that mBERT tends to prioritize the majority class, resulting in higher overall accuracy but less balanced performance across all entity categories. A similar pattern is observed in IndoNLU and IndoLEM, where relatively high accuracy is accompanied by lower F1-Macro scores.

The difference between F1-Macro and accuracy highlights the presence of class imbalance in the NER dataset. Models with higher F1-Macro scores are more effective at capturing minority entities, while models with higher accuracy tend to prioritize the dominant class.

3.2 SA Task Evaluation Results

The results of the SA task evaluation on different variants of Indonesian BERT are shown in Table 7. Based on Table 7, the performance evaluation results are explained as follows:

Table 7. SA Task Evaluation Results

Model	F1-Macro	Accuracy	Training Time (s)	Validation Time (s)	GPU (GB)	RAM (GB)
IndoRoBERTa	0.883	0.910	1399.91	**181.02**	3.95	**3.56**
IndoNLU	**0.891**	0.904	1365.27	188.85	4.04	3.57
IndoLEM	0.876	**0.914**	**1341.45**	191.24	**3.75**	3.58
mBERT	0.827	0.866	1958.82	252.29	5.58	3.62

IndoNLU achieved the highest F1-Macro score, demonstrating a strong balance between precision and recall across all sentiment classes. However, IndoLEM achieved the highest accuracy, indicating that this model performs exceptionally well on dominant sentiment classes.

The relatively small differences between IndoNLU, IndoRoBERTa, and IndoLEM suggest that monolingual Indonesian models consistently perform well for sentiment classification. In contrast, mBERT demonstrated lower performance in both F1-Macro and accuracy, indicating that multilingual representations may be less effective at capturing nuanced sentiments in Indonesian text.

In terms of efficiency, IndoLEM again demonstrates the fastest training time and lowest GPU memory usage, highlighting its suitability for resource-constrained environments. On the other hand, mBERT exhibits significantly higher training times and memory consumption, reinforcing the trade-off between model generality and computational cost.

3.3 EmoT Task Evaluation Results

The results of EmoT task evaluation on various BERT Indonesia variants are shown in Table 8. Based on Table 8, the performance evaluation results are explained as follows:

Table 8. EmoT Task Evaluation Results

Model	F1-Macro	Accuracy	Training Time (s)	Validation Time (s)	GPU (GB)	RAM (GB)
IndoRoBERTa	0.694	**0.688**	**457.63**	**56.83**	4.24	**3.67**

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v25i2.15843

IndoNLU	**0.696**	0.686	596.44	61.78	4.26	3.69
IndoLEM	0.650	0.648	604.01	63.05	**4.04**	3.71
mBERT	0.633	0.634	802.76	83.09	5.67	3.72

IndoNLU and IndoRoBERTa demonstrate comparable performance, with only very small differences in F1-Macro scores and accuracy. This suggests that both models are equally effective at capturing emotional nuances in text.

The relatively small performance gap among monolingual models suggests that task complexity, rather than model architecture, may be the limiting factor. In contrast, mBERT consistently demonstrates lower performance, further supporting the observation that multilingual models may not capture subtle emotional expressions as effectively as monolingual models.

From an efficiency perspective, IndoRoBERTa achieved the fastest training and validation times, while IndoLEM demonstrated lower memory consumption. These results highlight the trade-off between speed and memory efficiency among the evaluated models.

3.4 HS Task Evaluation Results

The results of the evaluation of HS tasks on different variants of BERT Indonesia are shown in Table 9. Based on Table 9, the performance evaluation results are described as follows:

Table 9. HS Task Evaluation Results

Model	**F1-Macro**	**Accuracy**	**Training Time (s)**	**Validation Time (s)**	**GPU (GB)**	**RAM (GB)**
IndoRoBERTa	0.848	0.847	1852.58	409.36	4.53	3.65
IndoNLU	**0.852**	**0.861**	3328.56	315.88	4.63	**3.64**
IndoLEM	0.845	0.856	**1724.01**	**273.75**	**4.19**	3.71
mBERT	0.752	0.771	2385.46	278.64	5.45	3.74

IndoNLU achieved the best performance in both F1-Macro and accuracy, demonstrating its strong capability in handling binary classification tasks with a good balance between precision and recall.

IndoRoBERTa and IndoLEM demonstrated comparable performance, indicating that monolingual models are generally effective for this task. In contrast, mBERT showed significantly lower performance, reinforcing the limitations of multilingual models in capturing context-specific linguistic patterns in Indonesian.

In terms of efficiency, IndoLEM again demonstrated the fastest training time and lowest memory usage, while mBERT required significantly greater computational resources. This further underscores the importance of considering efficiency alongside effectiveness in practical applications.

3.5 Overall Evaluation Results: Effectiveness vs Efficiency

Figure 3 presents a radar-based multi-dimensional comparison of model performance, integrating both effectiveness (F1-Macro and accuracy) and efficiency (training time, validation time, GPU, and RAM usage). All metrics are normalized, and cost-related metrics are inverted to ensure that higher values consistently indicate better performance.

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v25i2.15843

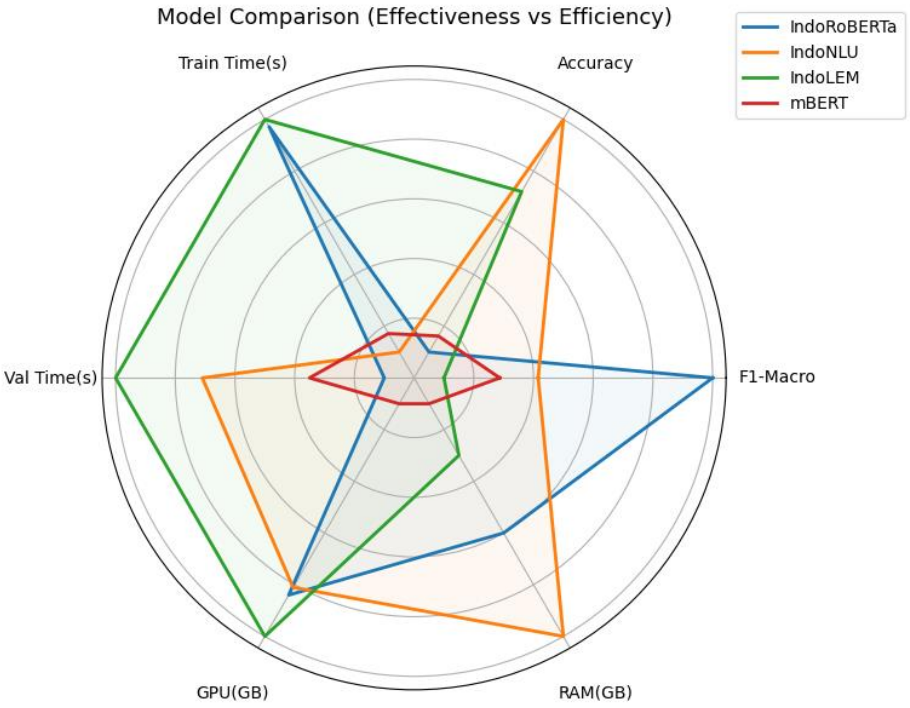

Figure 3. Overall Evaluation Results for All Models and Tasks

IndoRoBERTa demonstrates the most balanced overall performance across multiple dimensions. It achieves the highest F1-Macro score, indicating strong capability in handling class imbalance and capturing contextual semantics. Although its efficiency metrics are not the highest, they remain competitive, suggesting a favorable trade-off between effectiveness and computational cost.

IndoNLU achieves the highest accuracy among all models, indicating strong performance in predicting dominant classes. However, this advantage comes at the expense of computational efficiency, as reflected in its lower normalized scores in training and validation time as well as higher memory consumption. This suggests that IndoNLU prioritizes classification accuracy over efficiency.

IndoLEM exhibits the strongest performance in efficiency-related metrics, including training time, validation time, and GPU utilization. This indicates that IndoLEM is the most computationally efficient model among those evaluated. However, its lower effectiveness scores suggest a trade-off, where efficiency is achieved at the cost of reduced predictive performance.

In contrast, mBERT consistently shows lower performance across both effectiveness and efficiency dimensions. Its relatively weaker F1-Macro and accuracy scores indicate limitations in capturing Indonesian-specific linguistic features, while its efficiency metrics are also less favorable compared to monolingual models. This reinforces the observation that multilingual models may be less optimal for language-specific tasks.

Overall, the radar visualization highlights a clear trade-off between effectiveness and efficiency across models. IndoRoBERTa offers the most balanced performance, IndoNLU emphasizes accuracy, and IndoLEM prioritizes efficiency. These findings indicate that no single model is universally optimal, and model selection should be aligned with specific application requirements and computational constraints. This multi-dimensional evaluation further confirms that relying on a single metric may lead to suboptimal model selection, highlighting the necessity of holistic benchmarking.

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v25i2.15843

4. CONCLUSION

This study presents a comprehensive multi-task evaluation of Indonesian BERT models by jointly considering effectiveness and computational efficiency across four NLP tasks. The results show that monolingual models consistently outperform the multilingual mBERT model in both performance and efficiency.

IndoRoBERTa achieves the highest F1-Macro score, indicating strong capability in handling class imbalance, while IndoNLU attains the highest accuracy, reflecting its strength in predicting dominant classes. IndoLEM demonstrates superior efficiency, making it suitable for resource-constrained environments.

The radar-based multi-dimensional analysis highlights the trade-offs between performance and efficiency, showing that no single model is universally optimal. These findings emphasize that model selection should be aligned with specific application requirements and computational constraints rather than relying on a single evaluation metric.

Overall, this study reinforces the importance of holistic benchmarking in evaluating pretrained language models, particularly in real-world scenarios where both effectiveness and efficiency must be considered. Future work may include incorporating additional datasets, conducting repeated experiments with statistical significance analysis, and evaluating more recent pretrained models.

REFERENCES

[1] Statista Research Department, “Natural language processing market growth worldwidefrom 2021-2030,” Statista, Feb. 14, 2024. https://www.statista.com/forecasts/1449874/world- nlp-market-size-growth (accessed Jul. 06, 2024).

[2] J. Castanha, Indrawati, S. K. B. Pillai, G. Ramantoko and T. Widarmanti, ”A Systematic Literature Review on Natural Language Processing (NLP),” 2022 International Con- ference on Advanced Creative Networks and Intelligent Systems (ICACNIS), Bandung, In- donesia, 2022, pp. 1-6, doi: 10.1109/ICAC- NIS57039.2022.10055568.

[3] A. A. Abro, M. S. H. Talpur, and A. K. Jumani, “Natural Language Processing Challenges and Issues: A Literature Review”, Gazi University Journal of Science, vol. 36, no. 4, pp.1522–1536, 2023, doi: 10.35378/gujs.1032517.

[4] S. Gupta, S. Lakra and M. Kaur, ”Study on BERT Model for Hate Speech Detection,” 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2020, pp. 18, doi: 10.1109/ICECA49313.2020.9297560.

[5] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” Proceedings of the 2019 Conference of the North American Chapter of the Asso- ciation for Computational Linguistics: Human Language Technologies, Vol. 1, Jan. 2019.

[6] B. Wilie et al., “INDONLU: Benchmark and Resources for Evaluating Indonesian Natural Language Understanding,” International Joint Conference on Natural Language Processing, pp. 843–857, Sep. 2020, [Online]. Available: https://www.aclweb.org/anthology/2020.aacl-main.85.pdf

[7] F. Koto, A. Rahimi, J. H. Lau, and T. Bald- win, “IndoLEM and IndoBERT: a benchmark dataset and pre-trained language model for Indonesian NLP,” arXiv.org, Nov. 02, 2020. https://arxiv.org/abs/2011.00677v1

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” arXiv.org, Oct. 11, 2018. https://arxiv.org/abs/1810.04805

[9] S. Mohammadi and M. Chapon, ”Investigating the Performance of Fine-tuned Text Classification Models Based-on Bert,” 2020 IEEE 22nd International Conference on High

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v25i2.15843

Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Yanuca Island, Cuvu, Fiji, 2020, pp. 1252-1257.

[10] K. Taneja and J. Vashishtha, ”Comparison of Transfer Learning and Traditional Machine Learning Approach for Text Classification,” 2022 9th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2022, pp. 195-200, doi: 10.23919/INDIA- Com54597.2022.9763279.

[11] D. Vucetic, M. Tayaranian, M. Ziaeefard, J. J. Clark, B. H. Meyer and W. J. Gross, ”Efficient Fine-Tuning of BERT Models on the Edge,” 2022 IEEE International Symposium on Circuits and Systems (ISCAS), Austin, TX, USA, 2022, pp. 1838-1842, doi: 10.1109/IS-CAS48785.2022.9937567.

[12] S. O. Khairunnisa, A. Imankulova, and M. Komachi, “Towards a standardized dataset on Indonesian named entity recognition,” ACL Anthology, Dec. 01, 2020. https://aclanthology.org/2020.aacl-srw.10/

[13] F. Koto, J. H. Lau, and T. Baldwin, “IndoBERTweet: A Pretrained Language Model for Indonesian Twitter with Effective Domain- Specific Vocabulary Initialization,” arXiv.org, Sep. 10, 2021. https://arxiv.org/abs/2109.04607

[14] M. S. Saputri, R. Mahendra and M. Adriani, ”Emotion Classification on Indonesian Twitter Dataset,” 2018 International Conference on Asian Language Processing (IALP), Bandung, Indonesia, 2018, pp. 90-95, doi: 10.1109/IALP.2018.8629262.

[15] I. Alfina, R. Mulia, M. I. Fanany and Y. Ekanata, ”Hate speech detection in the Indonesian language: A dataset and preliminary study,” 2017 International Conference on Advanced Computer Science and Infotmation Systems (ICACSIS), Bali, Indone- sia, 2017, pp. 233-238, doi: 10.1109/ICAC- SIS.2017.8355039.

[16] M. O. Ibrohim and I. Budi, “Multi-label Hate Speech and Abusive Language Detection in Indonesian Twitter,” Association for Computational Linguistics, Jan. 2019, doi: 10.18653/v1/w19-3506

[17] K. De Angeli et al., “Class imbalance in out-of- distribution datasets: Improving the robustness of the TextCNN for the classification of rare cancer types,” Journal of Biomedical Informat- ics, vol. 125, p. 103957, Jan. 2022.

[18] J. M. Johnson and T. M. Khoshgoftaar, “Survey on deep learning with class imbalance,” Journal of Big Data, vol. 6, no. 1, Mar. 2019, doi: 10.1186/s40537-019-0192-5.

[19] J. Opitz and S. Burst, “Makro F1 and Makro F1,” arXiv.org, Nov. 08, 2019. https://arxiv.org/abs/1911.03347

[20] Y. Gao et al., “An Empirical Study on Low GPU Utilization of Deep Learning Jobs,” Association for Computing Machinery, pp. 1–13, Apr. 2024, doi: 10.1145/3597503.3639232.

[21] E. Badidi and D. Gopinathan, “On the CPU Usage of Deep Learning Models on an Edge Device,” in Lecture notes in networks and systems, 2023, pp. 209–219. doi: 10.1007/978-3-031-21438-7 18.

[22] L. Zancato, A. Achille, A. Ravichandran, R. Bhotika, and S. Soatto, “Predicting training time without training,” arXiv.org, Aug. 28, 2020. https://arxiv.org/abs/2008.12478

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v25i2.15843