Matrik: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer
Vol. 25, No. 2, March 2026, pp. 311∼322
ISSN: 2476-9843, accredited by Kemenristekdikti, Decree No: 10/C/C3/DT.05.00/2025
DOI: 10.30812/matrik.v25i2.5847

Comparative Analysis of Indonesian Pre-trained BERT Models for the Extractive Question Answering Task on an Indonesian-Translated SQuAD Dataset

Fattah Al Ilmi Suhendra[1], **Astie Darmayantie**[1], **Adang Suhendra**[1], **Pa Pa Min**[2]
[1]Universitas Gunadarma, Depok, Indonesia
[2]Multimedia University, Melaka, Malaysia

Article Info

Article history:

Received October 20, 2025
Revised December 11, 2025
Accepted February 24, 2026

Keywords:

Fine-tuning;
IndoBERT;
Natural Language Processing;
Pre-training;
Questioning-Answering.

ABSTRACT

Transformer-based architectures have significantly advanced Natural Language Processing (NLP), with Bidirectional Encoder Representations from Transformers (BERT) serving as a strong baseline for extractive Question Answering (QA). This study aims to evaluate the performance of Indonesian BERT models on extractive QA tasks and to identify the most effective model for low-resource language settings. This research employed a comparative experimental method using two Indonesian BERT variants: indobert-base-uncased (IndoLEM) and indobert-base-p1 (IndoNLU/IndoBenchmark). Both models were fine-tuned on an Indonesian version of SQuAD 2.0, automatically translated via the Google Translate API. Answer-span alignment errors caused by translation were corrected using fuzzy string matching. Evaluation was conducted under identical hyperparameter settings and training schemes, using Exact Match (EM) and F1-score as performance metrics. The results indicate that IndoLEM achieved superior performance, with better loss convergence and a higher F1-score (71.58) than IndoNLU (63.59), and the difference was statistically significant ($p < 0.001$). In conclusion, IndoLEM is a more effective baseline model for Indonesian extractive QA systems. The findings also demonstrate that the composition and scale of pre-trained corpora substantially influence model performance in low-resource language contexts and highlight the importance of transfer learning for advancing NLP in underrepresented languages.

Corresponding Author:

Fattah Al Ilmi Suhendra, +6287784759707
Postgraduate Program, Management of Information Systems,
Universitas Gunadarma, Depok, Indonesia,
Email: fattahilmi@student.gunadarma.ac.id

How to Cite:
F. A. I. Suhendra, A. Darmayantie, A. S. Suhendra, and Pa Pa Min, ”Comparative Analysis of Indonesian Pre-trained BERT Models for the Extractive Question Answering Task on an Indonesian-Translated SQuAD Dataset”, *MATRIK: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer*, Vol. 25, No. 2, pp. 311-322, March, 2026.

1. INTRODUCTION

Recent advancements in natural language processing (NLP) have significantly enhanced the development of question answering (QA) systems, which aim to automatically extract or generate accurate answers to user queries from large textual datasets [1]. QA systems are increasingly deployed across domains such as education, healthcare, customer service, and knowledge management [2]. Despite these advancements, achieving seamless and accurate QA remains a challenge. Ambiguities inherent in language, multi-hop reasoning, domain-specific knowledge retrieval, and ethical considerations in generating reliable answers pose significant obstacles that researchers and practitioners strive to overcome [3]. The emergence of pre-trained models has revolutionized NLP research by providing powerful contextual representations learned from massive text corpora. These models leverage transfer learning to adapt general linguistic knowledge to a downstream task using relatively small, labelled datasets [4, 5].

Transformer-based language models such as Bidirectional Encoder Representations from Transformers (BERT) [6] and Generative Pre-Training (GPT) [7] have demonstrated significant progress in understanding contextual dependencies and semantic relationships in text. Several studies have developed QA systems using pre-trained BERT models. Lee et al. [8] proposed BioBERT as a domain-specific language model to enhance biochemical text-mining tasks. Their model enhances performance in biomedical NLP tasks, including named entity recognition (NER), relation extraction (RE), and QA. However, BioBERT's gains stem from domain adaptation in a high-resource setting, not from solving the fundamental challenges of cross-lingual transfer or resource scarcity—making it inapplicable to general-purpose Indonesian QA. Similarly, Zheng et al. [9] proposed PAL-BERT, which achieves efficiency gains by fine-tuning on the English SQuAD dataset. While effective in English, this approach assumes linguistic and structural compatibility between the source and target languages—an assumption that fails for Indonesian due to differences in morphology, word order, and tokenization. However, none of these studies have examined how differences in pretraining corpora affect the performance of Indonesian BERT models on extractive QA tasks.

Most transformer-based models have been primarily developed and optimized for high-resource languages such as English; however, their applicability and performance in low-resource languages remain comparatively underexplored. The original BERT base model was primarily trained in English, potentially overlooking linguistic structures and nuances specific to languages like Bahasa Indonesia. To address this limitation, several Indonesian BERT variants have been developed. Koto et al. introduced IndoBERT, trained on preprocessed Indonesian text comprising over 220 million words [10]. Another notable contribution, IndoNLU, presented by Wilie et al., established a comprehensive benchmark for Indonesian NLP tasks spanning twelve datasets of varying domains and complexities, built on top of the Indo4B pretraining corpus comprising approximately 23.4 GB of raw Indonesian text [11]. These contributions highlight active efforts to strengthen Indonesian NLP, yet systematic comparisons of Indonesian BERT variants, particularly regarding the influence of their distinct pretraining corpora on QA performance, remain limited.

Although several Indonesian BERT variants have been proposed, systematic empirical evaluations on extractive QA remain limited, particularly with respect to fine-tuning effectiveness, generalization behavior, and performance comparability. Prior studies have predominantly emphasized model development or dataset construction, leaving controlled benchmarking underexplored. To address this gap, this study empirically evaluates two widely used monolingual Indonesian BERT models—indobert-base-uncased (IndoLEM) and indobert-base-p1 (IndoBenchmark/IndoNLU)—on an extractive QA task using a standardized Indonesian translation of the SQuAD 2.0 dataset. Both models share identical architectural configurations but differ in the composition of their pretraining corpora, enabling a controlled analysis of how pretraining data characteristics influence QA performance. Model effectiveness is assessed using Exact Match (EM) and F1-score metrics, alongside an examination of training and generalization behavior. To the best of our knowledge, no prior study has conducted a controlled evaluation of monolingual Indonesian BERT models for extractive QA that jointly incorporates statistical significance testing and systematic error analysis.

This research, entitled "Comparative Analysis of Indonesian Pre-trained BERT Models for Extractive Question Answering Task on an Indonesian-Translated SQuAD Dataset", aims to provide empirical evidence on how differences in pretraining corpora shape model effectiveness in Indonesian QA. Both models were fine-tuned under identical hyperparameters and training configurations to ensure a fair comparison. Additionally, an extractive QA system was implemented to demonstrate the practical applicability of these models in real-world scenarios.

2. RESEARCH METHOD

The research process begins with problem identification, specifically determining a relevant and effective method or language model for completing the QA task. Developing a language model from scratch requires extensive resources, high computational costs, and a large dataset. To address these challenges, the author leverages a pre-trained model trained on large-scale data and fine-tunes it for the specific task. Data preparation encompasses both data collection and preprocessing, starting with gathering a labeled dataset suitable for the question-answering (QA) task in Figure 1.

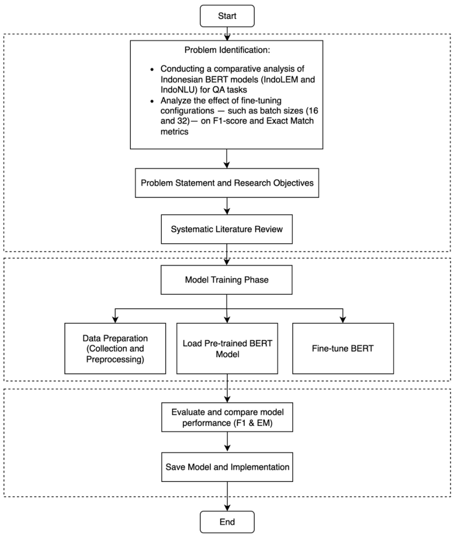

Figure 1. Research Methodology

The text data was then tokenized using the BERT Tokenizer. Furthermore, the pre-trained BERT model, along with its weights and architecture, is loaded. In this case, the Indonesian BERT base model, IndoBERT, is used. The model fine-tuning process was performed by adding task-specific layers on top of BERT, typically two linear layers: a start-position layer and an end-position layer, which predict the answer span's start and end positions. Then, this model was trained using the Trainer API and/or native PyTorch. After training, the fine-tuned BERT model was evaluated on a separate test or validation set to assess its performance and compute relevant evaluation metrics. The model's weights and architecture are then saved for future inference or deployment. The conceptual workflow pipeline of the research is illustrated in Figure 2.

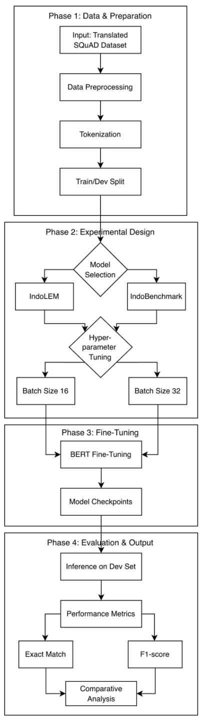

Figure 2. Workflow Pipeline

Based on the conceptual pipeline workflow, the research methodology follows a four-phase systematic approach. The process begins with Data Preparation, where the QA dataset is cleaned, tokenized, and split into training and development sets. The workflow then proceeds to the Experimental Design phase, in which two Indonesian BERT models—IndoLEM and IndoBenchmark—are selected and fine-tuned under controlled hyperparameter settings, specifically Batch Size 16 and Batch Size 32. Each model–configuration pair is subsequently processed through the Fine-Tuning Pipeline, generating corresponding model checkpoints. All models were fine-tuned using identical hyperparameters to ensure a fair comparative evaluation. Finally, in the Evaluation & Output phase, the performance of each checkpoint is measured using key QA metrics—F1 Score and EM—to enable a comprehensive comparative analysis and identify the optimal experimental configuration.

2.1. Data Preparation

The Indonesian-translated Stanford Question Answering Dataset (SQuAD) 2.0, provided by Muis et al. [12], was used as the primary dataset in this study. SQuAD has long been recognized as a benchmark for developing and evaluating question-answering systems due to its well-structured format, high-quality annotations, and inclusion of both answerable and unanswerable questions [13]. In their translation process, Muis et al. used the Google Translate API v2 to translate the English SQuAD data into Bahasa Indonesia, followed by a post-processing step that updated answer texts and their corresponding character positions. To resolve inconsistencies introduced by translation, they applied fuzzy string matching based on the Levenshtein edit distance, enabling the system to locate approximate matches whenever direct character alignment was not preserved.

The original Indonesian-translated dataset consists of 85,812 training samples and 8,170 validation samples, all of which are answerable question–answer pairs. However, additional preprocessing was conducted in this study to ensure that all answer start indices are correctly aligned with the first character of the annotated answer text. Since SQuAD stores answer annotations as the answer text and its character-based start location within the context, misalignments can lead to incorrect span extraction during fine-tuning. Therefore, entries with mismatched indices and text spans were removed. After this cleaning step, the dataset used in this study contained 85,776 training samples and 7,587 validation samples. Before tokenization, the data needs to be transformed into a format that is compatible with Hugging Face's Datasets library. This library simplifies the organization and management of multiple datasets by storing them in a dictionary-like structure, enabling efficient, streamlined processing.

2.2. Model

A pre-trained BERT architecture was employed, as illustrated in Figure 3, focusing on two Indonesian BERT base models trained on large-scale Bahasa Indonesia text corpora. The first model, indobenchmark/indobert-base-p1, has been downloaded 920,000 times and was trained on the Indo4B dataset, which consists of 4 billion words of pre-processed text data with a total file size of 23GB. The dataset was obtained from social media, online news, online articles, Wikipedia, parallel datasets, and video subtitles. This model contains approximately 124.5 million parameters. The second model, indolem/indobert-base-uncased, has been downloaded 7,960 times. It was trained on a corpus of over 220 million words, including 55 million from news sources such as Kompas, Tempo, and Liputan6, 74 million from Indonesian Wikipedia, and 90 million from the Indonesian web corpus. For QA tasks, an additional QA output layer was added on top of the BERT base model [6]. This layer consists of two vectors: a start-position vector that predicts the start position of the answer span within the input text, and an end-position vector that predicts the end position of the answer span. These vectors allow the model to precisely locate the portion of text that answers a question. The entire system, including the BERT base and the QA output layer, is then fine-tuned using a QA dataset to enhance its effectiveness for this task.

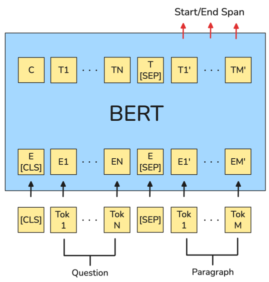

Figure 3. BERT architecture for Question Answering Tasks [6]

2.2.1. Hyperparameter Initialization

In this study, three key hyperparameters were configured to ensure stable, efficient fine-tuning of BERT-based models. First, the batch size, which determines the number of training samples used to compute the gradient at each optimization step, was set to 16 and 32, following the recommendation of Devlin et al. [6], who observed that batch sizes of 16 or 32 provide an effective balance between training stability, memory usage, and model performance. Second, the number of epochs, defined as the number of complete passes through the training dataset, was set to 4, aligning with common fine-tuning practices for BERT in QA tasks to prevent underfitting while avoiding overfitting. Lastly, the learning rate, a crucial parameter that controls the magnitude of weight updates during training, was set to 2×10^{-5}, a widely adopted value in BERT-based fine-tuning to promote stable convergence. Together, these hyperparameter choices aim to optimize model performance while maintaining training efficiency.

By default, the Trainer API uses AdamW, a sophisticated optimizer that blends elements of RMSprop and SGD with momentum. It adapts the learning rate for each parameter independently and updates weights based on estimates of the first (mean) and second (variance) moments of the gradients. AdamW is particularly effective for transformer models like BERT, as it manages sparse gradients well and incorporates weight decay during optimization.

For regularization, BERT incorporates dropout layers, typically with a rate of 0.1, to enhance robustness and generalization. This dropout rate ensures that a small proportion of neurons are dropped in each layer during training, preventing overfitting. Additionally, weight decay, commonly set to 0.01, penalizes large weights, further improving generalization and model stability.

2.2.2. Model Training

Trainer API was utilized to simplify the process of training and fine-tuning transformer models. It offers a high-level interface that streamlines tasks like data management, optimization, and evaluation. The API model is compatible with a wide range of models from the Hugging Face hub, enabling effortless fine-tuning of pre-trained models. Using the Trainer API ensures an efficient, organized training workflow, allowing researchers to focus on experimental design. Throughout training, the model iteratively updates its parameters to reduce loss, improving its accuracy in predicting correct answer spans within the provided context. An NVIDIA A100 GPU, a high-performance deep learning-optimized device, was used for model training. As shown in Figure 4 , the workflow included multiple steps to ensure efficient data handling and training.

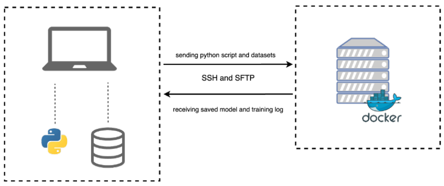

Figure 4. Fine-tuning BERT Model Environment

Training took place on a remote server equipped with the A100 GPU, accessed securely via SSH (Secure Shell). Data and model files, including datasets, pre-trained models, and scripts, were securely transferred via SFTP (Secure File Transfer Protocol). To maintain a consistent environment, training was carried out within a Docker container, ensuring consistent application and dependency management across sessions. Once the setup was complete, the training scripts were executed within the Docker container using the docker exec command, enabling smooth training and evaluation within the containerized environment. To ensure reproducibility, detailed specifications of the computational environment are provided. The specific software versions and hardware configurations used in this study are summarized in Table 11.

Table 1. Experimental Environment and Specifications

Component	Specification
Python Version	3.12.7
HF Transformers Version	4.45.0
PyTorch	2.4.0
CUDA	12.6.0
RAM	40 GB

2.2.3. Model Evaluation

Model performance was evaluated using quantitative metrics to ensure an objective assessment of prediction quality. The evaluation procedure adopts the standard metrics proposed in the SQuAD benchmark [13], as implemented in the Hugging Face evaluation library. Exact Match (EM), as defined in (1), measures the proportion of predictions that exactly match the ground truth answer, while the F1-score computes the harmonic mean of precision and recall at the token level. Both metrics were calculated following the original SQuAD evaluation formulation:

$$EM = \frac{\text{Number of exact matches}}{\text{Total number of questions}} \tag{1}$$

Exact Match refers to cases where the predicted answer exactly matches the ground truth answer span, while the total number of questions represents the number of QA pairs evaluated. The F1-score , defined in (2), measures the overlap between the predicted answer and the ground truth answer and is calculated as the harmonic mean of precision (3) and recall (4). Precision is defined as the number of correct positive results divided by the total number of positive results predicted by the model, whereas recall is the number of correct positive results divided by the total number of positive results that should have been returned. In QA tasks, precision and recall were adapted to assess how accurately the model predicts answer spans within the text.

$$Precision = \frac{\text{Number of overlapping tokens between the predicted and true span}}{\text{Total number of tokens in the predicted span}} \tag{2}$$

$$Recall = \frac{\text{Number of overlapping tokens between the predicted and true span}}{\text{Total number of tokens in the true span}} \tag{3}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

3. RESULT AND ANALYSIS

3.1. Training Results

During training, two versions of the BERT model, IndoLEM and IndoNLU, were fine-tuned using the Indonesian translation of the SQuAD dataset. Although both models share the same BERT-base architecture, they differ slightly in parameter size: IndoLEM (indobert-base-uncased) contains approximately 110 million parameters, while IndoNLU (indobert-base-p1) contains approximately 124.5 million parameters due to differences in vocabulary size and embedding configuration. To examine the impact of training stability, each model was fine-tuned with two batch sizes (16 and 32), enabling analysis of how mini-batch scaling influences convergence and final performance. The number of training epochs was set to 4, based on the recommendation from previously reported [6], which suggests that fine-tuning BERT for 2 to 4 epochs generally yields optimal performance across various downstream tasks. In this study, 4 epochs were selected to ensure sufficient learning without overfitting, while remaining within the recommended range. With a batch size of 16, the final epoch training losses were 0.5183 for IndoLEM and 0.6036 for IndoNLU, as illustrated in Figure 5. These values reflect the models' ability to minimize errors during training. For a more detailed analysis, the training loss was evaluated every 500 steps, yielding five measurements per epoch.

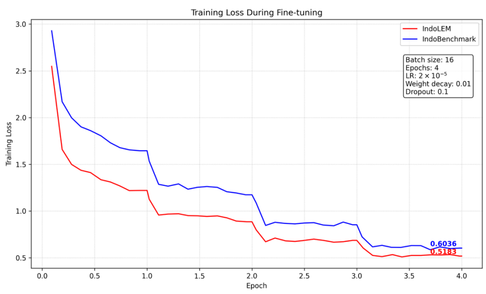

Figure 5. Training loss with a batch size of 16

The training loss value occurred when the batch size was set to 32, as depicted in Figure 6. In this setting, the final training losses for IndoLEM and IndoNLU models were 0.6686 and 0.7456, respectively. These values provide insights into the models' performance and convergence behavior at larger batch sizes, illustrating how their learning processes adapt across different batch configurations.

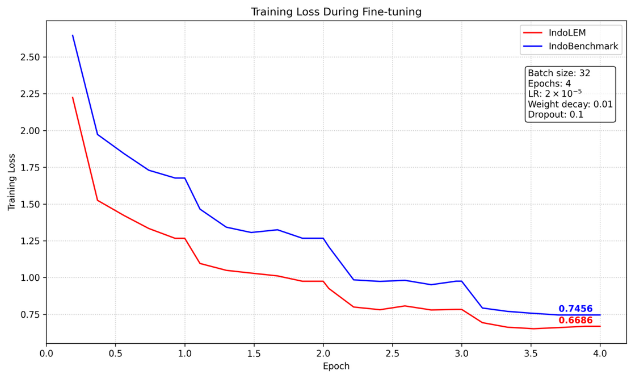

Figure 6. Training loss with a batch size of 32

Based on these results, there is no significant difference between the two batch size settings. The difference in the final training loss values between the two models suggests that each has adapted differently to the QA task. The IndoLEM model, with a slightly lower final training loss, provides a better fit to the training data than the IndoNLU model. Training losses were tracked and visualized to monitor the models' learning curves. The loss curves for both the IndoLEM and IndoNLU models showed a steady decrease in training loss over epochs, with batch size 16 converging faster than batch size 32. This visualization provided insights into how batch size affected training dynamics and model performance. In terms of execution and training time, with two batch size settings, both models have similar total execution time, ranging from 40 to 45 minutes.

3.2. Validation Results

The validation loss is calculated at the end of each epoch and remains relatively stable, providing insights into how well the models were generalizing to unseen data, as shown in Figure 7. At the final epoch (epoch 4), IndoLEM achieved a validation loss of 2.7150, while IndoBenchmark (IndoNLU) recorded a higher validation loss of 2.8193. The steeper slope of the IndoLEM curve suggests faster convergence and better optimization stability during training. This lower final loss aligns with IndoLEM's superior QA performance (F1 = 71.58 vs. 63.59), indicating stronger generalization to unseen data.When the batch size is set to 32, the validation loss is 2.3911 for the IndoLEM model and 2.7096 for the IndoNLU (IndoBenchmark) model, as illustrated in Figure 8.

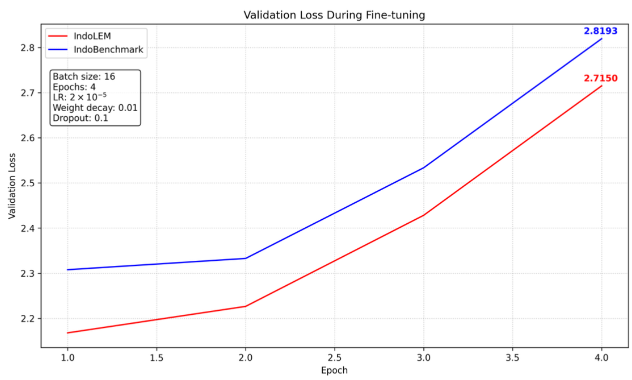

Figure 7. The validation loss with a batch size of 16

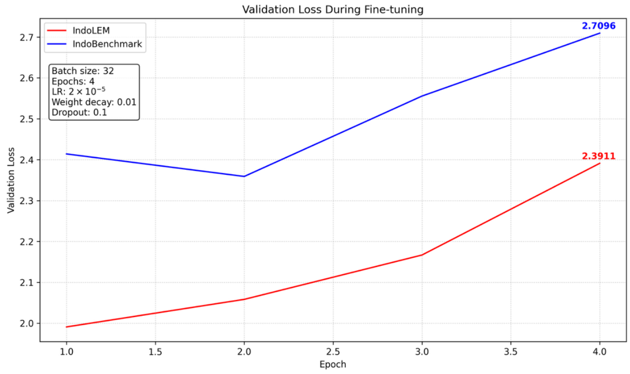

Figure 8. The validation loss with a batch size of 32

3.3. Model Evaluation

As shown in the evaluation results of both pre-trained models presented in Table 2, a specific set of hyperparameter was applied to perform a comparative analysis between the two Indonesian BERT models. The settings included an epoch count of 4 and batch sizes of 16 and 32, with an initial learning rate set to 2e-5. These specific hyperparameters were selected to ensure a fair and effective comparison of the models' performance.

Table 2. Evaluation Results

Component	Specification
Python Version	3.12.7
HF Transformers Version	4.45.0
PyTorch	2.4.0
CUDA	12.6.0
RAM	40 GB

The evaluation showed that IndoLEM delivered more dependable, accurate performance on Indonesian QA tasks. Its higher F1 Scores (71.58 and 71.48) and EM scores (60.58 and 60.11) highlight its superior ability to comprehend and process Indonesian, making it better suited for practical NLP applications in this domain. Furthermore, no notable differences in F1-score and EM were observed between batch sizes of 16 and 32. The superior performance of IndoLEM may be attributed to its cleaner, more domain-diverse corpus, which enhances the quality of contextual embeddings.

3.4. Statistical Significance

To assess whether the observed performance difference between IndoLEM and IndoNLU is statistically significant, bootstrap resampling was performed on the validation set with 10,000 iterations—a standard approach in QA evaluation when models are fine-tuned in a single run. This method estimates the sampling distribution of the F1 and Exact Match (EM) score differences by repeatedly resampling questions with replacement and recomputing metrics on each bootstrap sample. The results shown in Table 3 confirm that IndoLEM's advantage is highly significant. The mean F1-score difference of 7.10 points (71.58 vs. 63.59) has a 95% confidence interval of [6.32, 7.89] and a one-tailed p-value < 0.001, indicating that the probability of observing such a gap by chance is negligible. Similarly, the EM difference of 8.46 points (60.58 vs. 51.02) is also highly significant (95% CI: [7.53, 9.40], $p < 0.001$). These findings provide strong statistical evidence that IndoLEM's superior performance is not due to random variation but reflects a genuine improvement in answer extraction capability.

Table 3. Bootstrap Resampling Results

Metric	Difference	CI (95%)	P-value
F1-score	7.10	[6.32, 7.89]	< 0.001
Exact Match	8.46	[7.53, 9.40]	< 0.001

3.5. Error Analysis

To gain deeper insight into the performance gap between IndoLEM and IndoNLU, an error analysis was conducted on a controlled set of 15 questions derived from a single Indonesian passage about "Danau Toba", as shown in Table 4. Among these, 4 cases were identified in which IndoNLU failed to predict the correct answer while IndoLEM produced the correct result. The analysis was restricted to these instances to isolate model-specific weaknesses. All IndoNLU errors were found to be boundary mismatches: the predicted spans were either truncated or extended with extraneous words, resulting in inexact matches with the ground truth. These consistent boundary errors suggest that the tokenizer used in IndoNLU—shaped by a smaller and noisier pretraining corpus—struggled to align answer spans precisely with the linguistic structures of Bahasa Indonesia. This pattern confirms that pretraining corpus quality—particularly its impact on tokenization and phrase integrity—plays a decisive role in QA effectiveness for morphologically rich languages like Indonesian.

Table 4. Boundary Mismatch Errors in IndoNLU on the "Danau Toba" Passage

Question	IndoNLU Prediction	Ground Truth	Error Pattern
Mengapa Danau Toba dianggap penting bagi masyarakat lokal?	sumber mata pencaharian melalui perikanan dan pariwisata	sebagai sumber mata pencaharian melalui perikanan dan pariwisata	Missing leading word
Tantangan apa yang sedang dihadapi kawasan Danau Toba?	pencemaran air	pencemaran air akibat limbah domestik dan penurunan kualitas ekosistem perairan	Truncated answer
Di provinsi manakah Danau Toba berada?	Sumatera Utara, Indonesia	Sumatera Utara	Extra trailing words
Berapa tahun yang lalu Danau Toba terbentuk?	74.000 tahun yang lalu	74.000	Extra descriptive words

3.6. Save Model and Implementation

After fine-tuning the BERT model for a specific question-answering task, the resulting model was saved locally and uploaded to the Hugging Face Hub for easier deployment and accessibility. To showcase the capabilities of the fine-tuned model, a simple web-based application was developed using the FastAPI framework, chosen for its high performance and ease of API development. The application allows users to interact with the model by first providing a context paragraph related to their query, followed by a specific question based on that context. The system processes the inputs and returns a generated answer along with a confidence score that reflects the model's estimated reliability of the response. The implementation of the model is in Figure 9.

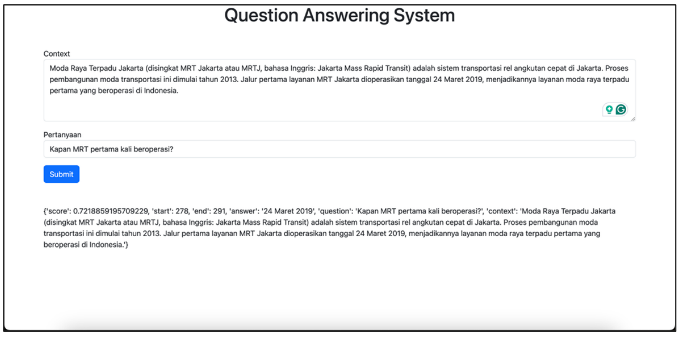

Figure 9. Model Implementation for Question Answering System

3.7. Discussions

The findings of this study advance current understanding of pretrained language model performance for Indonesian QA by offering a controlled and linguistically grounded comparison between two monolingual BERT variants. While previous work has evaluated architectures such as RoBERTa and IndoBERT-lite on translated SQuAD datasets, those comparisons often conflate architectural differences with variability in pretraining corpora and evaluation protocols. By holding the fine-tuning setup constant and comparing two full-sized Indonesian-specific models—IndoLEM and IndoNLU—this study isolates the effect of pretraining corpus design on downstream QA performance.

The performance gaps reported by Richardson and Wicaksana [14]—where RoBERTa outperforms IndoBERT-lite—reflects an inherently unbalanced comparison, as IndoBERT-lite is a distilled, efficiency-oriented model, while RoBERTa benefits from extensive multilingual pretraining. More notably, the higher scores reported by Suwarningsih et al. [15] (F1 = 78.29, EM = 67.89) stem from methodological differences: their approach involves heuristic post-processing of answers and evaluation on a filtered subset that excludes translation-induced misalignments. In contrast, the present study evaluates on the full, systematically cleaned Indonesian-SQuAD split (85,776 training; 7,587 validation), ensuring that results reflect true model capability. Under this stricter evaluation protocol, IndoLEM's F1-score of 71.58 represents a realistic upper bound for base-sized monolingual models on the benchmark.

While validation loss measures model confidence in predicting answer spans, the F1-score evaluates the precision and recall of the extracted text. It is possible for a model to achieve high F1 scores despite relatively high loss values—particularly when the model correctly identifies answer spans but assigns them moderate confidence. In this study, IndoLEM's lower validation loss (2.7150 at epoch 4) compared to IndoBenchmark (2.8193) aligns with its higher F1 score (71.58 vs. 63.59), indicating that IndoLEM not only predicts more accurate spans but also does so with greater confidence. This suggests better optimization and generalization during fine-tuning. Although indobert-base-p1 by IndoNLU is pretrained on the substantially larger Indo4B corpus (approximately 23.4 GB of raw text, containing several billion tokens), IndoLEM's indobert-base-uncased—trained on a smaller but cleaner 220M-word corpus—achieves higher F1 and EM scores in this study. This advantage is explained by corpus quality rather than corpus size: IndoLEM's pretraining data consists of well-curated Indonesian text (Wikipedia, news, and web sources) with minimal noise, yielding a more coherent subword vocabulary and better handling of Indonesian morphology. In contrast, Indo4B contains a broader range of user-generated, heterogeneous web content, introducing noise that affects tokenization consistency and reduces span extraction accuracy during QA fine-tuning.

These findings carry practical significance. For industry applications—such as customer service chatbots, legal information retrieval, and e-government QA systems—the approximately 8-point F1 improvement achieved by IndoLEM can meaningfully reduce incorrect answers and improve system reliability without requiring more complex architectures. From a research perspective, the results reinforce that, in morphologically rich and low-resource languages, the quality and linguistic suitability of the pretraining corpus exert greater influence on downstream performance than architectural choices alone. Consistent with observations from Ahmad and Romadhony [16] and TyDiQA [17]. The study further highlights the limitations of machine-translated benchmarks and underscores the need for native, human-authored Indonesian QA datasets to support rigorous evaluation and continued progress. Overall, this study demonstrates that IndoLEM's superiority is systematic and linguistically explainable: corpus quality drives tokenization fidelity, which in turn governs span-level QA accuracy. Nevertheless, these findings establish a general principle for low-resource NLP: native-language data curation is more impactful than architectural innovation when developing performant QA systems.

4. CONCLUSION

A language model was developed through the fine-tuning of two Indonesian variants of the pre-trained BERT architecture-indobert-base-p1 from IndoNLU/IndoBenchmark and indobert-base-uncased from IndoLEM – on an extractive QA task using a translated version of the SQuAD dataset. The experimental results showed that IndoLEM outperforms IndoBenchmark across multiple QA performance metrics. Specifically, IndoLEM achieved an F1-score of 71.58 and an EM score of 60.58, significantly surpassing IndoBenchmark's F1-score of 63.59 and EM score of 51.02. These scores highlight IndoLEM's stronger ability to accurately answer questions in Indonesian. Furthermore, IndoLEM consistently achieved lower training and validation losses than IndoBenchmark, indicating better convergence during fine-tuning and stronger generalization to new data. The reduced loss values suggest that IndoLEM is more effective at minimizing errors and enhancing QA task performance. Despite these findings, several limitations should be acknowledged. First, the study relies on a machine-translated Indonesian version of SQuAD, which may introduce translation noise, inconsistencies in answer alignment, and deviations from natural Indonesian linguistic structures. Second, only one QA dataset was used for benchmarking, limiting the generalizability of the comparative analysis across different question types, domains, or difficulty levels. Third, the analysis focused on standard EM and F1 metrics; more nuanced evaluations—such as robustness testing, domain adaptation, or error-type analysis—were not conducted. Lastly, due to computational constraints, advanced fine-tuning strategies (e.g., parameter-efficient training, multi-stage pretraining, or cross-lingual transfer techniques) were not explored.

Future research could expand comparisons to include native Indonesian QA datasets, providing more realistic insights into model capabilities. Collecting or curating such datasets—preferably across multiple domains such as education, legal, healthcare, or government—would be a significant step forward. Further improvements may be achieved by experimenting with ensemble approaches, data augmentation, adversarial training, or domain-specific pretraining. Investigating parameter-efficient methods (e.g., LoRA, adapters) may also improve model performance while reducing computational cost. Additionally, extending the evaluation to include robustness, zero-shot, or few-shot performance would yield a deeper understanding of model generalization. Overall, the findings of this study provide one of the few empirical benchmarks comparing Indonesian BERT models on extractive QA tasks, offering a valuable reference point for future model development. The results highlight the importance of pretraining corpus quality in low-resource languages, demonstrating that models trained on larger and more diverse Indonesian corpora can deliver substantial performance gains. This work also emphasizes the need for high-quality, native Indonesian QA datasets, which remain scarce but are essential to advancing Indonesian NLP. More broadly, the study contributes to ongoing efforts to improve NLP technologies for underrepresented and linguistically diverse languages, reinforcing the potential for transformer-based architectures to support inclusive AI development.

5. ACKNOWLEDGEMENTS

I would like to extend my deepest gratitude to Universitas Gunadarma for its generous financial support, made possible by the invaluable resources and opportunities it has provided. I am deeply grateful to my advisors and colleagues for their guidance and support, whose insights have significantly contributed to the development of this work.

6. DECLARATIONS

AI USAGE STATEMENT

During the preparation of this work, the authors used ChatGPT (OpenAI) and Gemini to improve the language and clarity of the manuscript. After using this tool, the authors reviewed and edited the content as needed and took full responsibility for the publication's content.

AUTHOR CONTIBUTION

All authors contributed significantly to the writing of this article. They were actively involved in drafting and developing the content presented in the manuscript. In addition, all authors participated in revising and refining the article to ensure its clarity and accuracy.

FUNDING STATEMENT

This project is financially supported by Universitas Gunadarma. The funding provided by the university played an important role in enabling the completion of this research. The authors gratefully acknowledge the financial assistance and institutional support received for this project.

COMPETING INTEREST

The authors confirm that there are no conflicts of interest, financial or non-financial, that could influence the research results or the interpretation of the data in this article.

REFERENCES

[1] M. Zaib, W. E. Zhang, Q. Z. Sheng, A. Mahmood, and Y. Zhang, "Conversational question answering: A survey," *Knowledge and Information Systems*, vol. 64, no. 12, pp. 3151–3195, Dec. 2022, https://doi.org/10.1007/s10115-022-01744-y.

[2] Y. Chen, "Intelligent question answering system for internet of things data analysis and educational technology," in *Proceedings of the 2021 International conference on Smart Technologies and Systems for Internet of Things (STS-IOT 2021)*. Atlantis Press, 2022, pp. 274–279, https://doi.org/10.2991/ahis.k.220601.052.

[3] S. Cai, Q. Ma, Y. Hou, and G. Zeng, "Knowledge graph multi-hop question answering based on dependent syntactic semantic augmented graph networks," *Electronics*, vol. 13, p. 1436, 4 2024, https://doi.org/10.3390/electronics13081436.

[4] X. Luo, Z. Deng, B. Yang, and M. Y. Luo, "Pre-trained language models in medicine: A survey," *Artificial Intelligence in Medicine*, vol. 154, p. 102904, Aug. 2024, https://doi.org/10.1016/j.artmed.2024.102904.

[5] H. Wang, J. Li, H. Wu, E. Hovy, and Y. Sun, "Pre-Trained Language Models and Their Applications," *Engineering*, vol. 25, pp. 51–65, jun 2023, https://doi.org/10.1016/j.eng.2022.04.024.

[6] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang, "Pre-trained models for natural language processing: A survey," *Science China Technological Sciences*, vol. 63, pp. 1872–1897, 10 2020, https://doi.org/10.1007/s11431-020-1647-3.

[7] H. W. Chung, L. Hou, S. Longpre, B. Zoph, T. Yi, W. Fedus, Y. Li, X. Wang, M. Dehghani, and S. Brahma, "Scaling instruction-finetuned language models," *Journal of Machine Learning Research*, vol. 25, pp. 1–53, 2024.

[8] H. Wang, J. Li, H. Wu, E. Hovy, and Y. Sun, "Pre-trained language models and their applications," *Engineering*, vol. 25, pp. 51–65, 6 2023, https://doi.org/10.1016/j.eng.2022.04.024.

[9] W. Zheng, S. Lu, Z. Cai, R. Wang, L. Wang, and L. Yin, "PAL-BERT: An Improved Question Answering Model," *Computer Modeling in Engineering & Sciences*, vol. 139, no. 3, pp. 2729–2745, 2024, https://doi.org/10.32604/cmes.2023.046692.

[10] F. Koto, A. Rahimi, J. H. Lau, and T. Baldwin, "Indolem and indobert: A benchmark dataset and pre-trained language model for indonesian nlp," in *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, 2020, pp. 757–770, https://doi.org/10.18653/v1/2020.coling-main.66.

[11] B. Wilie, K. Vincentio, G. Winata, X. Li, Z. Lim, S. Soleman, R. Mahendra, P. Fung, S. Bahar, and A. Purwarianti, "Indonlu: Benchmark and resources for evaluating indonesian natural language understanding," in *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th*, 3 2020, pp. 843–857, https://doi.org/10.48550/arXiv.2009.05387.

[12] F. J. Muis and A. Purwarianti, "Sequence-to-sequence learning for indonesian automatic question generator," in *2020 7th International Conference on Advance Informatics: Concepts, Theory and Applications (ICAICTA)*. IEEE, 9 2020, pp. 1–6, https://doi.org/10.1109/ICAICTA49861.2020.9429032.

[13] S. Moon, H. He, H. Jia, H. Liu, and J. W. Fan, "Extractive clinical question-answering with multianswer and multifocus questions: Data set development and evaluation study," *JMIR AI*, vol. 2, p. e41818, 6 2023, https://doi.org/10.2196/41818.

[14] B. Richardson and A. Wicaksana, "Comparison of indobert-lite and roberta in text mining for indonesian language question answering application," vol. 18, no. 06, pp. 1719–1734, July, 2022, https://doi.org/10.24507/ijicic.18.06.1719.

[15] W. Suwarningsih, R. A. Pramata, F. Y. Rahadika, and M. H. A. Purnomo, "RoBERTa: Language modelling in building Indonesian question-answering systems," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 20, no. 6, p. 1248, dec 2022, https://doi.org/10.12928/telkomnika.v20i6.24248.

[16] G. N. Ahmad and A. Romadhony, "End-to-end question answering system for indonesian documents using tf-idf and indobert," in *2023 10th International Conference on Advanced Informatics: Concept, Theory and Application (ICAICTA)*. IEEE, 10 2023, pp. 1–6, https://doi.org/10.1109/ICAICTA59291.2023.10390111.

[17] J. H. Clark, E. Choi, M. Collins, D. Garrette, T. Kwiatkowski, V. Nikolaev, and J. Palomaki, "T y di qa:a benchmark for information-seeking question answering in ty pologically di verse languages," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 454–470, Dec. 2020, https://doi.org/10.1162/tacl_a_00317.