Jurnal Ilmu Komputer dan Informasi (Journal of Computer Science and Information)
18/2 (2025), 155-165. DOI: http://dx.doi.org/10.21609/jiki.v18i2.1446

Classification of Economic Activities in Indonesia Using IndoBERT Language Model

Muhammad Rizki Syazal[1], Evi Yulianti[2]

Faculty of Computer Science, University of Indonesia, Depok, Indonesia

Email: [1]muhammad.rizki215@ui.ac.id, [2]evi.y@cs.ui.ac.id

Abstract

Classification of economic activities plays a vital role in understanding, analyzing, and managing complex economic processes in a society or country. It facilitates economic analysis, data collection, policy formulation, and informed decision-making. In Indonesia, economic activities are classified according to the Indonesian Standard Industrial Classification (KBLI). This classification process requires in-depth knowledge about KBLI, and this process is still performed manually, which is therefore time-consuming. To address this challenge, this paper proposes to use a transformer-based language model that was pretrained using a large Indonesian corpus, i.e., IndoBERT, to better understand the contextual meanings of text in order to improve the accuracy of automatic economic activity classification. Our results show that the finetuned $\text{IndoBERT}_{\text{LARGE}}$ model achieves superior results, with an F_1 score of 96.82% and a balanced accuracy of 96.10%, outperforming other recent methods used for similar task, i.e., CatBoost and DistilBERT models.

Keywords: *Multiclass Classification, IndoBERT, DistilBERT, CatBoost, Activity Economy*

1. Introduction

In statistics, standardization is a crucial aspect. Standards ensure that data on the same characteristic is collected and communicated uniformly every time. This allows for consistent comparison of data from different sources. To address this issue, Statistics Indonesia (BPS) has developed standardized classifications of data, e.g., KBLI (*Klasifikasi Baku Lapangan Usaha Indonesia* or translated as Indonesian Standard Industrial Classification). KBLI is a coherent and consistent classification structure for economic activities in Indonesia based on a set of nationally agreed concepts, definitions, principles, and classification rules. Currently, KBLI is utlized to classify data according to the type of economic activity in the fields of economic and social statistics, such as to obtain the statistics of national accounts, demography of enterprises, employment, and others. In addition, KBLI is also employed for business licensing, company registration, and investment licensing.

The KBLI is organized hierarchically into five levels: Category, Main Class, Class, Subclass, and Group. The lower levels provide more specific category information compared to the upper levels. Each level has its own code format. The Category level is coded by a one-digit letter (Sections A to U). The Main Class, Class, Subclass and Group levels are respectively coded by 2-digit, 3-digit, 4-digit, and 5-digit numbers. The structure details and examples of codes in each level are presented in Table 1.

Determining the 5-digit KBLI code can be a challenging and time-consuming task due to the numerous codes and manual classification process. In the economy census conducted by Statistics Indonesia, a supervisor of enumerator usually determines manually the best suitable 5-digit KBLI code for the given description of enterprise's main activity and the product produced. Using a method called "top-down approach", a supervisor will trace the hierarchy of categories to find the 5-digit code in the lowest level of the hierarchy, i.e., group level. He begins by identifying a relevant category that contributes the most to the value added at the highest level and continues down to the lower levels until reaching the lowest level with a 5-digit code. For example, when classifying *"Five-Star Hotels"* ac-

Table 1. KBLI 2015 edition's structure and examples

Level	Number of Categories	Code Format	Example	
			Code	Description
Category	21	1-digit, Alphabet	I	Accomodation and Food Service Activities
Main Class	88	2-digit, Number	55	Accomodation
Class	240	3-digit, Number	551	Short Term Accomodation Activities
Subclass	520	4-digit, Number	5511	Star Hotel
Group	1573	5-digit, Number	55111	Five-Star Hotels

cording to the KBLI 2015, the classification process starts with the broad category *"I"*, which refers to *"Accommodation and Food Service Activities"*. This is followed by the main class *"55"*, which indicates *"Accommodation"*. Further refinement leads to class *"551"*, which denotes *"Short Term Accommodation Activities"*, and then to sub-class *"5511"*, which is specific to *"Star Hotels"*. Finally, the most precise group '55111' identifies*"Five-Star Hotels"*, illustrating how the classification system moves from general to specific, culminating in the 5-digit code *"55111"*, which accurately encapsulates this particular economic activity. Furthermore, the effectiveness of this process is highly dependent on the expertise of the supervisor, as not all supervisors possess the same level of knowledge. Consequently, even with identical descriptions, the speed and accuracy of classification may vary for each supervisor. For instance, in the animal feed industry, the KBLI code is 10802 for processed animal feed manufacturing, while it is 47754 for selling animal feed in stores. Despite the similarity of the physical product, the nature of the operation is different; one is manufacturing, and the other is trading.

Considered as a multiclass classification problem, classifiying an enterprises's economic activity into a 5-digit KBLI code can be automated by applying machine learning approach. It will lead to more reliable and less subjective results while significantly enhance the efficiency and accuracy of the classification process. Several studies [1–5] have used machine learning approach in addresing multiclass classifcation problem across various domains. Tradisional machine learning models such as Support Vector Machines (SVM) [6], Naïve Bayes [7], K-Nearest Neighbor [8] and Decision Trees [9] have been extensively employed in addressing classification problems. However, their performance accuracy depends heavily on the quality and relevance of the data features extracted. Since 2010s, multiclass classiffication techniques gradually changed from traditional model to deep neural network model. These deep learning models enable the automatic extraction of semantically meaningful representations from text, as demonstrated by methods like Word2Vec [10], and Glove [11].

Currently, there are several pre-trained language models that can be used to produce contextualized word embedding, such as ELMo and BERT [12]. These models have proven to be more effective as input representation than context-independent word embeddings like Word2Vec and Glove [13]. Bidirectional Encoder Representations from Transformers (BERT) is a contextual language model that has been reported to achieve superior performance in many NLP tasks [12]. BERT architecture consists of multilayer bidirectional Transformer encoders that learns the context of a language in a bidirectional way. In addition to producing contextualized word embeddings, BERT can also be fine-tuned for specific downstream tasks, such as text classification [14] and question answering [15]. These advancements have inspired researchers to create models that are specifically designed to address the intricacies of different languages and cultural contexts. One notable example is IndoBERT [16], a state-of-the-art contextual language model based on the BERT model that was pre-trained using a large size of Indonesian-language dataset (Indo4B) collected from various sources such as social media texts, blogs, news, and websites. Experimental results demonstrate that IndoBERT outperforms a cross-lingual pre-trained language model.

In previous research on classifying economic activities in Indonesia, various methods have been employed, ranging from traditional machine learning approaches to more recent advancements in language models. Aldania et al. [17] compared the Double Random Forest (DRF) and Catboost machine learning methods using the Economic Census 2016 data, and the results showed that Catboost outperformed DRF with a balanced accuracy of 92.45%. On the other hand, Dwicahyo and Yuniarto [18] employed the Gated Recurrent Unit (GRU) method with word embedding fastText using a similar dataset. However, the accuracy of their model was shown to be very low, only gaining a 48.24% success rate. These studies, though valuable, haven't explored the potential of Transformer-based language models, which have demonstrated exceptional performance

in classification task. A recent work by Bechara et al. [19] has also explored the use of some pretrained language models to classify economic activities into the International Standard Industrial Classification (ISIC) codes and showed that the DistilBERT model achieved the best performance with an accuracy of 77.9%. Despite they use a transformer-based language model, the dataset used is only in English. Consequently, the Distilbert model, which has been pre-trained exclusively on English data, may not be as effective in accurately classifying Indonesian economic activities.

Our work is different to previous work in which we use transformer-based language model, IndoBERT, which has been exclusively pretrained on Indonesian dataset. We employ IndoBERT to classify economic activities in Indonesia into 5-digit KBLI codes on the Indonesian Economic Census 2015 dataset, as used in [17]. We hypothesize that IndoBERT model can better capture the semantics of Indonesian text by looking at both left and right context, therefore it can help to distinguish the characteristics of text representing economic activities between classes in the group level of KBLI hierarchy. In our approach, IndoBERT is utilized through a fine-tuning process, where it serves as end-to-end model to directly solve our downstream task of classifying the economic activities in Indonesia. To the best of our knowledge, no prior research has explored the use of IndoBERT for this task.

The rest of this paper is organized as follows: In Section 2, we cover the theory behind our proposed method. In Section 3, we provide details about the dataset, our method, how we evaluate it, and the experiment process. Section 4 discusses our results and the implications, while Section 5 concludes the paper by summarizing our main points and highlighting the limitations encountered during the study.

2. Proposed Method

This section provides an overview of the theoretical background of Bidirectional Encoder Representations from Transformers (BERT), which serves as the foundational model for the proposed approach. It also discusses DistilBERT and IndoBERT, which are variants of BERT.

BERT, short for Bidirectional Encoder Representations from Transformers, is a pre-trained language model introduced by Devlin et al. [12]. It composed a multi-layered Transformer [20] encoder and was trained on a large amount of text data, BooksCorpus (800M words) and English Wikipedia (2,500M words). The training process of BERT is divided into two main phases: pre-training and fine-tuning, as illustrated in Figure 1. During the pre-training phase, two primary tasks are employed: Masked Language Model (MLM) and Next Sentence Prediction (NSP). MLM enhances bidirectional learning from text through word masking in a sentence, where words on both the left and right sides of the masked word are used to predict the masked word. This process helps the model to understand the context from both directions, making it excel at handling ambiguous words and sentence structures. NSP is used to help the model understand the relationship between two sentences. Given two sentences A and B, BERT will predict whether B is the actual next sentence that comes after A in the corpus or just a random sentence. After pre-training, the BERT model can be fine-tuned for specific downstream tasks, such as text classification, named entity recognition, and question answering. When it was proposed, it achieved state-of-the-art performance on eleven NLP tasks.

Although BERT has shown remarkable performance in diverse NLP tasks, its implementation is limited on devices with limited resources due to its large size and high computational demand [21]. To address this problem, several studis have utilized knowledge distillation, a model compression technique in which a smaller model (student) is trained to match or even exceed the performance of a larger and more powerful model (teacher). Sanh et al. [22] leverage this method by introducing a smaller pre-trained model known as DistilBERT. BERT was used as the teacher model, resulting in a student model with 40% fewer parameters and 60% faster in execution speed, while maintaining 97% of BERT's performance on various downstream tasks.

Moreover, the fact that BERT was trained exclusively on an English corpus motivated researchers to develop BERT models for other languages. One such model is proposed by Willie et al [16]. They introduced a pre-trained language model called IndoBERT, along with its smaller variant, IndoBERT-lite, which is based on ALBERT [23]. IndoBERT and IndoBERT-lite were trained on a vast Indonesian corpus (4,000M words). The corpus was collected from various sources, including online news, social media, Wikipedia, online articles, and video subtitles. From the experiment result, $\text{IndoBERT}_{\text{LARGE}}$ achieved the top-3 best performance results in both classification task and sequence labeling task [16].

In this study, we employe three pre-trained BERT based models, $\text{IndoBERT}_{\text{BASE}}$, $\text{IndoBERT}_{\text{LARGE}}$, and DistilBERT. We fine-tuned these pre-trained models on a Indonesian corpus to perform the classification of economic activities in Indonesia into 5-digit KBLI codes. Details of the

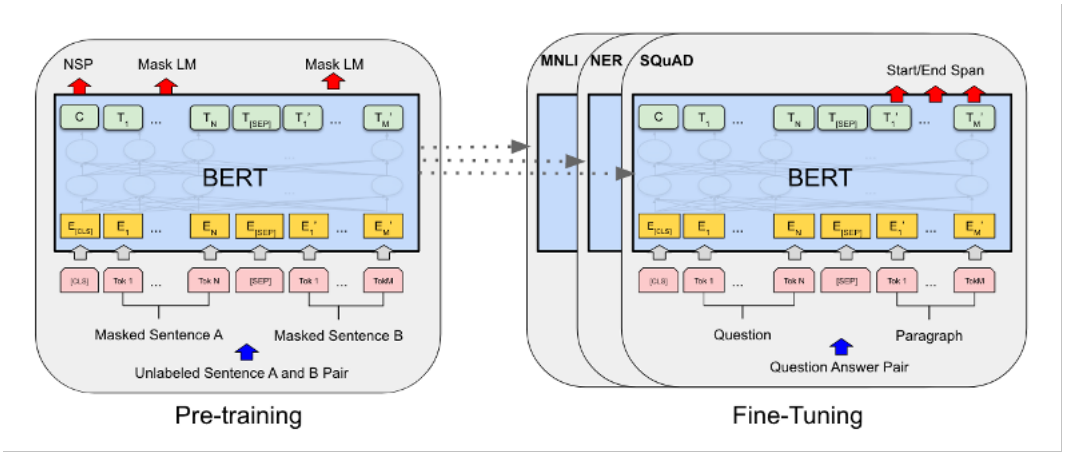

Figure 1. Pre-traning and fine-tuning procedure for BERT [12].

models are presented in Table 2.

Table 2. The details of pre-trained model

Model	Transformer Layers	Attention heads	Hidden Size	Parameters
$\text{IndoBERT}_{\text{BASE}}$	12	12	766	124.5M
$\text{IndoBERT}_{\text{LARGE}}$	24	16	1024	335.2M
DistilBERT	6	12	768	66M

3. Methodology

3.1. Dataset

The dataset utilized in this study was obtained from the 2016 economic census listing of enterprises in Indonesia. It is an identical dataset employed by Aldania et al. [17]. The research employed three variables: the description of a company's primary activity (x_1), the primary product or service produced by the enterprise (x_2), and the 5-digit KBLI code (y). For better understanding of the dataset, we provide a snapshot of the dataset in Table 3.

Before training the model, several processes were undertaken, including preprocessing data, data exploration, and feature extraction. Preprocessing involves creating a new variable x as a description that represent company's primary activity and product. This is achived by concatenating x_1 and x_2, converting it to lowercase, removing the stopword, and stemming. The stemming process was carried out using the algorithm proposed by Andriani et al. [24], implemented via the PySastrawi python library. Subsequently, the variable y is encoded as codes 0 to 25.

Table 3. Snapshot of the dataset

No	Main Activity (x_1)	Main Product (x_2)	KBLI (y)
1	*Katering* (Catering)	*Makanan* (Food)	56210
2	*Hotel melati* (Low-priced hotel)	*Penyediaan kamar* (Room accomodation)	55120
3	*Menjual jamu keliling bersepeda motor minuman jamu tradisional* (selling herb by motorcycle)	*minuman jamu tradisional* (traditional herbal drinks)	56306
4	*Jual makanan siap saji di bangunan tetap* (selling ready-to-eat meals in a fixed building)	*Nasi, sayur dan lauk pauk* (rice, vegetables and side dishes)	56102
5	*Penyediaan akomodasi dan penyediaan makan minum* (Accommodation and food service activities)	*Jasa boga untuk suatu event tertentu* (event catering)	56210

Following preprocessing, an exploratory analysis of the dataset was performed to gain insights into its characteristics. Figure 2 depicts the distribution of the 5-digit KBLI codes within the dataset. As illustrated, there is a notable imbalance among the KBLI codes. To illustrate, code '56102' dominates the dataset with 391 samples, while code '55909' has only 5 samples. Furthermore, over half of the 26 classes have fewer than 110 samples. This highly skewed distribution may cause the prediction model to be biased, as the model more often predicts the input as belonging to the majority class, resulting in low performance in the minority class.

In addition, we also analyzed the distribution of the number of words in the combined description text (x). As illustrated in Figure 3, most of the description texts consist of between 4 and 6

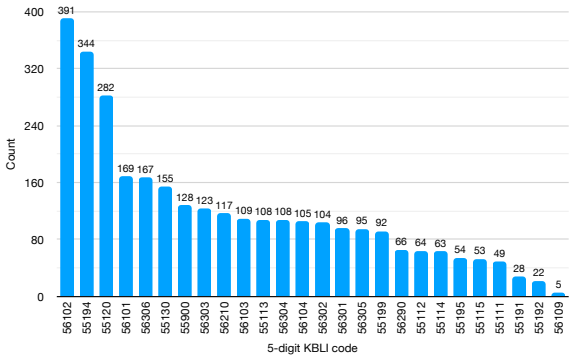

Figure 2. Class distribution of 5-digit KBLI code

words, with the minimum number of words being 2 words, while the maximum number reaches 14 words. Furthermore, we observed that some data had the same KBLI code but with different word counts. For instance, the combined descriptions '*katering makanan*' and '*penyediaan akomodasi dan penyediaan makan minum jasa boga untuk suatu event tertentu*' (Table 3, entries 1 and 5) are classified into the same 5-digit KBLI code, '56210', but have significantly different word counts. This variation highlights a challenge of extracting meaningful representations for text classification tasks. Short descriptions may lack sufficient context, while longer descriptions can introduce noise or redundancy.

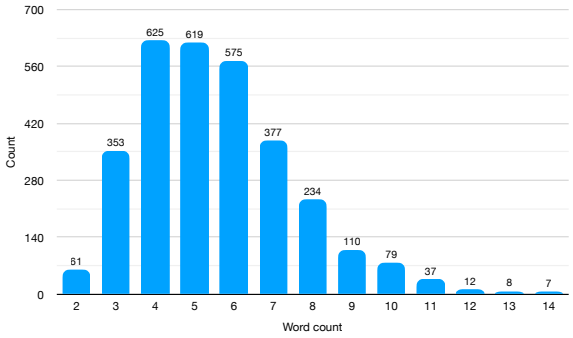

Figure 3. Word count distribution of combined description texts (x)

3.2. Fine-tuning IndoBERT

We fine-tune $\text{IndoBERT}_{\text{LARGE}}$ and $\text{IndoBERT}_{\text{BASE}}$ models using our dataset. We initially prepare the input data to conform to BERT's specific input format. BERT uses special token, [CLS] and [SEP]. [CLS] token added in the beginining of every input, while [SEP] token is used for separating two sentences. Converting text into BERT input format can be done by utilizing the tokenizer provided by the model. Following this step, we initialize the model and fine-tune it. During the fine-tuning phase, the model produces a sequence of 512 vectors as output. However, for our specific task, we only extract the first vector, which is distinctly denoted by the special [CLS] token. This vector encapsulates critical contextual information and is subsequently fed into a classifier. The classifier is composed of a fully connected layer with a softmax activation function, allowing it to make predictions across the 26 distinct classes associated with our classification task. The entire fine-tuning process is illustrated comprehensively in Figure 4.

3.3. Evaluation Metrics

An appropriate measure is needed to evaluate the performance of a multiclass classification model, especially if there is an imbalance in the data [17]. F_1-score is one of the commonly used evaluation metrics in classification tasks, especially when there is an imbalance between the number of samples in each class. F_1-score combines two other metrics, precision and recall, to provide a more comprehensive overview of the classification model's performance. Precision is the proportion of samples that are correctly classified as positive (TP) compared to the total number of samples classified as positive ($TP + FP$). Precision measures the extent to which the model's positive predictions are correct. Recall, also known as sensitivity or true positive rate, is the proportion of positive samples that are correctly classified (TP) compared to the total number of positive samples ($TP + FN$). Recall measures the extent to which the model can find all the existing positive samples. The F_1-score has a range of values between 0 and 1, where a value of 1 indicates perfect performance, while a value of 0 indicates very poor performance. Higher F_1-score values indicate better model performance in accurately classifying the positive class and the ability to find most of the positive samples.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{3}$$

In this study, the accuracy metric is not used because it tends to be biased for unbalanced data. We

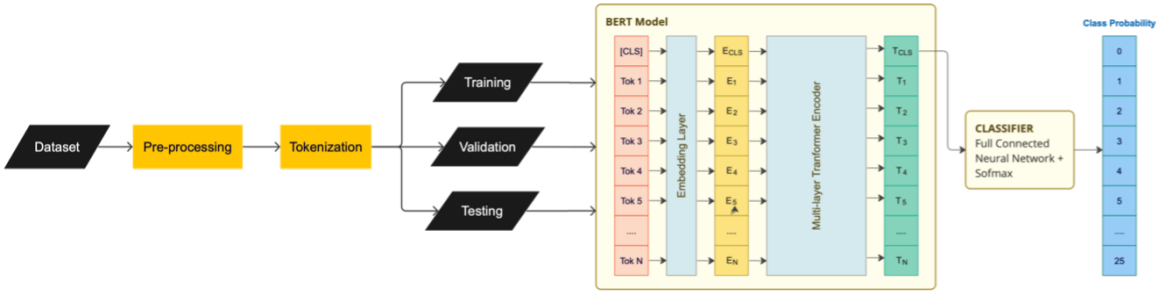

Figure 4. The fine-tuning processes of IndoBERT.

use Balance Accuracy (BA) which is more suitable for unbalanced data. It is designed to provide a fair assessment of the overall accuracy by considering the imbalance between the number of samples in different classes [25]. It is defined as the average of recall obtained on each class. We also calculated the False Positive Rate (FPR) and Imbalance Accuracy Metric (IAM). FPR is often used in evaluating the performance of classification models, especially in the context of class imbalance. False Positive Rate (FPR) is an evaluation metric that measures the proportion of predictions that are incorrectly classified as positive (false positives) compared to the total number of actual negatives. IAM is a performance metric specifically designed to address the issue of class imbalance in classification tasks [26]. IAM ranges from 0 to 1, where a value of 1 indicates perfect classification performance for both the majority and minority classes, regardless of their imbalance. A higher imbalanced accuracy score indicates better model performance in handling class imbalance. The calculation is presented in Formula 6.

$$FPR = \frac{FP}{TN} \tag{4}$$

$$BalanceAccuracy = \frac{1}{k}\sum_{i=1}^{k} Recall_i \tag{5}$$

$$IAM = \frac{1}{k}\sum_{i=1}^{k} \frac{c_{ii} - max(\sum_{j\neq i}^{k} c_{ij}, \sum_{j\neq i}^{k} c_{ji}}{max(c_{.i}, c_{i.})} \tag{6}$$

3.4. Experiment

The experiment was carried out with data of 3097 records, which were divided into training, validation, and testing sets with ratio 7:2:1. We employed the CatBoost and DistilBERT Model as the baseline models. The CatBoost model was found to be the top-performing method in [17], while Bechara et al. [19] reported that the DistilBERT model was the best-performing method. Two experimental scenarios were implemented: (1) CatBoost with text features; (2) fine-tuning IndoBERT$_{\text{BASE}}$, IndoBERT$_{\text{LARGE}}$, and DistilBERT.

Catboost is a gradient boosting algorithm on decision trees [27]. This technique involves sequentially adding weaker models to an ensemble model while fitting each new predictor to the residual errors made by the previous model [28]. This iterative process continues until the model meets certain criteria, such as a sufficiently small loss function value. CatBoost has several features. It supports numerical, categorical, and text features. It can be trained on both CPU and GPU, resulting in faster training times. Additionally, it can handle overfitting by employing a technique proposed by Prokhorenkova1 et al. [29]. When compared with other existing implementations of gradient-boosted decision trees, such as XGBoost, LightGBM1, and H2O2, CatBoost outperforms them on a diverse range of popular tasks. In the first experiment, due to its ability to automatically handle text features without requiring further pre-processing, data preprocessing is limited to lowercasing, stemming, and removing stopwords. The CatBoost model was then trained on this dataset with the identical hyperparameters as used by Aldania, et al. [17], specifically iterations of 1000, 2 L2 regularization of 2, 43 random seeds of 43.

In the second scenario, we fine-tuned the DistilBERT model over 10 epochs, adopting the epoch settings from Bechara et al. [19]. However, the paper did not specify the hyperparameter settings for other parameters such as the learning rate and batch size. Thus, we used the same values as the hyperparameter settings that we used for fine-tuning the IndoBERT models.

For the hyperparameter settings of the IndoBERT

model, we refer to the recommendations provided by Devlin et al. [12], which provide a range of optimal hyperparameter values across various NLP tasks, including classification tasks. The recommended range of values includes batch sizes between 16 and 32, epochs between 2 and 4, and learning rates between 5e-5 and 2e-5. In this work, we use the smallest value of the recommended learning rate, which is 2e-5. Despite the training process being slightly slower, this value provides a more stable model weight adjustment and minimises the risk of overshooting during optimization. Through experimentation, we found that four epochs provided the best performance without overfitting, while utilising more than 4 epochs did not result in a significant improvement in model performance. Additionally, a batch size of 16 was chosen to accommodate the memory capacity of the hardware available for this study, allowing the model to process a sufficient amount of data per iteration.

In order to ensure consistency, all experiments were conducted within the same computing environment. Google Colab was utilized as the experimentation platform, equipped with a Tesla T4 GPU with 15 GB memory and running on CUDA 11.2. The environment utilized 12 GB of RAM and 107 GB of available disk space. The training process for each model was relatively fast, with CatBoost requiring approximately 3.5 minutes, DistilBERT completing in 8 seconds, IndoBERT$_{\text{BASE}}$ finishing in 11 seconds, and IndoBERT$_{\text{LARGE}}$ taking 32 seconds. All models were trained with their respective hyperparameters as discussed earlier. Once trained, the models were evaluated on the test set, and their performance was measured in terms of F_1-score, balance accuracy, precision, recall, false positive rate, and imbalance accuracy metric, as shown in the results section.

4. Results and Analysis

This study investigates the effect of utilizing the Indonesian pre-trained language model based on Transformer to classify economic activities into 5-digit KBLI codes. Our work is different from previous studies that performed classification of economic activities into 5-digit KBLI codes using traditional machine learning CatBoost and Double Random Forest [17] and Gated Recurrent Unit (GRU) with word embedding from fastText [18]. All of these prior works have not yet investigated the use of state-of-the-art pre-trained language models, which have been shown to be effective in multiclass classification tasks. A recent prior study has utilized pre-trained language models to classify economic activities, but only into 2-digit codes of ISIC [19] and KBLI [30], which does not reflect the real settings that typically use 5-digit codes in practice.

In order to achieve the objectives mentioned above, we carried out two experimental scenarios. We discuss the experimental results and analysis in three sections. Section 4.1 provides a comparison of the performance between the baseline model and the proposed method. In this section, we also discuss the impact of data preprocessing on the performance of each model. Section 4.2 focuses on error analysis, and in Section 4.3, we elaborate on our statistical testing approaches to validate the significance of the obtained results.

4.1. Comparisons of Model Perfomance

Table 4. presents the performance comparison of different models in a multiclass classification task based on evaluation metrics such as F_1 score, Balanced Accuracy (BA), Precision, Recall, False Positive Rate (FPR), and IAM. We found that IndoBERT$_{\text{LARGE}}$ outperforms CatBoost, DistilBERT, and IndoBERT$_{\text{BASE}}$ on both preprocessed and unpreprocessed Dataset with an F_1-score of 96.82%, BA of 96.10%, and Recall of 96.96%, while also displaying a low FPR of 0.217 and a notable IAM value of 0.771. Achieving the highest IAM value, it demonstrated its capability to performs more accurate classification even on imbalance datasets. IndoBERT$_{\text{BASE}}$ and DistilBERT also performed well but slightly lower than IndoBERT$_{\text{LARGE}}$.

Table 4. The comparison results of using CatBoost and fine-tuned pre-trained language model.

Model	Evaluation Metrics					
	F_1	BA	Precision	Recall	FPR	IAM
Preprocessed Dataset						
CatBoost [17]	90.17	90.21	92.28	90.21	0.325	0.709
DistilBERT [19]	88.34	88.00	93.79	88.00	0.284	0.66
IndoBERT$_{\text{BASE}}$	86.98	86.40	93.15	86.40	0.325	0.632
IndoBERT$_{\text{LARGE}}$	92.28	92.20	95.49	92.20	0.217	0.771
Unpreprocessed Dataset						
CatBoost [17]	85.55	84.25	90.37	84.25	0.325	0.709
DistilBERT [19]	95.34	95.18	96.96	95.18	0.284	0.660
IndoBERT$_{\text{BASE}}$	96.22	95.87	**97.04**	95.87	0.325	0.632
IndoBERT$_{\text{LARGE}}$	**96.82**	**96.10**	96.94	**96.96**	**0.217**	**0.771**

Additionally, there is a significant difference in model performance between preprocessed and unprocessed datasets. The CatBoost model performs better on preprocessed datasets, indicating that data preprocessing steps such as stopword removal and stemming help reduce noise in the input data, allowing the model to focus on key terms and patterns that are critical for classification tasks. This

improvement occurs because CatBoost, as a tree-based model, relies heavily on structured and noise-free input features for optimal performance.

In contrast, pretrained language models such as DistillBERT and IndoBERT exhibit better results when using unprocessed data, where the input text remains in its natural language form. These models are specifically designed to capture complex linguistic patterns, semantic relationships, and contextual information within text. Removing stopwords or stemming words can disrupt the natural flow of language, potentially stripping away contextual clues that are vital for these models to fully understand the input. This is consistent with the findings of [31], which highlight that maintaining the original language structure enables pretrained language models to better comprehend linguistic context and nuances, resulting in more precise and relevant outcomes.

4.2. Error Analysis

An error analysis of IndoBERT$_{\text{LARGE}}$ was conducted using a confusion matrix. The confusion matrix is a table used to evaluate the performance of a classification model by comparing its predictions against the actual class labels of a dataset. As illustrated in Figure 5, the model accurately classified almost all test inputs, with only ten misclassifications. This error occurs from the model's limitations in handling context-specific keywords, lexical variations (e.g., typos or non-standard spellings), and imbalanced dataset distribution.

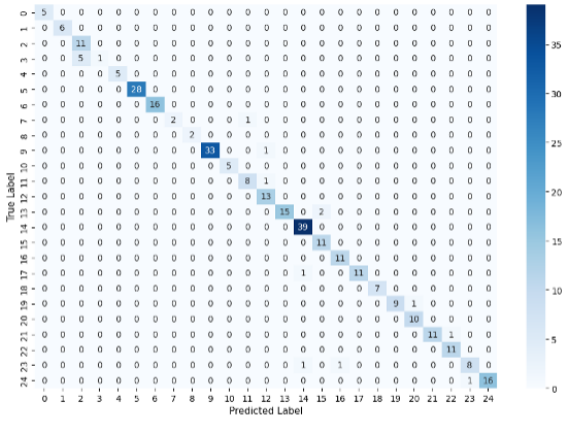

Figure 5. Multiclass confusion matrix of IndoBERT$_{\text{LARGE}}$.

A detailed analysis of these misclassifications is summarized in Table 5. The table show that the most common errors were caused by the model failing to capture the meaning of context-specific keywords. For instance, the input *"penjual kfc kfc"* was misclassified as food stall (56102) instead of the correct class restaurant (56101), as the model failed to recognize that *"kfc"* denotes a restaurant name. Similarly, the input *"warung bandrek minuman"* was misclassified as tavern (56303) instead of traditional herbal stall (56305). The model is unable to associate the term *"bandrek"*, which is a specific herbal drink, with the right classification and select a more general classification. Finally, in the third example in Table 5, the input *"membuat minuman jamu jamu gendong"* was classified as a traditional herbal stall (56305) instead of a mobile beverage vendor (56306). In this case, the model failed to capture the meaning of *"gendong"*, which refers to the method of selling by moving from one place to another while carrying a basket of jamu bottles. in the KBLI, there is a clear code distinction between selling jamu in a fixed place and selling jamu by moving from one place to another.

Table 5. Examples of common misclassifications by IndoBERT$_{\text{LARGE}}$

No	Input (x)	Groundtruth (y)	Prediction ($\hat{y}$)
Context-specific keywords			
1	*Penjual kfc kfc* (Kfc seller kfc)	Restaurant (56101)	Food stall (56102)
2	*Warung bandrek minuman* (Bandrek stall drink)	Traditional herbal stall (56305)	Tavern (56303)
3	*Membuat minuman jamu jamu gendong* (making herbal drinks jamu gendong)	Mobile beverage vendors (56306)	Traditional herbal stall (56305)
Typos or non-standard spellings			
4	*katring katring*	Event Catering (56210)	Food stall (56102)
Biased data distribution			
5	*kedai minuman kafe minuman ringan* (tavern soft drink cafe)	Cafeterias (56303)	Tavern (56303)

In addition, we observed that the model struggled with typos or non-standard spellings. As shown in the fourth example in Table 5, the input *"katring katring"* was misclassified as food stall (56102) instead of event catering (56210). The model cannot recognize that the word *"katring"* is a misspelled word from the word *"katering"*. Furthermore, utilization of an imbalanced training dataset can also contribute to classification errors, particularly when there are keywords in the input that make it classifiable into various different classes. For instance, the input *"kedai minuman kafe minuman ringan"*, which could be classified into either cafeteria (56303) or tavern (56303), was misclassified as a cafeteria (56303) instead of tavern (56303). This error occurs because the model is disproportionately influenced by the majority class during training, leading to a bi-

ased predictor. As a result, the model tends to rely on keywords that frequently occur in the majority class to classify inputs. In this case, the word *"kedai"* occurs more frequently in the tavern class than the word "kafe" in the cafeteria class, making the model incorrectly classify the input as tavern (56304). This shows how imbalanced training datasets can affect model predictions and lead to misclassification.

We also present the predictions made by each model in Table 6. All models correctly predict the input *"provides five-star hotel accommodation"*. However, only the IndoBERT model can classify the input "Provides 5-star hotel accommodation" accurately. This is because the IndoBERT model is trained on a large Indonesian corpus, which enables it to comprehend contextual and ambiguous aspects of Indonesian language more effectively. The CatBoost model does not convert text into word representations that can overcome semantic and contextual aspects of language, leading to inaccurate predictions for words absent from the training data. Although DistilBERT can understand contextual and ambiguous language, it has limitations in comprehending them in non-English language input texts.

Table 6. The prediction results of using Catboost and fine-tuning pre-trained model.

Input	Model	Prediction	Groundtruth
Menyediakan akomodasi hotel bintang lima (Provides five-star hotel accommodation)	CatBoost [17]	55111	55111
	DistilBERT [19]	55111	
	$\text{IndoBERT}_{\text{BASE}}$	55111	
	$\text{IndoBERT}_{\text{LARGE}}$	55111	
Menyediakan akomodasi hotel bintang 5 (Provides 5-star hotel accommodation)	CatBoost [17]	55114	**55111**
	DistilBERT [19]	55114	
	$\text{IndoBERT}_{\text{BASE}}$	55111	
	$\text{IndoBERT}_{\text{LARGE}}$	55111	

4.3. Statistical Analysis

To further validate the performance superiority of $\text{IndoBERT}_{\text{LARGE}}$, we applied the McNemar's test, a nonparametric method commonly used in binary classification problems to compare the performance of two classifiers. The test is based on a 2x2 contingency table of the predictions made by two models, as illustrated in Figure 6. The value n_{11} represents the number of items that were correctly classified by both models, while n_{01} indicates the number of items that were misclassified by model A but not by model B. Similarly, n_{10} indicates the number of items that were misclassified by B but not by model A, and n_{00} indicates the number of items misclassified by both model A and B.

In the McNemar test, the null hypothesis assumes that the two models have the same error rates.

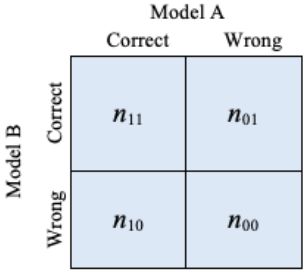

Figure 6. Contingency tables.

It is basically a form of paired chi-square test (with 1 degree of freedom), which is calculated as follows:

$$\chi^2 = \frac{(|n_{11} - n_{00}| - 1)^2}{n_{11} + n_{00}} \tag{7}$$

If the calculated p-value (probability value) associated with the test statistic is less than a chosen significance level (in this study, we set α = 0.05), then the null hypothesis is rejected, indicating a significant difference in performance between the two models.

Table 7. McNemar's test for $\text{IndoBERT}_{\text{LARGE}}$ againts existing methods (α = 0.05).

Model	χ^2 statistics	p-value	Null Hypothesis
CatBoost [17]	12.07	< 0.01	Reject H_0
DistilBERT [19]	24	0.72	Failed to reject H_0
$\text{IndoBERT}_{\text{BASE}}$	6	0.62	Failed to reject H_0

Table 7 presents the outcomes of McNemar's test for $\text{IndoBERT}_{\text{LARGE}}$ againts existing methods with a significance level (α) of 0.05. For CatBoost, the Chi-square statistic is 12.07 with a p-value of less than 0.01. Since the p-value is below the significance level, the null hypothesis is rejected, indicating a significant difference in performance between $\text{IndoBERT}_{\text{LARGE}}$ and CatBoost. On the other hand, DistilBERT and $\text{IndoBERT}_{\text{BASE}}$ yield Chi-square statistics of 24 and 6, respectively, with p-values above 0.05. Consequently, it failed to reject the null hypothesis, indicating that there is no significant difference in performance between $\text{IndoBERT}_{\text{LARGE}}$ and these models.

5. Conclusion

In this study, we propose implementing a transformer-based language model pretrained on a large Indonesian corpus to classify economic activity in Indonesia. Our findings reveal that our

proposed approach, known as IndoBERT$_{\text{LARGE}}$, consistently outperforms the baseline model. Specifically, it achieves a F_1 score of 96.82%, a balanced accuracy (BA) of 96.10%, and recall at 96.96%. Furthermore, IndoBERT$_{\text{LARGE}}$ exhibits a low false positive rate (FPR) of 0.217% and a notable Imbalance Accuracy Metric (IAM) value of 0.771. Our study also highlights the significant performance enhancements achieved by this approach. Notably, it can improve the F_1 score of the baseline model by up to 7.37% for Catboost and by 1.55% for DistilBERT model. Due to the slight difference in performance exhibited by IndoBERT$_{\text{BASE}}$ and DistilBERT, they are still promising to be used as a predictor.

Although our approach has yielded valuable insights, the limited dataset used may pose a potential constraint. Despite the fact that the 5-digit KBLI code encompasses 1573 categories, we only focused on 26 classes in our analysis. The original dataset we acquired still contained numerous misclassified records, which necessitated manual reclassification into the 26 classes due to time constraints. To ensure the accuracy of our classification, further investigation is necessary. This includes expanding the scope of classification classes and addressing potential misclassifications of certain economic activities due to data limitations. A more extensive dataset would facilitate a thorough examination of these aspects.

References

[1] N. V. Babu and E. G. M. Kanaga, "Sentiment Analysis in Social Media Data for Depression Detection Using Artificial Intelligence: A Review," *SN Computer Science*, vol. 3, no. 1, p. 74, Jan. 2022.

[2] H. Chen, L. Wu, J. Chen, W. Lu, and J. Ding, "A comparative study of automated legal text classification using random forests and deep learning," *Information Processing & Management*, vol. 59, no. 2, p. 102798, Mar. 2022.

[3] C. Colón-Ruiz and I. Segura-Bedmar, "Comparing deep learning architectures for sentiment analysis on drug reviews," *Journal of Biomedical Informatics*, vol. 110, p. 103539, Oct. 2020.

[4] G. Rabby and P. Berka, "Multi-class classification of COVID-19 documents using machine learning algorithms," *Journal of Intelligent Information Systems*, vol. 60, no. 2, pp. 571–591, Apr. 2023.

[5] K. Purwandari, J. W. Sigalingging, T. W. Cenggoro, and B. Pardamean, "Multi-class Weather Forecasting from Twitter Using Machine Learning Aprroaches," *Procedia Computer Science*, vol. 179, pp. 47–54, 2021.

[6] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1–27, Apr. 2011.

[7] A. McCallum and K. Nigam, "A Comparison of Event Models for Naive Bayes Text Classification," *Proc. AAAI/ICML-98 Workshop on Learning for Text Categorization*, pp. 41–48.

[8] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967.

[9] L. Breiman, J. Friedman, C. Stone, and R. Olshen, *Classification and Regression Trees*. Wadsworth, New York: Chapman and Hall, 1984.

[10] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," Sep. 2013, arXiv:1301.3781 [cs].

[11] J. Pennington, R. Socher, and C. Manning, "Glove: Global Vectors for Word Representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, 2014, pp. 1532–1543.

[12] 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 American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186.

[13] C. Wang, P. Nulty, and D. Lillis, "A Comparative Study on Word Embeddings in Deep Learning for Text Classification," in *Proceedings of the 4th International Conference on Natural Language Processing and Information Retrieval*. Seoul Republic of Korea: ACM, Dec. 2020, pp. 37–46.

[14] N. Rai, D. Kumar, N. Kaushik, C. Raj, and A. Ali, "Fake News Classification using transformer based enhanced LSTM and BERT," *International Journal of Cognitive Computing in Engineering*, vol. 3, pp. 98–105, 2022.

[15] S. Cheon and I. Ahn, "Fine-Tuning BERT for Question and Answering Using PubMed Abstract Dataset," in *2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*.

Chiang Mai, Thailand: IEEE, Nov. 2022, pp. 681–684.

[16] B. Wilie, K. Vincentio, G. I. Winata, S. Cahyawijaya, X. Li, Z. Y. 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 International Joint Conference on Natural Language Processing*. Suzhou, China: Association for Computational Linguistics, Dec. 2020, pp. 843–857.

[17] A. N. A. Aldania, A. M. Soleh, and K. A. Notodiputro, "A Comparative Study of CatBoost and Double Random Forest for Multiclass Classification," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 1, pp. 129–137, Feb. 2023.

[18] M. Dwicahyo and B. Yuniarto, "Deep Learning for Indonesia Standard Industrial Classification," in *2020 International Conference on Electrical Engineering and Informatics (ICELTICs)*. Aceh, Indonesia: IEEE, Oct. 2020, pp. 1–6.

[19] H. Bechara, R. Zhang, S. Yuan, and S. Jankin, "Applying NLP Techniques to Classify Businesses by their International Standard Industrial Classification (ISIC) Code," in *2022 IEEE International Conference on Big Data (Big Data)*. Osaka, Japan: IEEE, Dec. 2022, pp. 3472–3477.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 6000–6010.

[21] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep Mutual Learning," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT: IEEE, Jun. 2018, pp. 4320–4328.

[22] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," Feb. 2020, arXiv:1910.01108 [cs].

[23] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations," *CoRR*, vol. abs/1909.11942, 2019, arXiv: 1909.11942.

[24] M. Adriani, J. Asian, B. Nazief, S. Tahaghoghi, and H. Williams, "Stemming indonesian: A confix-stripping approach," *ACM Transactions on Asian Language Information Processing*, vol. 6, no. 4, Dec. 2007.

[25] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann, "The Balanced Accuracy and Its Posterior Distribution," in *2010 20th International Conference on Pattern Recognition*. Istanbul, Turkey: IEEE, Aug. 2010, pp. 3121–3124.

[26] E. Mortaz, "Imbalance accuracy metric for model selection in multi-class imbalance classification problems," *Knowledge-Based Systems*, vol. 210, p. 106490, Dec. 2020.

[27] A. V. Dorogush, V. Ershov, and A. Gulin, "CatBoost: gradient boosting with categorical features support," *arXiv preprint arXiv:1810.11363*, 2018.

[28] A. Geron, "Ensemble Learning and Random Forests," in *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Inc, 2019.

[29] A. V. Dorogush, V. Ershov, and A. Gulin, "Catboost: gradient boosting with categorical features support," *CoRR*, vol. abs/1810.11363, 2018. [Online]. Available: http://arxiv.org/abs/1810.11363

[30] L. H. Suadaa, F. Ridho, A. K. Monika, and N. W. K. Projo, "Automatic Text Categorization to Standard Classification of Indonesian Business Fields (KBLI) 2020," in *2023 International Conference on Electrical Engineering and Informatics (ICEEI)*. Bandung, Indonesia: IEEE, Oct. 2023, pp. 1–6.

[31] Z. Dai and J. Callan, "Deeper Text Understanding for IR with Contextual Neural Language Modeling," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. Paris France: ACM, Jul. 2019, pp. 985–988.