IJCCS (Indonesian Journal of Computing and Cybernetics Systems)
Vol.x, No.x, July xxxx, pp. 1~5
ISSN (print): 1978-1520, ISSN (online): 2460-7258
DOI: 10.22146/ijccs.xxxx

IndoBERT Optimization for Sentiment Analysis on DeepSeek App Reviews

Muh. Sunan*[1], Unique Desyrre A. Resiloy[2], Desy Endriani[3], Cici Suhaeni[4], Bagus Sartono[5], Gery Alfa Dito[6]
[1,2,3,4,5,6]Department of Statistic and Data Science, SSMI IPB University, Bogor, Indonesia
e-mail: ***[1]muh.sunan@apps.ipb.ac.id**, [2]uniqueda_resiloy@apps.ipb.ac.id, [3]endrianidesy@apps.ipb.ac.id, [4]cici_suhaeni@apps.ipb.ac.id, [5]bagusco@gmail.com, [6]gerrydito@apps.ipb.ac.id

Abstrak

Dalam era digital, analisis sentimen menjadi krusial untuk mengevaluasi opini publik, khususnya dalam konteks aplikasi Play Store dengan ulasan berbahasa Indonesia. Penelitian ini bertujuan untuk meningkatkan kinerja model IndoBERT dalam analisis sentimen ulasan aplikasi DeepSeek dengan memanfaatkan teknik augmentasi data dan penyetelan hyperparameter. Augmentasi data dilakukan melalui teknik back-translation, sementara hyperparameter yang diuji meliputi jumlah epoch, learning rate, dan batch size. Hasil eksperimen menunjukkan bahwa kombinasi augmentasi data dengan epoch 10, learning rate 2e-5, dan batch size 16 menghasilkan akurasi tertinggi sebesar 93,95% dan F1-score 0,94, dengan stabilitas yang lebih baik dibandingkan model tanpa augmentasi. Model tanpa augmentasi menunjukkan fluktuasi kinerja, mengindikasikan adanya overfitting pada beberapa konfigurasi. Temuan ini menegaskan pentingnya penerapan teknik augmentasi dan penyetelan hyperparameter dalam meningkatkan akurasi dan stabilitas model analisis sentimen, serta berkontribusi pada pengembangan model NLP untuk bahasa Indonesia dan bahasa lain yang memiliki sumber daya terbatas.

Kata kunci*— Analisis Sentimen, IndoBERT, Data Augmentasi, Hyperparameter*

Abstract

In the digital era, sentiment analysis has become crucial to evaluate public opinion, especially in the context of Play Store apps with Indonesian-language reviews. This research aims to improve the performance of the IndoBERT model in sentiment analysis of DeepSeek app reviews by using data augmentation and hyperparameter tuning techniques. Data augmentation is done through the back-translation technique, while the hyperparameters tested include the number of epochs, learning rate, and batch size. Experimental results show that the combination of data augmentation with epoch 10, learning rate 2e-5, and batch size 16 produces the highest accuracy of 93.95% and F1-score of 0.94, with better stability than the model without augmentation. The model without augmentation showed fluctuations in performance, indicating overfitting in some configurations. These findings confirm the importance of applying augmentation techniques and hyperparameter tuning in improving the accuracy and stability of sentiment analysis models, and contribute to the development of NLP models for Indonesian and other resource-constrained languages.

Keywords*— Sentiment Analysis, IndoBERT, Data Augmentation, Hyperparameter*

Received June 1st,2012; Revised June 25th, 2012; Accepted July 10th, 2012

1. INTRODUCTION

In the digital era, public opinion is disseminated via social media, discussion forums, and online reviews. Opinion quantification has emerged as a pivotal strategy across diverse domains, including business, public policy, and scientific research [1]. Sentiment analysis, also known as opinion mining, has emerged as a method capable of extracting and interpreting text-based public opinion. This approach has been applied in various domains, including financial markets, healthcare, marketing, politics, and disaster mitigation [2]. Although Natural Language Processing (NLP) technology continues to evolve, one of the main challenges in sentiment analysis remains the limitation of language models in capturing local nuances, especially in Bahasa Indonesia. The development of Transformer-based NLP technology has provided significant progress, especially through the BERT (Bidirectional Encoder Representations from Transformers) model introduced by Devlin et al. (2019) [3]. The model employs a two-way contextual representation and can be fine-tuned flexibly for various NLP tasks. For the local context, the IndoBERT model was developed as a pre-trained version of BERT in Bahasa Indonesia [4]. Various studies have shown that IndoBERT can provide superior performance on text classification, entity extraction, and sentiment analysis tasks in Bahasa Indonesia [5][6]. However, the performance of IndoBERT can be further enhanced through the implementation of additional optimization strategies, such as data augmentation and hyperparameter tuning.

Data augmentation techniques have been developed to enrich the linguistic variety in the training data [7]. By adding diverse text examples, this method supports the model in learning a more general and robust representation of language [8]. To illustrate, word substitutions, wherein lexical items are supplanted with their synonymous counterparts along with back-translations, a linguistic process that entails translating text into a foreign language and subsequently back into the original language, has demonstrated efficacy in augmenting the vector space of text representation without altering the core semantic content [9]. This technique is proven to improve the generalization ability of the model in recognizing more complex sentiment patterns [10][11].

In addition to data augmentation techniques, the process of hyperparameter tuning plays an instrumental role in enhancing the training efficiency of transformer models [12]. The effectiveness of the training process is contingent on key parameters such as the number of epochs, learning rate, and batch size. Inappropriate hyperparameter settings have the potential to compromise the efficiency of the training process, manifesting in suboptimal outcomes such as delayed convergence or the complete failure to reach optimal results. For instance, an insufficient number of epochs can result in underfitting, a scenario in which the model is unable to discern salient patterns from the training data. Conversely, the overuse of epochs can lead to overfitting, defined as the model's adaptation to the training data, resulting in a loss of its capacity to generalize to new, previously unseen data [13]. Several studies have shown that the combination of augmentation and tuning techniques results in significant performance improvements in sentiment analysis. Huong & Hoang's (2020) research successfully improved the accuracy of Vietnamese sentiment classification by applying data augmentation to the BERT model. Another study by Elgeldawi et al. (2021) also emphasized that proper hyperparameter tuning can improve sentiment analysis performance in Arabic [14][15].

However, research on the application of augmentation and tuning optimization strategies in IndoBERT in the context of Indonesian-language app reviews is still rare. This condition opens up research opportunities that this study aims to answer. This research focuses its case study on user reviews of DeepSeek, an AI assistant application available on the Google Play Store. This app was chosen due to the large number of public reviews that reflect users' opinions on the app's features and performance. These reviews contain valuable information that can be utilized for product evaluation and development decision-making. By utilizing IndoBERT-based sentiment analysis, this research aims to systematically evaluate public perception of the app. A combination of data augmentation and hyperparameter tuning strategies was chosen as they have complementary characteristics, where augmentation enriches the diversity of input representations, while tuning allows the model to learn these patterns more efficiently and purposefully.

This combined approach is expected to improve prediction accuracy while reducing the risk of overfitting the training data. Through the applied approach, this research is expected to make a significant empirical contribution to the development of more accurate and reliable sentiment analysis models in Bahasa Indonesia, particularly in the context of app review processing. In addition, the findings from this research also have the potential to be an important reference for the development of NLP systems in languages with limited resources, such as Bahasa Indonesia.

2. METHODS

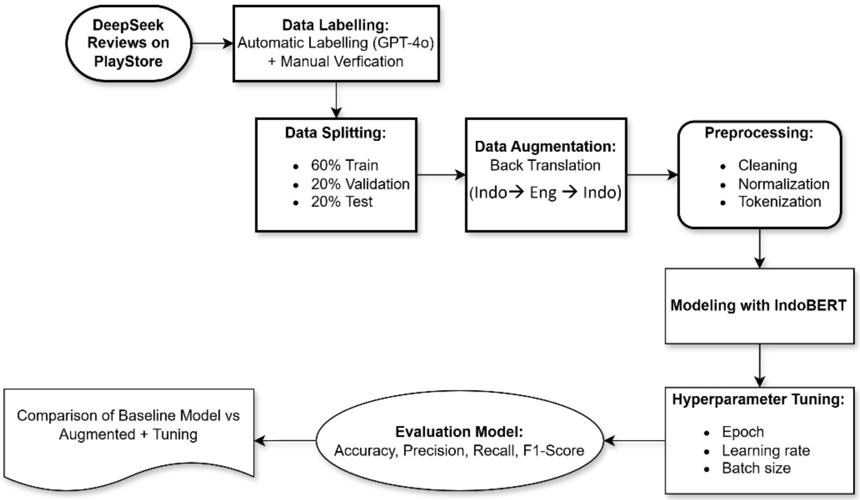
DeepSeek Reviews on PlayStore
Data Labelling: Automatic Labelling (GPT-4o) + Manual Verfication
Data Splitting:
60% Train
20% Validation
20% Test
Data Augmentation: Back Translation (Indo→ Eng → Indo)
Preprocessing:
Cleaning
Normalization
Tokenization
Modeling with IndoBERT
Hyperparameter Tuning:
Epoch
Learning rate
Batch size
Evaluation Model: Accuracy, Precision, Recall, F1-Score
Comparison of Baseline Model vs Augmented + Tuning

Figure 1 Research Methodology

This research details the methodology for developing a sentiment classification model for Deepseek app reviews using the IndoBERT architecture. The stages include data collection and processing, data augmentation with back-translation, model training, hyperparameter optimization, and performance evaluation. The primary focus is on comparing the baseline model against the model optimized with these techniques. The entire process was conducted using Python in a Jupyter Notebook environment, with the workflow illustrated in Figure 1.

2.1 Dataset

The initial stage of this research was data collection in the form of Indonesian-language reviews from the Deepseek application on the Google Play Store platform. This process was done automatically using web scraping techniques with Python libraries, covering the period from January to May 2025, which resulted in a total of 2,672 review entries. This dataset became the main source in training and testing the sentiment classification model. Examples of documents obtained from the scarping process can be seen in Table 1.

Table 1 Example of DeepSeek Scraping Data on Google Playstore

Index	Review Text
1	Suka busy (*like being busy*)
2	bagus banget responsif dan jawabannya masuk akal serta tidak bertele tele jawabnya (*Very good responsive and the answers make sense not long-winded*)

3	respon yang cepat dan tepat👍👍 (*quick and precise response*)
4	jawaban yang diberikan selalu benar hebat (*the answers given are always correct, amazing*)
5	Sangat membantu dan menambah pengetahuan dalam segala hal (*very helpful and adds knowledge in all matters*)

2.2 Sentiment Labeling

The collected data was automatically labeled using the GPT-4o language model. GPT-4o is part of the OpenAI GPT generative language model development, which has been widely used in various natural language processing applications, including text classification and information extraction [16]. Engineering prompts guide the model to classify each review into three sentiment categories: positive, negative, and neutral. Out of the total 2,672 reviews, 951 (35.59%) were labeled as positive, 901 (33.72%) as neutral, and 820 (30.69%) as negative. The distribution of sentiment classes can be seen in Figure 2.

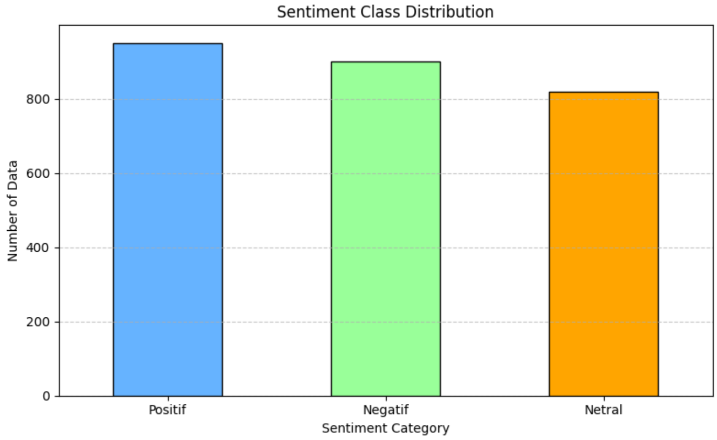

Figure 2 Sentiment Class Distribution

To evaluate the reliability of manual verification against GPT-4o labels, we computed inter-annotator agreement using Cohen's Kappa, obtaining a value of 0.959. This indicates almost perfect agreement between the automatic and manual annotations, confirming that the ground truth labels used in training are highly reliable.

2.3 Data Splitting

The labeled datasets were divided into three subsets for training and model evaluation purposes. The proportion is 60% training data, 20% validation data, and 20% testing data. The division technique is done by stratification, which ensures that the distribution of sentiment labels remains balanced in each subset, to avoid class distribution bias during training and evaluation [17].

2.4 Data Augmentation

To increase the diversity and generalization ability of the model, data augmentation techniques are applied to a subset of the training data. One of the augmentation techniques used is back-translation, which involves translating text to another language and then translating it back to the original language [18]. In this research, MarianMT from the HuggingFace library is used for the augmentation process [19]. MarianMT is a neural machine translation model based on a sequence-to-sequence architecture that can support multiple language pairs, including Bahasa Indonesia and English [20]. The back-translation process helps generate text variations that retain the core meaning, thus enriching the training data used without requiring additional data collection. An example of the results of data augmentation is shown in Table 2.

Table 2 Example of Data Augmentation Results Using Back-Translation Technique

Index	Original Text	Translate Text
1	Parah jangan didownload hanya sesaat aja sesudah itu gak ada respon lagi ,gak tau berbayar atau gratis ,sehrsnya yg gratis itu bisa dipakai,,,, pokoknya parah ai deepseek,lebih bagus ai chatgpt gak ada batasan kita mau chat kapan aja bisa,Negatif	itu buruk jangan di unduh sebentar setelah itu, tidak ada lagi respon, tidak ada bayaran atau gratis.
2	deepseek kadang suka ga respon jadi tolong diperbaiki respon dan integrasi nya	Pencarian mendalam kadang-kadang tidak merespon, jadi tolong perbaiki respon dan integrasi.
3	ini sangat terbest, semoga update yg terbaru semakin baik, dan jangan ada batasan dalam chat ya tolong banget	Ini sangat menarik, berharap update terbaru menjadi lebih baik, dan tidak ada batas untuk chatting.

To ensure that the sentiment polarity of the augmented texts remained consistent with the original labels, we manually verified a set of augmented reviews. This manual inspection confirmed that the augmented texts preserved the intended sentiment polarity of the original sentences, thereby ensuring the validity of the labels used in training.

2.5 Data Preprocessing

Before the data can be used in the model, it needs to undergo a series of steps to prepare it. This process has several important steps to ensure the data is in a clean format and ready for the model training.

2.5.1 Cleaning Text

In this section, elements that may interfere with the analysis, such as URLs, HTML tags, non-alphabetic characters, non-contextual punctuation, and excess spaces, are removed. Next, the entire text is converted to lowercase to reduce unnecessary text format diversity. In addition, repetitive character normalization was performed (e.g., changing the word “baguuuus” to “bagus”) to improve data consistency [21].

2.5.2 Normalization Text

Word normalization involves the replacement of nonstandard or informal words with standard Indonesian forms. This process utilizes the standardized word dictionary accessible on the Kaggle platform, which provides pairs of nonstandard words and their corresponding standard equivalents. Word normalization is a process that aims to equalize the language variations used in the text so that the model can process the information more accurately [22].

2.5.3 Tokenization

After the review text is cleaned and normalized, it is then tokenized. In this research, tokenization is done using the official IndoBERT tokenizer, which is a WordPiece-based model that has been trained with the Indonesian corpus in the IndoBERT pretraining process. Unlike the English version of the BERT tokenizer, the IndoBERT tokenizer has a vocabulary and sub-word segmentation specifically tailored to the language structure, vocabulary, and morphology of Bahasa Indonesia [4].

2.6 Modeling

The main model used in this research is IndoBERT, a pre-trained language model based on BERT architecture developed by IndoNLU and optimized for Bahasa Indonesia. IndoBERT has been trained on a large corpus consisting of various text domains, such as news, social media, and other formal data, so it has good semantic representation capabilities. The model is proven to be effective for various natural language processing (NLP) tasks in Bahasa Indonesia, including sentiment classification [23].

2.7 Hyperparameter Tuning

To obtain the optimal model configuration for sentiment classification, the experiments were conducted using Google Colab equipped with an T4 GPU, and 12 Gb system RAM. The three main hyperparameter tuning tested were the number of epochs, learning rate, and batch size [24]. Epochs were tested at 3, 5, and 10 epochs to evaluate the effect of the number of iterations on the model convergence process. Furthermore, for the learning rate, the two values tested were 2e-5 and 5e-5 to see the model's sensitivity to the speed of weight update. In batch size, two batch size configurations were used, 16 and 32, which affect the stability and speed of training. This experiment aims to evaluate the impact of parameter configuration on model performance, both in terms of convergence speed and classification accuracy. Hyperparameter tuning is critical to ensure that the model not only performs well in the training data, but can also generalize well to unseen data [25].

2.8 Evaluation Model

Model evaluation is performed on a subset of test data to measure the final performance of each model variant. The assessment uses four main evaluation metrics: accuracy, precision, recall, and F1-Score, which are commonly used in multi-class classification tasks.

Accuracy is a proportion that indicates how correct the model is overall, by calculating the percentage of correct predictions out of the total data, as expressed in the following equation:

$$Accuracy = \frac{TP + TN}{(TP + FN + FP + TN)} \tag{1}$$

Precision is a measure of how many of the positive predictions generated by the model are positive, defined as:

$$Precision = \frac{TP}{(TP + FP)} \tag{2}$$

Sensitivity, also known as "recall," is a metric that quantifies the model's ability to detect true positive cases, defined as:

$$Recall = \frac{TP}{(TP + FN)} \tag{3}$$

F1-score is a metric that combines precision and sensitivity into a single value by taking the average of the two [26] It is calculated as:

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

Next, a performance comparison was made between two approaches: a model without augmentation and hyperparameter tuning techniques and a model using both techniques. The goal is to determine how much influence the augmentation and tuning strategies have on improving the model's overall performance.

In addition to accuracy, precision, recall, and F1-Score, we also recorded the average training time for each model variant to provide further insight into computational efficiency. This comparison allows us to assess not only the predictive performance but also the practical feasibility of deploying the models.

3. RESULTS AND DISCUSSION

3.1 Experiment Results

During the testing phase, 24 hyperparameter combinations were evaluated, considering the number of epochs (3, 5, 10), learning rate (2e-5, 5e-5), and batch size (16, 32). The experiments were conducted using two distinct datasets: the original training data and the augmented data. The results in Table 3 present the IndoBERT model's performance in accuracy, F1 score, precision, and recall.

Table 3 Test results of the IndoBERT model with different hyperparameters on the training data, with and without augmentation.

Data	**Epoch**	**Learning rate**	**Batch Size**	**Accuracy**	**F1-Score**	**Precision**	**Recall**
Training Data with Augmentation	3	2e-5	16	0,9263	0,93	0,93	0,93
	3	2e-5	32	0,9319	0,93	0,93	0,93
	3	5e-5	16	0,913	0,91	0,91	0,91
	3	5e-5	32	0,9149	0,91	0,91	0,91
	5	2e-5	16	0,9338	0,93	0,93	0,93
	5	2e-5	32	0,9338	0,93	0,93	0,93
	5	5e-5	16	0,9055	0,91	0,9	0,91
	5	5e-5	32	0,9225	0,92	0,93	0,92
	10	2e-5	16	0,9395	0,94	0,94	0,94
	10	2e-5	32	0,9225	0,92	0,92	0,92
	10	5e-5	16	0,9187	0,92	0,92	0,92
	10	5e-5	32	0,9263	0,93	0,93	0,93
Training Data without Augmentation	3	2e-5	16	0,8639	0,86	0,87	0,86
	3	2e-5	32	0,8733	0,87	0,88	0,87
	3	5e-5	16	0,8318	0,83	0,83	0,83
	3	5e-5	32	0,8601	0,86	0,86	0,86
	5	2e-5	16	0,8771	0,88	0,88	0,88
	5	2e-5	32	0,8752	0,88	0,88	0,88
	5	5e-5	16	0,8469	0,85	0,87	0,85
	5	5e-5	32	0,8866	0,89	0,89	0,89
	10	2e-5	16	0,8904	0,89	0,89	0,89
	10	2e-5	32	0,879	0,88	0,88	0,88
	10	5e-5	16	0,8809	0,88	0,88	0,88
	10	5e-5	32	0,8166	0,82	0,83	0,82
Random Forest + TF-IDF				0,853	0,851	0,854	0,853

Based on Table 3, the IndoBERT model trained on augmented data with the best hyperparameter configuration (epoch = 10, learning rate = 2e-5, batch size = 16) achieved the highest performance with an accuracy of 93.95% and F1-score of 0.94. In contrast, the model trained on non-augmented data performed best with an accuracy of 89.04% with the same hyperparameter configuration. As evidenced by the more noticeable accuracy fluctuations, which varied from from 81.66% to 89.04%, this experiment shows that the model is more sensitive to changes in hyperparameters when trained on non-augmented data. On the other hand, with an accuracy range of 90.55% to 93.95%, the model trained with enhanced data showed a more consistent accuracy. Data augmentation not only improves performance but also stabilizes training outcomes.

As a traditional baseline, the Random Forest classifier was also evaluated. The Random Forest baseline achieved an accuracy of 85.30% and an F1-score of 85.10%. While lower than IndoBERT, this baseline provides a meaningful reference point, confirming that transformer-based models offer significant improvements over traditional machine learning methods for Indonesian sentiment analysis.

Additionally, models with a 16-batch size were more accurate than those with a 32-batch size, consistent with research by Keskar et al. (2017) that found that smaller batch sizes support improved generalization [27]. A learning rate 5e-5 was the most effective, speeding convergence while maintaining stability. While the enhanced data showed a more notable improvement, the combination of epoch = 10, learning rate = 5e-5, and batch size = 16 produced the most outstanding results across all datasets, highlighting the critical significance of data augmentation in training parameter optimization.

3.2 Analysis of the Effect of Data Augmentation

Based on the experimental results, there is a significant difference between models that use data augmentation and those that do not, as seen in Figure 3. Figure 3 shows that the model with augmentation has a narrower accuracy range (90.55% to 93.95%), indicating higher stability.

On the other hand, the accuracy range of the model without augmentation is broader (81.66% to 89.04%), suggesting more variation in the outcomes. This indicates that data augmentation works well to increase model performance stability, particularly under less-than-ideal training circumstances. However, as the same chart illustrates, augmentation also raises the F1-score. The F1-score values of augmentation models are higher and more stable, indicating a better trade-off between recall and accuracy. This enhancement is necessary for classification jobs in order to make the model responsive to pertinent input and accurate. Overall, these results show that data augmentation increases accuracy and improves the balance between the two metrics, affecting the model's classification effectiveness.

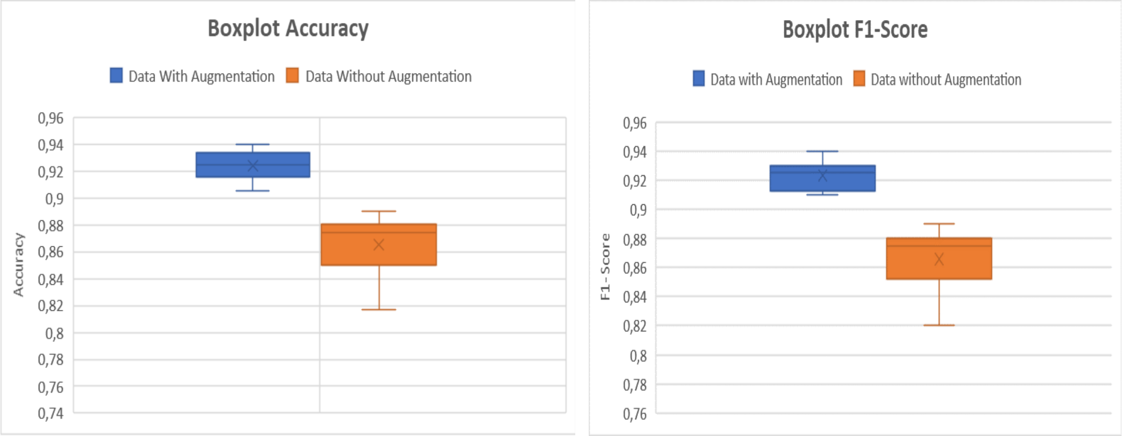

Figure 3 Boxplot of Model Accuracy and F1-Score with and without Augmentation

To verify whether the performance improvements from data augmentation were statistically significant, we applied a paired t-test on the accuracy values across all hyperparameter configurations. Each pair consisted of results from the same configuration (epoch, learning rate, batch size) trained with and without augmentation, evaluated on the same test set. The paired t-test results showed that the augmented models significantly outperformed the non-augmented models ($p < 0.05$), indicating that the observed improvements were not due to random chance.

Table 4 Results of Paired t-test Comparing Accuracy of Models With and Without Data Augmentation

Comparison	Mean		t-statistic	p-value	Significance
	Without Augmentation	With Augmentation			
All configs (n=12 pairs)	0,866	0,924	10,181	0,000001	Significant

3.3 *Analysis of the Effect of Hyperparameters*

Based on the experimental results, the combination of three main hyperparameters, namely the number of epochs, learning rate, and batch size, showed a significant effect on the model performance, both on augmented and un-augmented data. On epochs, increasing the number of training iterations from 3 to 10 improves the model performance, especially when used with augmented data. This can be seen in Figure 4, where the accuracy of the model with augmented data steadily increases as the epochs increase. However, on non-augmented data, increasing epochs is not always directly proportional to the performance improvement. Figure 5 shows a drastic decrease in some configurations, such as epoch 10, learning rate 5e-5, batch size 32, which indicates symptoms of overfitting due to overexploitation of limited data patterns.

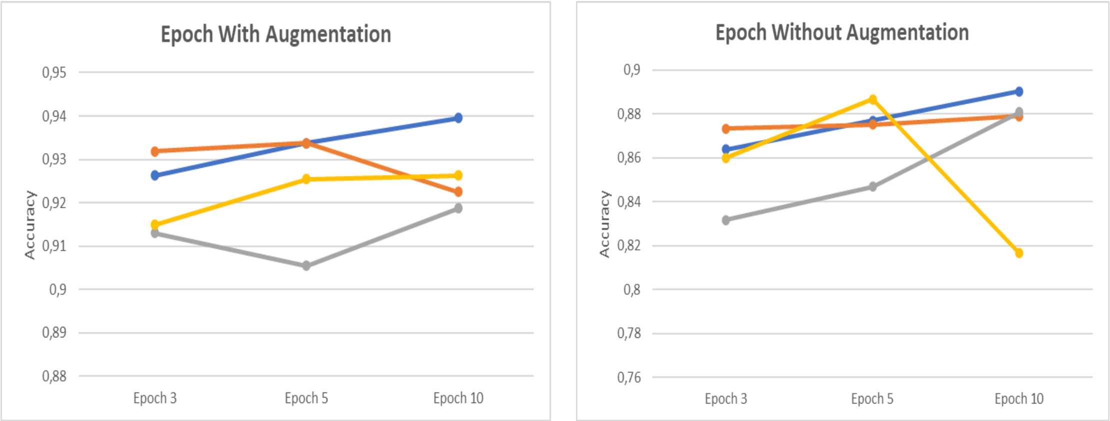

Figure 4 Epoch vs Accuracy with and without Augmentation

Learning rate also has a big impact on performance. A value of 2e-5 gives more stable and better results than 5e-5, both on augmented and non-augmented data. As seen in Figure 5, the 2e-5 learning rate supports smoother weight updates, preventing the unstable convergence that often occurs with larger learning rates. On the non-augmented data, a learning rate of 5e-5 causes unstable convergence, leading to lower performance.

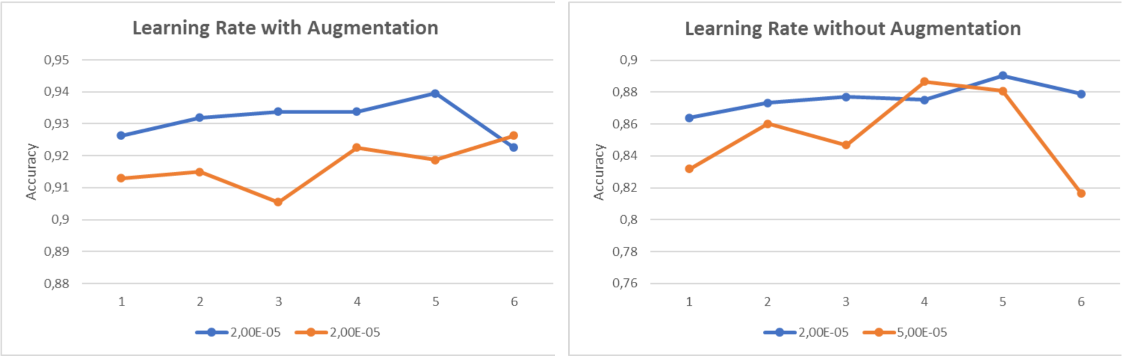

Figure 5 Learning Rate vs Accuracy with and without Augmentation

On the batch size, the augmented data shows that a batch size of 32 provides good performance stability, although a batch size of 16 still produces the best accuracy. On the non-augmented data, that a large batch size actually decreases performance, due to the lack of data variation that makes learning homogeneous. In contrast, a small batch size (16) provides a more diverse gradient, which improves the model's ability to generalize, as seen in Figure 6.

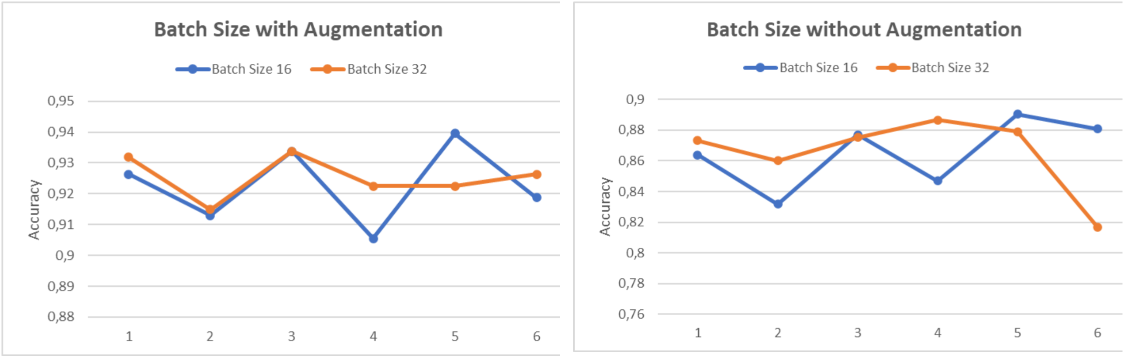

Figure 6 Batch Size vs Accuracy with and without Augmentation

The interaction between these hyperparameters demonstrates a robust synergy. The optimal configuration was determined to be a combination of high epoch, minimal learning rate, and a modest batch size in conjunction with augmented data. Conversely, the combination of a substantial learning rate and a large batch size on non-augmented data yielded unstable performance. This underscores the importance of achieving an optimal balance between model parameters to ensure the efficacy of the transformer model.

From a computational perspective, the integration of data augmentation and hyperparameter tuning increases the training time and computational resource requirements. In this study, the augmented dataset required approximately 50% longer training time compared to the original dataset, due to the increased data volume. For deployment in real-world applications, especially on devices with limited processing power, model size and inference latency need to be considered to ensure responsiveness and scalability.

Regarding the augmentation strategy, while back-translation effectively enriches linguistic diversity without altering semantic meaning, it also has limitations. The quality of the augmented data depends heavily on the translation model, which may introduce subtle semantic shifts or unnatural phrasing. Moreover, back-translation can increase dataset redundancy if the generated variations are too similar to the original sentences. Future work may explore alternative or complementary augmentation methods, such as synonym replacement, contextual word embeddings, or paraphrase generation, to address these limitations.

4. CONCLUSIONS

This study demonstrates that combining data augmentation and hyperparameter tuning enhances the performance of the IndoBERT model in sentiment analysis of DeepSeek app reviews. Data augmentation improves accuracy and F1-score stability, while the optimal hyperparameters epoch 10, learning rate 2e-5, and batch size 16 deliver the best results with augmented data. The model with augmented data showed more stable performance compared to the non-augmented model, which exhibited greater fluctuations and overfitting. Additionally, using a large batch size on non-augmented data degraded performance, emphasizing the importance of balancing parameters for effective fine-tuning. For future research, exploring diverse data augmentation techniques and applying Bayesian optimization for hyperparameter tuning can further enhance model performance. Additionally, testing models like XLNet or RoBERTa could provide valuable comparisons with IndoBERT, particularly in the context of

Indonesian. Beyond the specific case of sentiment analysis on the DeepSeek application, the findings of this study have broader implications for other Indonesian NLP tasks, such as topic classification, question answering, and named entity recognition. Furthermore, the optimization strategies applied here can be adapted for transfer learning to other Southeast Asian languages that share linguistic similarities or face similar challenges in terms of resource limitations, such as Malay, Tagalog, and Thai.

REFERENCES

[1] X. Du, M. Kowalski, A. S. Varde, G. de Melo, and R. W. Taylor, "Public opinion matters: mining social media text for environmental management," *SIGWEB Newsl.*, vol. 2019, no. Autumn, Feb. 2020, doi: 10.1145/3352683.3352688.

[2] M. Rodríguez-Ibánez, A. Casánez-Ventura, F. Castejón-Mateos, and P. M. Cuenca-Jiménez, "A review on sentiment analysis from social media platforms," *Expert Syst. Appl.*, vol. 223, no. July, 2023, doi: 10.1016/j.eswa.2023.119862.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "{BERT}: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North {A}merican Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Jun. 2019, pp. 4171–4186. doi: 10.18653/v1/N19-1423.

[4] F. Koto, A. Rahimi, J. H. Lau, and T. Baldwin, "{I}ndo{LEM} and {I}ndo{BERT}: A Benchmark Dataset and Pre-trained Language Model for {I}ndonesian {NLP}," in *Proceedings of the 28th International Conference on Computational Linguistics*, Dec. 2020, pp. 757–770. doi: 10.18653/v1/2020.coling-main.66.

[5] D. C. Febrianto, M. A. Fitrian, M. Afrad, and M. A. Khadija, "ASPECT BASED SENTIMENT ANALYSIS MENGGUNAKAN INDOBERT MODEL Melek IT," vol. 10, no. 2, pp. 157–166, 2024.

[6] C. J. L. Tobing, I. G. N. L. Wijayakusuma, and L. P. Ida, "Perbandingan Kinerja IndoBERT dan MBERT untuk Deteksi Berita Hoaks Politik dalam Bahasa Indonesia," vol. 14, no. 1, pp. 114–123, 2025.

[7] C. Shorten, T. M. Khoshgoftaar, and B. Furht, "Text Data Augmentation for Deep Learning," *J. Big Data*, vol. 8, no. 1, p. 101, 2021, doi: 10.1186/s40537-021-00492-0.

[8] C. Coulombe, "Text Data Augmentation Made Simple By Leveraging NLP Cloud APIs," pp. 1–33, 2018, [Online]. Available: http://arxiv.org/abs/1812.04718

[9] M. Regina, M. Meyer, and S. Goutal, "Text Data Augmentation: Towards better detection of spear-phishing emails," pp. 1–22, 2020, [Online]. Available: http://arxiv.org/abs/2007.02033

[10] H.-T. Duong and T.-A. Nguyen-Thi, "A review: preprocessing techniques and data augmentation for sentiment analysis," *Comput. Soc. Networks*, vol. 8, no. 1, p. 1, 2021, doi: 10.1186/s40649-020-00080-x.

[11] M. Bayer, M. A. Kaufhold, and C. Reuter, "A Survey on Data Augmentation for Text Classification," *ACM Comput. Surv.*, vol. 55, no. 7, pp. 1–44, 2022, doi: 10.1145/3544558.

[12] M. Martinez, "The Impact of Hyperparameters on Large Language Model Inference Performance: An Evaluation of vLLM and HuggingFace Pipelines," pp. 1–12, 2024, [Online]. Available: http://arxiv.org/abs/2408.01050

[13] L. N. Smith, "A disciplined approach to neural network hyper-parameters: Part 1 -- learning rate, batch size, momentum, and weight decay," pp. 1–21, 2018, [Online]. Available: http://arxiv.org/abs/1803.09820

[14] T. H. Huong and V. T. Hoang, "A data augmentation technique based on text for Vietnamese sentiment analysis," 2020. doi: 10.1145/3406601.3406618.

[15] E. Elgeldawi, A. Sayed, A. R. Galal, and A. M. Zaki, "Hyperparameter tuning for machine learning algorithms used for arabic sentiment analysis," *Informatics*, vol. 8, no. 4, pp. 1–21, 2021, doi: 10.3390/informatics8040079.

[16] OpenAI, "Hello GPT-4o," 2023. https://openai.com/index/hello-gpt-4o/

[17] M. Lango, "Tackling the Problem of Class Imbalance in Multi-class Sentiment Classification: An Experimental Study," *Found. Comput. Decis. Sci.*, vol. 44, no. 2, pp. 151–178, 2019, doi: 10.2478/fcds-2019-0009.

[18] I. M. Subedi, M. Singh, V. Ramasamy, and G. S. Walia, "Application of back-translation: a transfer learning approach to identify ambiguous software requirements," in *Proceedings of the 2021 ACM Southeast Conference*, 2021, pp. 130–137. doi: 10.1145/3409334.3452068.

[19] J. Tiedemann, "Marianmt Model," *HuggingFace*. https://huggingface.co/docs/transformers/model_doc/marian

[20] M. Junczys-Dowmunt *et al.*, "Marian: Fast neural machine translation in c++," *ACL 2018 - 56th Annu. Meet. Assoc. Comput. Linguist. Proc. Syst. Demonstr.*, pp. 116–121, 2015, doi: 10.18653/v1/p18-4020.

[21] Louis Hickman, Stuti Thapa, Louis Tay, Mengyang Cao, and Padmini Srinivasan, "Text Preprocessing for Text Mining in Organizational Research: Review and Recommendations," *Organ. Res. Methods*, vol. 25, no. 1, pp. 114–146, Nov. 2020, doi: 10.1177/1094428120971683.

[22] R. Zarnoufi, H. Jaafar, and M. Abik, "Machine Normalization: Bringing Social Media Text from Non-Standard to Standard Form," *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, vol. 19, no. 4, Apr. 2020, doi: 10.1145/3378414.

[23] B. Wilie *et al.*, "{I}ndo{NLU}: Benchmark and Resources for Evaluating {I}ndonesian Natural Language Understanding," in *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, Dec. 2020, pp. 843–857. doi: 10.18653/v1/2020.aacl-main.85.

[24] H. Jin, W. Wei, X. Wang, W. Zhang, and Y. Wu, "Rethinking Learning Rate Tuning in the Era of Large Language Models," *Proc. - 2023 IEEE 5th Int. Conf. Cogn. Mach. Intell. CogMI 2023*, pp. 112–121, 2023, doi: 10.1109/CogMI58952.2023.00025.

[25] T. Yu and H. Zhu, "Hyper-Parameter Optimization: A Review of Algorithms and Applications," pp. 1–56, 2020, [Online]. Available: http://arxiv.org/abs/2003.05689

[26] C. Cabo, C. Ordóñez, F. Sáchez-Lasheras, J. Roca-Pardiñas, and A. J. de Cos-Juez, "Multiscale Supervised Classification of Point Clouds with Urban and Forest Applications.," *Sensors (Basel).*, vol. 19, no. 20, Oct. 2019, doi: 10.3390/s19204523.

[27] N. S. Keskar, J. Nocedal, P. T. P. Tang, D. Mudigere, and M. Smelyanskiy, "On large-batch training for deep learning: Generalization gap and sharp minima," *5th Int. Conf. Learn. Represent. ICLR 2017 - Conf. Track Proc.*, pp. 1–16, 2017.