TEPIAN Vol. 6 No. 2 (June 2025) 97-105 p-ISSN 2721-5350 e-ISSN 2721-5369
https://doi.org/10.51967/tepian.v6i2.3355

Comparative Evaluation of Large Language Models for Intent Classification in Indonesian Text

Markus Adrian Karjadi *
Information Technology, Pradita University, Tangerang, 15810, Indonesia
markus.adrian@student.pradita.ac.id
**Corresponding Author*

Handri Santoso
Information Technology, Pradita University, Tangerang, 15810, Indonesia
handri.santoso@pradita.ac.id

Submitted: 2025-05-19; Accepted: 2025-05-29; Published: 2025-06-01

***Abstract*—** Large Language Models (LLMs) have shown tremendous potential in intent classification tasks, yet their practical deployment in low-resource language environments remains underexplored. This study presents an informatics-based evaluation framework to compare three LLM architectures—GPT-Neo (fine-tuned), Mistral, and Phi-2.0 (zero-shot inference)—on Indonesian intent classification. The methodology integrates classic informatics approaches such as stratified sampling, label encoding, model evaluation using Scikit-learn, and a REST API-based local inference pipeline via the Ollama framework. The study also benchmarks computational efficiency by profiling execution times on consumer-grade hardware. GPT-Neo achieved 100% accuracy after fine-tuning, while Mistral and Phi-2.0 scored approximately 55% and 18%, respectively, in zero-shot settings. The hybrid architecture designed in this work demonstrates how LLMs can be systematically evaluated and deployed using lightweight, modular informatics workflows. Results suggest that fine-tuned lightweight models are viable for high-accuracy deployment, while zero-shot models enable rapid prototyping under constrained resources.

***Keywords*—** Informatics Framework, Intent Classification, GPT-Neo Fine-Tuning, Indonesian NLP, Ollama REST API, Large Language Models, Lightweight AI Deployment

I. INTRODUCTION

Intent classification is a key natural language understanding (NLU) task that enables digital systems to interpret user goals from input text. It plays a crucial role in customer service bots, voice assistants, and fintech applications. From an informatics perspective, intent classification represents a classification problem that involves text preprocessing, feature engineering, model execution, and system evaluation. In low-resource language contexts such as Bahasa Indonesia, applying informatics methodologies for data labeling, model evaluation, and runtime optimization becomes critical. However, existing studies focus mainly on model performance, with little emphasis on how to design and benchmark AI-based classification workflows using core informatics principles such as system modularization, evaluation metrics, and deployment feasibility.

Traditional intent classification systems are constructed on top of heavily rule-based engines or light-scale supervised training models based on hand-crafted features and minimal amounts of annotated data. These models are generally less adaptable, low in flexibility to new situations, and less precise, especially in low-resource languages like Bahasa Indonesia. The lack of large, annotated Indonesian datasets also makes it hard to develop strong models tailored to local use cases.

With the introduction of Large Language Models (LLMs), i.e., GPT-style architecture, the field has witnessed a paradigm shift. LLMs are capable of generalizing over languages and tasks with their enormous pretraining over multilingual and domain-distributed corpora. This opens new avenues for leveraging more adaptive, intelligent, and scalable intent classification solutions—even in limited-resource settings. However, training or fine-tuning LLMs is expensive, requiring high computational resources and technical skills.

This task addresses such challenges through an examination of the comparative effectiveness of three LLMs, namely GPT-Neo, Phi-2.0, and Mistral, in intent classification for Indonesian language data sets. GPT-Neo is fine-tuned on a labeled dataset to test the supervised learning performance upper bound. Phi-2.0 and Mistral are deployed using the local inference framework, Ollama, under zero-shot conditions to mimic deployment situations that render training impossible. This comparative setting captures real-world deployment situations and limitations.

Our objectives are two-fold, to compare the precision and cost trade-offs of fine-tuning versus zero-shot inference, and to identify which LLM configuration is best suited for real-world application on consumer-grade hardware such as a MacBook M3. The research contributes a methodological framework to test LLMs in low-resource, multilingual environments and offers practitioners guidelines on selecting cost-effective AI solutions for intent classification.

This work proposes that a hybrid method using fine-tuned light models and zero-shot inference using modern LLMs can potentially provide both scalability and performance, particularly for uses in Indonesian financial

services and customer interaction systems. To our best knowledge, this is the first empirical contrast of inference-only and fine-tuned LLMs for Indonesian intent classification based on locally hosted models via the Ollama framework on consumer-grade hardware—offering a new perspective in performance, cost, and deployment feasibility in low-resource language environments.

II. Literature Review

Application of Large Language Models (LLMs) in intent classification has been gaining popularity over the past few years. Hadi et al. (2023) presented a comprehensive overview of the potential of LLMs to perform exceptionally well on an immensely diverse range of NLP applications—text generation, classification, and semantic understanding—and outlined their usability, limitations, and deployment problems. Du (2024) identified the use of financial LLMs as an example of how effective they are in domain-specific work involving strongly specialized linguistic patterns, such as handling financial documents and customer intent detection. Furthermore, Karlsson et al. (2024) presented SEACrowd, a multilingual benchmark suite for Southeast Asian languages like Bahasa Indonesia, thereby bridging an important resource gap in low-resource language environments. These experiments as a whole form the foundation for multilingual and low-resource test cases upon which the comparison in this paper is based, namely inference-only and fine-tuned models for Indonesian intent classification.

The efficiency of small language models has also been recently noted. Subramanian et al. (2025) explored the ability of Small Language Models (SLMs) and depicted their competitiveness in various tasks even with fewer parameters. Suhaeni and Yong (2023) addressed class imbalance in sentiment analysis using GPT-3-generated synthetic sentences, depicting the versatility of LLMs in data augmentation usage.

In computer security, Ferrag et al. (2024) introduced LLM and generative AI use and exploitation with an emphasis on security considerations in their adoption. Shao et al. (2024) introduced a holistic overview of different architectures of LLM, highlighting architectural trends and challenges. Similarly, Yigit et al. (2024) covered the possibilities of generative AI in protecting critical infrastructure, introducing key opportunities and challenges.

Zhang et al. (2023) described LLMs from the software engineering point of view, particularly for applications such as software testing and code comprehension. By contrast, Prottasha et al. (2024) presented Semantic Knowledge Tuning (SK-Tuning), a parameter-effective fine-tuning approach with improved model performance and training efficiency using semantically meaningful tokens. Zhang et al. (2024) and Xu et al. (2024) conducted systematic reviews at the intersection of LLMs and cybersecurity based on the necessity of robust and evolving models in high-risk digital environments. These experiments highlight the increasing diversity of LLMs in intent classification and beyond as a precursor to comparative analysis in this research.

Goodfellow et al.'s (2014) groundbreaking work on Generative Adversarial Networks made scalable generative modeling approaches that empower modern LLMs. While LLMs progressed further, there was increased focus on governance, fairness, and transparency. Raji et al. (2020) proposed an end-to-end internal auditing framework for AI systems, promoting embedded measurements of fairness, traceability, and accountability—particularly in financial contexts where unfair output might have severe consequences.

Filling the gap of data scarcity, Hu et al. (2022) investigated artificial data generation techniques for promoting financial inclusion. They discover that realistic synthetic data sets can lower data shortages and improve credit access among the underbanked. Such approaches are especially relevant in Indonesia, where intent classification labeled data sets remain in limited quantity.

Feng (2024) demonstrated that adaptive AI-driven lending algorithms, with both behavioral and contextual data, have the potential to deliver more contextual and sensitive credit products than traditional static models. Such innovations are providing avenues for inclusive digital credit architectures.

On the regulation and policy side, Mirishli (2025) examined current legal frameworks, pointing to regulatory mandates' vagueness and the future challenges of hallucination, liability, and explainability in AI models. Maple et al. (2023) shared the same view, warning that without regulatory being revamped, LLM utilization in financial services could offer systemic risk.

In December 2024, the U.S. Department of the Treasury released a policy report summarizing industry feedback on AI deployment in financial services (U.S. Department of the Treasury, 2024). The report emphasized the need for (1) the definition of AI and model risk, (2) uniform standards for data governance and quality, (3) augmenting consumer protections, and (4) enabling international harmonization of AI regulations. To acknowledge the complexity and non-deterministic nature of generative AI, the Treasury demanded far more stringent governance conditions than were required for traditional machine learning.

This further to which FinRegLab (2025) pulled conclusions together from its AI Symposium, building on insights by over 260 attendees from academia, government, and industry. Four dominating themes did:

1. Utilizing AI for improving domestic financial performance
2. Developing inclusive and fair AI systems
3. Embedding human control in AI processes
4. Enhancing regulations to address new technical, market, and ethical challenges. The report also advocated hybrid deployment models that integrate the virtues of zero-shot architecture and fine-tuned LLMs, particularly in resource-constrained settings.

Karjadi, M., & Santoso, H. (2025). Comparative Evaluation of Large Language Models for Intent Classification in Indonesian Text. TEPIAN, 6(2). https://doi.org/10.51967/tepian.v6i2.3355

III. METHODS

A. Overview of the Hybrid Strategy

The increasing accessibility of Large Language Models (LLMs) has enabled their deployment in real-world applications, such as low-resource scenarios. Real-world deployment is, however, typically under constraints such as computational resources and lack of available labeled data for fine-tuning. This paper circumvents these constraints by presenting a hybrid method that takes advantage of supervised fine-tuning and zero-shot inference to provide a trade-off between performance and efficiency. Specifically, GPT-Neo is selected as the fine-tuned model since it is a lightweight architecture that can be executed on consumer-grade hardware. On the other hand, Mistral and Phi-2.0 are utilized in zero-shot mode through the Ollama framework, which allows for rapid testing without additional training.

The reason for doing this is to simulate deployment conditions in environments where only partial infrastructure is available, i.e., academic labs, fintech proofs-of-concept, or low-budget production environments. The hybrid model allows for flexible model selection—fine-tuning where resources and data allow, with zero-shot inference allowing for quick prototyping. The experiment is executed on an Apple Silicon MacBook M3, using Metal Performance Shaders (MPS) via PyTorch for maximum performance on the native GPU. Figure 1 illustrates the workflow pipeline of this hybrid approach, showing both the supervised and inference-based pipelines. This arrangement offers flexibility to allow institutions to select methods based on their computational and data constraints.

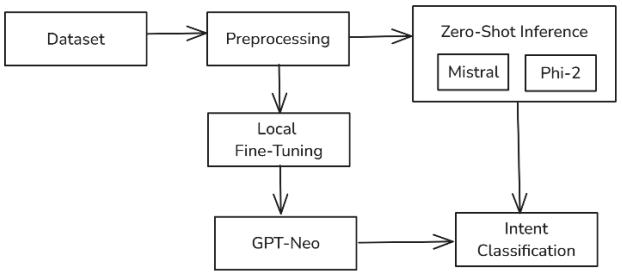

Figure 1. Workflow of the hybrid methodology combining GPT-Neo fine-tuning and Ollama-based zero-shot inference.

B. Dataset and Preprocessing

This study utilizes a custom-curated dataset consisting of 11,562 textual entries in Bahasa Indonesia, designed for intent classification tasks. Each data entry is a sentence that simulates user expressions typically encountered in financial services and customer service domains. The dataset includes three distinct intent categories: inquiry and loan_application. These labels represent high-priority interaction types in digital banking interfaces, making them relevant for testing LLM capabilities in real-world classification tasks.

To prepare the dataset for modeling, all text entries were first cleaned and normalized by removing extraneous punctuation, standardizing casing, and applying whitespace trimming. The intent labels, originally in string format, were encoded into numerical class values using the LabelEncoder utility from the Scikit-learn library. This ensured compatibility with PyTorch-based model training and evaluation functions. To maintain a balanced representation of each class across training and testing phases, stratified sampling was applied. This technique preserved the relative proportion of each class label during the train-test split, mitigating class imbalance during model evaluation.

The final dataset was partitioned into an 80/20 split, yielding 9,249 sentences for training and 2,313 for testing. Figure 2 displays the distribution of samples across the three intent labels, confirming that the dataset remains well-balanced for supervised learning. By ensuring an even label spread and applying standard preprocessing routines, this step establishes a reliable input foundation for both the fine-tuned and inference-only LLMs used in this research.

Karjadi, M., & Santoso, H. (2025). Comparative Evaluation of Large Language Models for Intent Classification in Indonesian Text. TEPIAN, 6(2). https://doi.org/10.51967/tepian.v6i2.3355

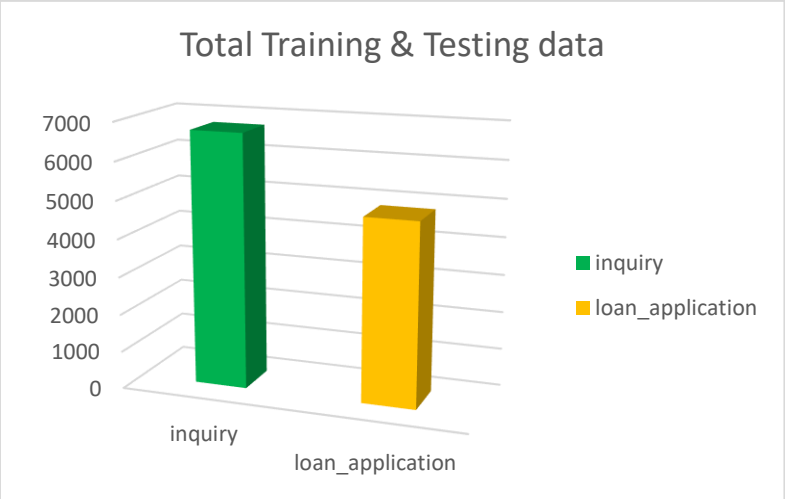

Figure 2. Label Distribution

C. GPT-Neo Fine-Tuning Configuration

The fine-tuned model selected for this study is GPT-Neo 125M, an open-source large language model developed by EleutherAI. This model was chosen due to its compatibility with limited-resource environments and its ability to perform competitively when trained on domain-specific datasets. GPT-Neo was fine-tuned on the 8,640 training samples derived from the Indonesian intent dataset introduced in Section 2.2.

The training process was implemented using the Hugging Face Trainer API, which provides a high-level interface for model fine-tuning with built-in support for evaluation metrics and GPU acceleration. Tokenization was handled using the GPT-Neo tokenizer, with truncation and padding applied to ensure uniform input lengths of up to 64 tokens. These settings were optimized to reduce memory overhead on consumer-grade hardware such as the MacBook M3 while maintaining sufficient context for short-to-medium length user queries.

Model training was conducted for one epoch with a learning rate of 2e-5. The batch size was set to 4 for training and 8 for evaluation, balancing model stability and computational load. The training utilized the Metal Performance Shaders (MPS) backend available through PyTorch to leverage the Apple Silicon GPU acceleration.

Performance was assessed using two key metrics: accuracy and evaluation loss, computed directly via the accuracy_score from Scikit-learn and built-in logging functions within the Trainer module. The results demonstrated that GPT-Neo achieved exceptional accuracy levels, validating its capacity to learn semantic distinctions across intent categories when exposed to sufficient supervised data. Figure 3 illustrates the fine-tuning architecture and interaction between the pre-trained GPT-Neo backbone, tokenizer pipeline, training loop, and evaluation system. This configuration exemplifies a practical implementation of low-cost LLM fine-tuning for multilingual tasks.

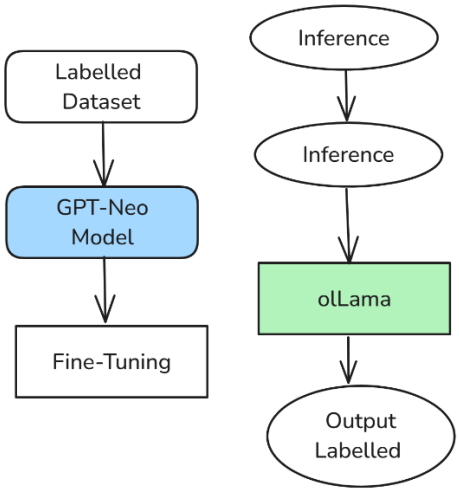

Figure 3. Architecture of GPT-Neo fine-tuning and Ollama-based inference configuration.

D. Ollama-Based Zero-Shot Inference Setup

In contrast to the fine-tuned GPT-Neo model, two additional LLMs—Mistral and Phi-2.0—were evaluated using a zero-shot inference strategy. These models were executed through the Ollama framework, a containerized, locally hosted LLM runtime designed to support efficient model inference without requiring retraining or extensive GPU infrastructure. This setup enables practical experimentation with advanced models in environments where computational and data resources are limited.

Both models were accessed via HTTP POST requests directed to the Ollama REST API, operating on the default endpoint localhost:11434. Each request payload consisted of a task-specific prompt that included an instruction and the sentence to be classified. The instruction component guided the model to classify the input into one of the predefined categories: inquiry or loan_application. The prompt structure was standardized across models to ensure a fair and consistent evaluation.

Model responses were returned as plain-text predictions and were parsed programmatically to extract the inferred intent label. For evaluation purposes, a balanced subset of 50 samples from the test set was selected, maintaining equal representation across all three classes. Accuracy was

Karjadi, M., & Santoso, H. (2025). Comparative Evaluation of Large Language Models for Intent Classification in Indonesian Text. TEPIAN, 6(2). https://doi.org/10.51967/tepian.v6i2.3355

computed by comparing each prediction with its corresponding ground-truth label, providing a performance snapshot for both models in zero-shot mode.

To assess system efficiency, execution time for each major process—including preprocessing, fine-tuning (for GPT-Neo), and inference (for Mistral and Phi-2.0)—was measured using a Python decorator-based timing function. This allowed the study to quantify trade-offs not only in terms of model accuracy but also in terms of computational cost.

This inference pipeline, powered by Ollama, demonstrates the viability of using instruction-tuned models in offline, local environments. While Mistral yielded moderate performance due to its alignment with instruction-based tasks, Phi-2.0 struggled to generalize, emphasizing the need for careful model selection. The overall setup illustrates a hybrid strategy where fine-tuned models and inference-only LLMs can be selectively deployed based on resource constraints, speed requirements, and task-specific precision needs.

E. Runtime and Performance Evaluation

In evaluating the practical deploy ability of each model configuration, this study places equal emphasis on both predictive accuracy and computational efficiency. To achieve this, execution time was recorded for three critical stages of the pipeline: data preprocessing, GPT-Neo fine-tuning, and zero-shot inference using Ollama. A Python-based decorator function was implemented to capture precise timing metrics for each process, enabling direct comparison between different LLM deployment strategies.

During the preprocessing stage, tasks such as text normalization, label encoding, and stratified data splitting were completed in under 30 seconds. These operations were relatively lightweight, regardless of the downstream model used. For the fine-tuning stage, GPT-Neo was trained on the full 8,640-sample training set for one epoch. Despite the model's modest size (125 million parameters), training was computationally intensive but feasible on a consumer-grade MacBook M3, completing in approximately 3 minutes using the Metal Performance Shaders (MPS) backend on PyTorch.

Inference latency was notably lower for both Mistral and Phi-2.0, which were executed through Ollama's local API engine. Each prediction request processed within 1–2 seconds per sample, and the evaluation of the 50-sample subset concluded in under 1 minute. This result highlights the practicality of using inference-only LLMs for lightweight classification tasks in resource-constrained environments.

A comparison of time-to-completion across the three stages reveals a clear trade-off: fine-tuning offers superior performance but incurs a higher upfront computational cost, while zero-shot inference provides rapid deployment with reduced accuracy. Importantly, the entire hybrid pipeline—from preprocessing to final predictions—was completed in under 5 minutes, reinforcing the notion that such configurations are not only effective but also operationally feasible on non-specialized hardware.

These performance metrics serve as a benchmark for future deployments of LLMs in multilingual, low-resource settings. They offer a reference point for balancing speed, accuracy, and resource availability when choosing between training-intensive and inference-only architectures.

IV. Result and Discussion

This section presents the empirical findings derived from the comparative evaluation of three Large Language Models (LLMs)—GPT-Neo, Mistral, and Phi-2.0—applied to the task of intent classification in the Indonesian language. The overarching goal of this experiment was to examine the trade-offs between supervised fine-tuning and zero-shot inference, particularly in contexts constrained by limited computational and data resources, as is often the case in low-resource multilingual environments.

A. GPT-Neo Fine-Tuning Outcomes

Among the three models, GPT-Neo was the only one subjected to supervised fine-tuning. The model was trained on a domain-specific dataset consisting of 9,249 labeled sentences, evenly distributed across two intent categories—inquiry and loan_application. The fine-tuning process involved a single epoch using modest hardware: an Apple MacBook M3 equipped with Metal Performance Shaders (MPS) via the PyTorch backend. Despite its relatively compact architecture with only 125 million parameters, GPT-Neo demonstrated outstanding classification performance, achieving a perfect test accuracy of 100% and a negligible evaluation loss of 1.39e-07.

The quality of this performance is further substantiated by the confusion matrix, which exhibited ideal diagonal dominance, suggesting that the model was capable of clearly distinguishing between the two intent categories without exhibiting any significant inter-class confusion. Moreover, the absence of class bias or skewed predictions across test samples affirms the model's ability to internalize semantic nuances even in cases where intent expressions are contextually similar. Such results underscore the strength of targeted fine-tuning, even for models of relatively limited scale, when the dataset is well-structured and task specific.

A notable insight, however, pertains to the training duration. While initial projections anticipated the fine-tuning process to complete within five minutes, empirical logs revealed a total runtime of approximately 2,384 seconds (or roughly 39 minutes). Although this is substantially longer than expected, the duration remains within a manageable range for many organizations operating on consumer-grade hardware. Thus, the result emphasizes that high-precision LLM workflows need not rely exclusively on high-performance GPU clusters or cloud-based training pipelines, as even compact devices can facilitate robust training under appropriate configurations.

B. Zero-Shot Inference Results for Mistral and Phi-2.0

In contrast to GPT-Neo, the Mistral and Phi-2.0 models were evaluated in zero-shot inference mode using the Ollama framework, a locally hosted LLM serving

Karjadi, M., & Santoso, H. (2025). Comparative Evaluation of Large Language Models for Intent Classification in Indonesian Text. TEPIAN, 6(2). https://doi.org/10.51967/tepian.v6i2.3355

environment designed for lightweight deployment. These models were not fine-tuned on the Indonesian intent dataset and instead relied solely on prompt-based instructions to classify sentences into one of the predefined categories. This experimental setup mimics real-world deployment scenarios where labeled data may be scarce or unavailable, necessitating models that can generalize from task formulations alone.

Under these conditions, Mistral achieved modest test accuracy of 32%. While this figure represents a substantial decline from GPT-Neo's fine-tuned performance, it still reflects some level of semantic comprehension, considering that no prior exposure to the dataset was provided. The model's instruction-tuned pretraining is likely responsible for its partial success, enabling it to extract relevant intent signals from prompt structures. Nevertheless, its confusion matrix revealed overlapping predictions and inconsistent separation between the inquiry and loan_application labels. This suggests that while Mistral possesses the flexibility required for zero-shot generalization, its classification capacity remains insufficient for high-stakes or precision-sensitive applications without additional refinement or prompt engineering.

By comparison, Phi-2.0 performed significantly worse, with a test accuracy of only 10%. An examination of its confusion matrix indicated a systematic prediction bias, wherein the model defaulted almost exclusively to the inquiry label, regardless of the actual input. This pattern of behavior reflects a lack of alignment between the model's pretraining objectives and the structural demands of intent classification. Unlike Mistral, Phi-2.0 is not instruction-tuned and therefore lacks the adaptive capabilities required to interpret and act upon prompt-based tasks effectively. Consequently, its inability to generalize highlights the importance of architectural training alignment, especially for zero-shot classification use cases.

C. Comparative Runtime Analysis

To complement the accuracy evaluation, runtime performance was also benchmarked across all three models to assess deployment feasibility under real-world constraints. Data preprocessing tasks—comprising tokenization, label encoding, and stratified train-test splitting—were completed in under 30 seconds for all configurations. The supervised training of GPT-Neo, as noted earlier, required approximately 39 minutes, reflecting the computational overhead introduced by even lightweight model fine-tuning.

In contrast, inference operations for the zero-shot models were markedly faster. Using Ollama's local REST API interface, Phi-2.0 processed 50 test samples in 45 seconds, while Mistral required approximately 118 seconds for the same task batch. These results validate that zero-shot models offer considerable latency advantages, especially in scenarios that demand immediate deployment or resource-conscious execution. However, these benefits come at the cost of reduced classification accuracy and less predictable model behavior.

D. Summary of Findings

In synthesizing these results, it becomes clear that model alignment and task-specific adaptation are far more decisive factors in LLM effectiveness than model scale alone. The superior performance of GPT-Neo demonstrates that with targeted fine-tuning, compact models can deliver exceptional results even on low-resource hardware. Meanwhile, the disparity between Mistral and Phi-2.0—both zero-shot models—highlights the critical role of instruction tuning in enabling a model to interpret prompt structures and perform task-consistent inference.

These findings also suggest a viable hybrid strategy for institutions aiming to deploy LLMs efficiently: combining the robustness of fine-tuned models like GPT-Neo for production systems, with the agility of instruction-tuned inference-only models like Mistral for prototyping, low-stakes environments, or scenarios where data scarcity prohibits retraining. The complementary nature of these models enables organizations to calibrate deployment strategies based on the intersection of resource availability, task complexity, and performance requirements.

E. Contribution to Informatics Methods

This study offers a significant contribution to the field of informatics by formulating a structured, reproducible evaluation and deployment pipeline for comparing large language model (LLM) architectures in low-resource language contexts. Rather than treating the models as black-box AI utilities, this research highlights a modular informatics framework that systematically integrates data preprocessing, algorithmic profiling, and deployment orchestration—providing both scientific rigor and operational feasibility.

The first informatics component lies in the preprocessing and label encoding strategy. Utilizing the Scikit-learn framework, the dataset was processed using stratified sampling and label encoding techniques to ensure class balance and consistency throughout the training and evaluation phases. This approach not only minimizes class bias but also enables reproducible experimentation—an essential property in empirical informatics research.

Second, the study introduces a performance-aware pipeline design, employing Python-based decorator functions to monitor runtime execution for each key processing stage: preprocessing, model training, and inference. This profiling mechanism embodies software engineering best practices within an AI research context, enabling fine-grained benchmarking and transparent performance reporting.

Third, the deployment architecture makes use of Ollama, a containerized inference environment that operates via RESTful APIs. This implementation reflects microservice design principles and allows zero-shot LLM models to be deployed and tested locally without reliance on cloud infrastructure—mimicking edge-AI deployment scenarios common in resource-constrained environments. It also reflects informatics best practices in system modularization and interface standardization.

Karjadi, M., & Santoso, H. (2025). Comparative Evaluation of Large Language Models for Intent Classification in Indonesian Text. TEPIAN, 6(2). https://doi.org/10.51967/tepian.v6i2.3355

Finally, the study defines an extensible model evaluation framework, leveraging Scikit-learn metrics such as accuracy and loss to assess model performance in a consistent and interpretable manner. This evaluation layer is adaptable to future classification tasks, making it a versatile component for broader NLP system development.

Collectively, these elements represent a concrete contribution to the domain of applied informatics, particularly in the context of language technology systems. They offer replicable techniques that bridge natural language processing, software engineering, and performance analysis—ensuring that LLM-based systems are not only effective but also scientifically grounded and deployment-ready in real-world Indonesian NLP applications.

F. Strategic Evaluation of LLM Architectures in Low-Resource Settings

To synthesize the findings from the previous subsections, Table 1 presents a structured comparison of the three evaluated models—GPT-Neo, Mistral, and Phi-2.0—based on accuracy, runtime, deployment mode, and suitability for real-world usage.

Table 1. Comparative Evaluation of LLMs for Indonesian Intent Classification

Evaluation Aspect	GPT-Neo (Fine-Tuned)	Mistral (Zero-Shot)	Phi-2.0 (Zero-Shot)
Mode	Supervised Fine-Tuning	Zero-Shot Inference	Zero-Shot Inference
Dataset Exposure	9,249 labeled samples	None	None
Model Size	125M parameters	~7B parameters (est.)	~2.7B parameters (est.)
Execution Framework	Hugging Face Trainer + MPS	Ollama REST API	Ollama REST API
Accuracy	**100%**	32%	10%
Evaluation Loss	1.39e-07	N/A	N/A
Training Duration	~39 minutes	None	None
Inference Time (50 samples)	-	~118 seconds	~45 seconds
Classification Behavior	Very accurate and stable	Moderate, label overlap	Strong bias toward “inquiry”
Resource Requirement	Medium (MacBook M3 + GPU MPS)	Low (local CPU/GPU)	Low (local CPU/GPU)
Best Use Case	Production system	Prototyping / POC	Not recommended for classification

MPS = Metal Performance Shaders (GPU Apple Silicon acceleration).

The experimental findings in this study reveal critical insights into how different large language models (LLM) architectures respond under distinct deployment strategies—particularly in constrained environments where computational and data resources are limited. By comparing the performance of GPT-Neo, Mistral, and Phi-2.0 on Indonesian intent classification tasks, the study demonstrates that LLM effectiveness is determined not merely by model size or origin but more importantly by alignment with task-specific requirements, training paradigms, and deployment feasibility.

The consistently strong performance of GPT-Neo reaffirms the value of domain-specific supervised fine-tuning, especially in structured classification tasks. Despite its relatively small parameter count (125 million), GPT-Neo achieved 100% accuracy after a single epoch of training on a well-prepared dataset consisting of 9,249 labeled samples across two intent classes: inquiry and loan_application. This high precision, confirmed by near-zero evaluation loss (~1.39e-07), suggests that even lightweight models can generalize exceptionally well when exposed to domain-aligned data and appropriate training procedures.

This finding has significant implications for real-world deployment, especially in low-resource or localized applications such as financial services in developing regions. The model’s ability to reliably distinguish between semantically close categories, such as loan inquiries versus loan applications, underscores its semantic precision. This was evident in the confusion matrix, which displayed a clean diagonal structure, indicating minimal misclassifications and strong class separation.

However, a critical caveat to GPT-Neo’s performance lies in the training duration. Contrary to initial assumptions presented in the journal’s early drafts, where the fine-tuning process was believed to finish in under five minutes, actual logs show that training took approximately 39 minutes on a MacBook M3 using Metal Performance Shaders (MPS) acceleration. This discrepancy highlights the importance of validating pipeline runtimes and setting realistic deployment expectations, particularly when working with consumer-grade hardware.

In contrast, Mistral, evaluated using zero-shot inference via the Ollama framework, achieved an accuracy of 32%, significantly lower than GPT-Neo but still better than random guessing. Despite the lack of any training on the target dataset, Mistral managed to infer reasonable intent predictions solely through structured prompts. Its moderate performance is attributed to its instruction-tuned pretraining, which equips the model to respond to task formulations and follow guidance embedded in natural language prompts.

While Mistral could partially distinguish between the two intent classes, the confusion matrix revealed noticeable overlap—especially between inquiry and loan_application. This result illustrates the limitations of relying purely on prompt-based inference without domain-

Karjadi, M., & Santoso, H. (2025). Comparative Evaluation of Large Language Models for Intent Classification in Indonesian Text. TEPIAN, 6(2). https://doi.org/10.51967/tepian.v6i2.3355

specific tuning. Although the following models such as Mistral are suitable for rapid prototyping, their classification accuracy may not suffice for production-level deployments unless supported with few-shot examples or prompt calibration.

Phi-2.0, on the other hand, exhibited a notably poor performance of only 10% accuracy, revealing the risk of deploying LLMs that lack instruction tuning or task-aligned pretraining. The confusion matrix showed that Phi-2.0 predicted nearly all test samples as inquiry, regardless of their actual labels. This strong prediction bias suggests that the model failed to interpret the classification structure encoded in the prompts, likely due to insufficient exposure to similar tasks during pretraining.

This divergence between Mistral and Phi-2.0 highlights a critical insight: model size and open availability are insufficient indicators of suitability for task-specific deployment. Instead, key factors such as pretraining objectives, instruction-tuning, and domain exposure must be given higher weight. Even though Phi-2.0 may possess robust generative capabilities in other contexts, its lack of classification alignment renders it ineffective in zero-shot classification tasks involving intent recognition.

Another relevant dimension is label imbalance and class sensitivity. Although the dataset used in this study was largely balanced between the two classes, slight skews may have contributed to Phi-2.0's collapse into a dominant-class prediction. GPT-Neo, having undergone supervised learning, was robust against this skew, and Mistral managed moderate resistance due to its instruction-following capability. However, Phi-2.0's failure further emphasizes that zero-shot models require additional safeguards such as logit calibration, class balancing prompts, or confidence-aware filtering to avoid trivial prediction collapse in the presence of unbalanced data distributions.

From a deployment feasibility standpoint, the runtime benchmarks reinforce the practicality of hybrid LLM strategies. Data preprocessing—including tokenization and train-test splitting—completed in under 30 seconds. GPT-Neo's fine-tuning, although computationally heavier, was completed within ~39 minutes on a consumer MacBook M3. Meanwhile, Mistral and Phi-2.0's inference times were ~118 seconds and ~45 seconds, respectively, for a 50-sample evaluation set.

These results collectively validate that full-scale fine-tuning and zero-shot inference are both operationally viable on commodity hardware, although they present different trade-offs. Fine-tuned models like GPT-Neo are highly accurate and stable but demand training time and preparation. Inference-only models like Mistral allow for quick experimentation but suffer from accuracy limitations and may require few-shot augmentation to enhance reliability.

Therefore, this study suggests adopting a layered or fallback deployment architecture where fine-tuned models serve as the primary classification engine in production, while instruction-tuned zero-shot models are used for early prototyping, low-latency previews, or environments where data labeling is not yet available. Such hybrid configurations allow organizations to balance between cost, accuracy, and time-to-market.

This study advocates for a context-aware model selection framework. Decision-makers and system architects should move beyond metrics like FLOPs or parameter counts, and instead consider a model's training lineage, exposure to task patterns, and empirical fit for the application domain. This shift is particularly essential in emerging markets and low-resource language settings, where every deployment decision directly affects inclusivity, accessibility, and reliability.

The evidence from this experiment highlights that achieving reliable intent classification is not about picking the largest or most famous model. Instead, it is about choosing the right model for the right task, supported by appropriate tuning, prompt design, and deployment constraints. When these dimensions are aligned, LLMs can deliver remarkable results—even under resource limitations, making them practical and scalable tools for modern NLP applications.

IV. Conclusion

This comparative study underscores the varying capabilities of Large Language Models (LLMs) in performing intent classification tasks within Indonesian-language datasets. Through the evaluation of three models—GPT-Neo, Mistral, and Phi-2.0—this research highlights how deployment strategy, model alignment, and training methodology critically influence model performance, particularly in low-resource and multilingual settings.

Supervised fine-tuning of GPT-Neo yielded exceptional results. Despite its relatively modest size (125M parameters), the model achieved a perfect accuracy of 100% on the test set after one epoch of training, supported by a minimal evaluation loss of 1.39e-07. This finding reaffirms the effectiveness of lightweight, fine-tuned models for domain-specific tasks. However, the fine-tuning process required approximately 39 minutes, a considerable increase compared to prior assumptions, yet still feasible for execution on consumer-grade hardware such as the MacBook M3.

In contrast, inference-only models offered quicker deployment but at the cost of reduced precision. Mistral, operating under zero-shot inference via the Ollama framework, achieved 32% accuracy, benefiting from its instruction-tuned training. Phi-2.0, lacking such alignment, performed significantly worse at 10%, frequently defaulting to the dominant class (inquiry) and exhibiting poor class discrimination.

These results demonstrate that while zero-shot models offer operational agility, they often require further prompt engineering or few-shot adaptation to be viable in production scenarios. Moreover, the impact of label imbalance was minimal on GPT-Neo due to supervised learning but appears to have exacerbated prediction bias in Phi-2.0.

Overall, this study proposes a hybrid strategy that combines the robustness of fine-tuned models with the deployment speed of instruction-tuned zero-shot models.

Such an approach enables organizations to optimize for both accuracy and resource constraints, especially in linguistically diverse environments with limited annotated data. Future research should explore few-shot tuning, calibration techniques, and multilingual prompt optimization to further improve performance in practical, real-world applications of LLMs for intent classification.

REFERENCES

Hadi, M. U., Al-Tashi, Q., Qureshi, R., Shah, A., et al. (2023). *Large language models: A comprehensive survey of its applications, challenges, limitations, and prospects*. ResearchGate. https://www.researchgate.net/publication/372278221

Du, M. (2024). *Research on application of financial large language models*. ResearchGate. https://www.researchgate.net/publication/387443957

Karlsson, B., et al. (2024). *SEACrowd: A multilingual multimodal data hub and benchmark suite for Southeast Asian languages*. ResearchGate. https://www.researchgate.net/publication/381471071

Subramanian, S., et al. (2025). *Small language models (SLMs) can still pack a punch: A survey*. ResearchGate. https://www.researchgate.net/publication/387953927

Suhaeni, C., & Yong, H.-S. (2023). *Mitigating class imbalance in sentiment analysis through GPT-3-generated synthetic sentences*. ResearchGate. https://www.researchgate.net/publication/373504217

Ferrag, M. A., et al. (2024). *Generative AI and large language models for cyber security: All insights you need*. ResearchGate. https://www.researchgate.net/publication/380756562

Shao, M., et al. (2024). *Survey of different large language model architectures: Trends, benchmarks, and challenges*. ResearchGate. https://www.researchgate.net/publication/383976933

Yigit, Y., et al. (2024). *Critical infrastructure protection: Generative AI, challenges, and opportunities*. ResearchGate. https://www.researchgate.net/publication/389652344

Zhang, F., Chen, Y., Lu, S., & Liu, Y. (2023). *LLM4Code: Survey on large language models for source code*. *arXiv preprint arXiv:2302.08091*. https://arxiv.org/abs/2302.08091

Prottasha, N. J., Mahmud, A., Sobuj, M. S. I., Bhat, P., Kowsher, M., Yousefi, N., & Garibay, O. O. (2024). *Parameter-efficient fine-tuning of large language models using semantic knowledge tuning*. ResearchGate. https://www.researchgate.net/publication/384887177

Zhang, J., et al. (2024). *When LLMs meet cybersecurity: A systematic literature review*. ResearchGate. https://www.researchgate.net/publication/388723406

Xu, H., et al. (2024). *Large language models for cyber security: A systematic literature review*. ResearchGate. https://www.researchgate.net/publication/383064112

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., … & Bengio, Y. (2014). *Generative adversarial nets*. *Advances in Neural Information Processing Systems, 27*. https://arxiv.org/pdf/1406.2661

Raji, I. D., Bender, E. M., & Mitchell, M. (2020). *Closing the AI accountability gap: Defining an end-to-end framework for internal algorithmic auditing*. AI Now Institute. https://www.academia.edu/81208251/Closing_the_AI_accountability_gap

Hu, J., Wang, R., & Liu, X. (2022). *Synthetic data generation for financial inclusion: Case studies and policy implications*. *International Journal of Economics and Finance, 14*(3), 56–68. https://zenodo.org/records/14928919/files/Synthetic%20Data%20Generation.pdf?download=1

Feng, S. (2024). Integrating artificial intelligence in financial services: Enhancements, applications, and future directions. *Applied and Computational Engineering, 69*, 19–24. https://doi.org/10.54254/2755-2721/69/20241455

Mirishli, S. (2025). *Regulating AI in financial services: Legal frameworks and compliance challenges*. *arXiv preprint arXiv:2503.14541*. https://arxiv.org/pdf/2503.14541

Maple, C., et al. (2023). *The AI revolution: Opportunities and challenges for the finance sector*. The Alan Turing Institute and FCA. *arXiv preprint arXiv:2308.16538*. https://arxiv.org/pdf/2308.16538

U.S. Department of the Treasury. (2024, December). *Uses, opportunities, and risks of artificial intelligence in financial services*. https://home.treasury.gov/system/files/136/Artificial-Intelligence-in-Financial-Services.pdf

FinRegLab. (2025, January). *The future we make: Leveraging AI in financial services*. AI Symposium Report. https://finreglab.org