IndoBERTSkill: pretrained domain-specific language model for recognition of Indonesian skill

Meilany Nonsi Tentua [a,1], Suprapto Suprapto [b,2,*], Afiahayati Afiahayati [b,3]

[a] Universitas PGRI Yogyakarta, Yogyakarta, Indonesia
[b] Universitas Gadjah Mada, Yogyakarta, Indonesia
[1] meilany@upy.ac.id; [2] sprapto@ugm.ac.id; [3] afia@ugm.ac.id
* corresponding author

ARTICLE INFO

Article history
Received August 29, 2023
Revised February 20, 2026
Accepted March 8, 2026
Available online May 31, 2026

Keywords
BERT
Skill Recognition
Named Entity Recognition
domain-specific
Pretrained Language Model

ABSTRACT

The pretrained language model in Indonesian is already available for natural language processing tasks. However, this pre-trained model has been trained on Indonesian text, which has a different structure from the job description. Due to this, the pre-trained language model is less effective for skill recognition purposes. IndoBERTSkill is a novel pre-trained domain-specific language model that recognizes Indonesian language skills. It is built on the Bidirectional Encoder Representations from Transformers (BERT) architecture. IndoBERTSkill was trained on an extensive collection of Indonesian language texts from the Indonesian Wikipedia, the English Wikipedia, and Indonesian job descriptions from the job portal. IndoBERTSkill's performance was evaluated through two main approaches: (1) language modeling via Masked Language Model (MLM) prediction, and (2) fine-tuning on a custom annotated dataset (NERSkill) for Named Entity Recognition (NER) tasks. The fine-tuning process involved training a classification layer on top of the IndoBERTSkill model using BIO tagging to identify hard skills, soft skills, and technology entities. Similarly, the skill recognition model derived from IndoBERTSkill exhibits the highest F1-Score among various pre-trained language models, precisely at 87%, thus demonstrating robustness and strong generalizability for skill entity recognition in Indonesian job descriptions. IndoBERTSkill provides valuable resources for developing Indonesian natural language processing applications that require skills introduction. This could increase the accuracy and efficiency of skills recognition across various domains, including job matching, education, and training.

1. Introduction

The Bidirectional Encoder Representations from Transformers (BERT) method has been widely used in various Natural Language Processing (NLP) tasks due to its highly impressive performance [1], [2]. BERT is a Transformer-based model [3] that implements two training strategies based on the mechanism of attention to Language Modeling [4]. The language model built on pre-trained BERT is trained using the Masked Language Model (MLM) and Next Sentence Prediction (NSP) approaches [5]. Deep learning-based language models trained on large amounts of unannotated text have been developed for transfer learning efficiency in NLP [6], [7]. To correctly predict words and use them for other tasks, it must be trained repeatedly on large amounts of data. The results of the previously trained model will be used in other specific tasks, such as text classification [8], sentiment analysis [9], [10], POS Tagging [11], and Named Entity Recognition (NER) [12], by updating the weights, referred to as

fine-tuning. The fine-tuning strategy is widely used because it does not require much labeled data and can be more efficient in building models for specific tasks in NLP [13].

Several researchers have created domain-specific language models to resolve domain-specific problems [14], [15], [16], [17]. These models are built by training the BERT architecture from scratch on a domain-specific corpus or by continuous training on models with a similar domain. A Pre-trained Language Model (PLM) must have a suitable vocabulary specific to the domain, unlike the original BERT model [18]. A suitable domain-specific vocabulary solves NLP problems well [19].

Currently, when considering the Indonesian Language, numerous well-known pre-trained language models are accessible, including multilingual BERT [4], IndoBERT [20], IndoNLU [21], and IndoBERTweet [22]. These models serve as pre-trained solutions for processing Indonesian text. They were trained using the same architecture and pre-training methodology as BERT, albeit on a substantial corpus of Indonesian text. These models have demonstrated exceptional performance in various Indonesian NLP tasks, encompassing sentiment analysis [23], [24], text classification [25], named entity recognition [26], [27], and question answering [28].

Sentences of job requirements in job vacancies have a different structure than formal sentences, which have a subject, predicate, and object structure. The sentence structure in the job description consists of several sentences arranged in Predicate, Object, and Description. Fig. 1 shows a sample of job vacancies in the Indonesian Language.

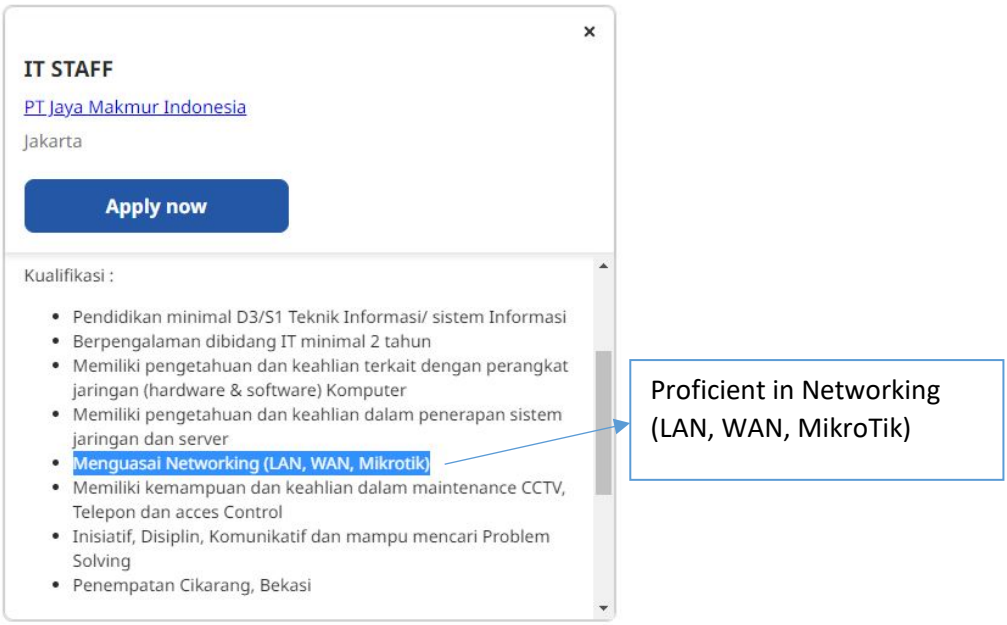


Fig. 1. A sample of job vacancies in the Indonesian Language.

The presentation of information on job vacancies often uses a mixed language, Indonesian and English. This problem causes the identification of skill entities in job vacancies to require a pre-trained BERT model trained on the corpus of job requirements to get the best performance when fine-tuning BERT. Fig. 2 shows a sample of recognition of Indonesian skills.

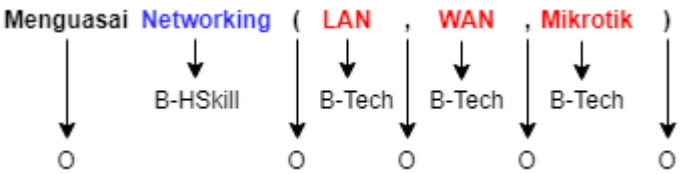


Fig. 2. A sample of recognition of Indonesian skills.

Researchers have explored Skill Extraction from Job Descriptions using machine learning methods such as SVM [29], Naïve Bayes [30], Linear Regression [31], and XGBoost [32]. However, a growing trend among researchers is using pre-trained models that have undergone training on extensive datasets and can be fine-tuned for specific tasks. Adopting pre-trained models offers advantages by achieving high accuracy even with limited training data [33], thereby reducing the time and cost required to develop a custom skill extraction model [34]. Skill Extraction from Job Descriptions using pre-trained models has been conducted across various languages, including English [35], [36], [37], Russian [38], China [39], German [40], Spain [41], and Lithuania [42]. The emphasis on pre-trained language models for Named Entity Recognition (NER) has been highlighted, and a comprehensive survey of existing models has been provided in this context.

In this paper, we present IndoBERTSkill, a novel pre-trained language model for skill recognition in job descriptions based on BERT, which efficiently reduces the need to spend high-quality labeled skill entity data. In contrast to existing general-purpose Indonesian PLMs such as IndoBERT and multilingual PLMs like mBERT, this study introduces a novel domain-specific language model, IndoBERTSkill, which is pre-trained exclusively on Indonesian job descriptions and optimized for skill entity recognition. This domain encompasses job vacancy content collected from multiple Indonesian job portals, characterized by informal formatting, non-standard sentence structures, and extensive code-mixing between Indonesian and English.

This work contributes to the development of domain-specific Indonesian NLP by introducing two main deliverables. First, we perform domain-adaptive pretraining using an MLM objective on a job description corpus enriched with Indonesian-English mixed-language structures. This results in IndoBERTSkill, a language model checkpoint specifically adapted to the job recruitment domain, along with its vocabulary file. Second, we fine-tune IndoBERTSkill for a NER task to extract skill entities, categorized as hard skills, soft skills, and technology, using our annotated NERSkill dataset. This leads to a second deliverable: the fine-tuned NER model and labeled dataset for benchmarking. These two tasks, MLM-based domain adaptation and NER-based skill extraction, represent complementary components for understanding job descriptions in Indonesian.

2. Method

This part explains the methodology for developing the IndoBERTSkill model to extract skills from job requirements. Several steps were taken in the methodology: data pre-processing, pre-training the model language, and NER modelling for skill recognition. An overview of our proposed model is presented in Fig. 3.

2.1. Data Pre-processing Component

We collected job descriptions from four Indonesian job portals, specifically Indeed, Jobstreet, loker.id, and Job.id. Job description data is obtained by extracting data from the job portals using the BeautifulSoup Python Library. Subsequently, standard data pre-processing techniques, including converting the text to lowercase and removing non-textual symbols, are applied to the 34,966 job descriptions, covering 55 unique job categories from the IT sector. The accumulated data was split into two separate datasets: the NERSkill dataset, which contains 4,394 job descriptions, and the Pretrain dataset, which includes 30,354 job descriptions. Fig. 4 shows sample data from the job descriptions.

The Pretrain dataset for training the IndoBERTSkill model has no annotations. Compared to the data used to train IndoBERT (220 million words) [20] or IndoBERTweet (409 million words) [22], the Pretrain dataset is relatively small. To address this issue, we have included the Indonesian Wikipedia corpus, similar to the corpus used in pre-training the IndoBERT model. Furthermore, since the job requirements dataset includes Indonesian and English, we have also incorporated the English Wikipedia to improve the training data. So the datasets used in IndoBERTSkill combine the Indonesian Wikipedia corpus1, English Wikipedia2, and the job requirements dataset. The dataset used in IndoBERTSkill has 162 million words.

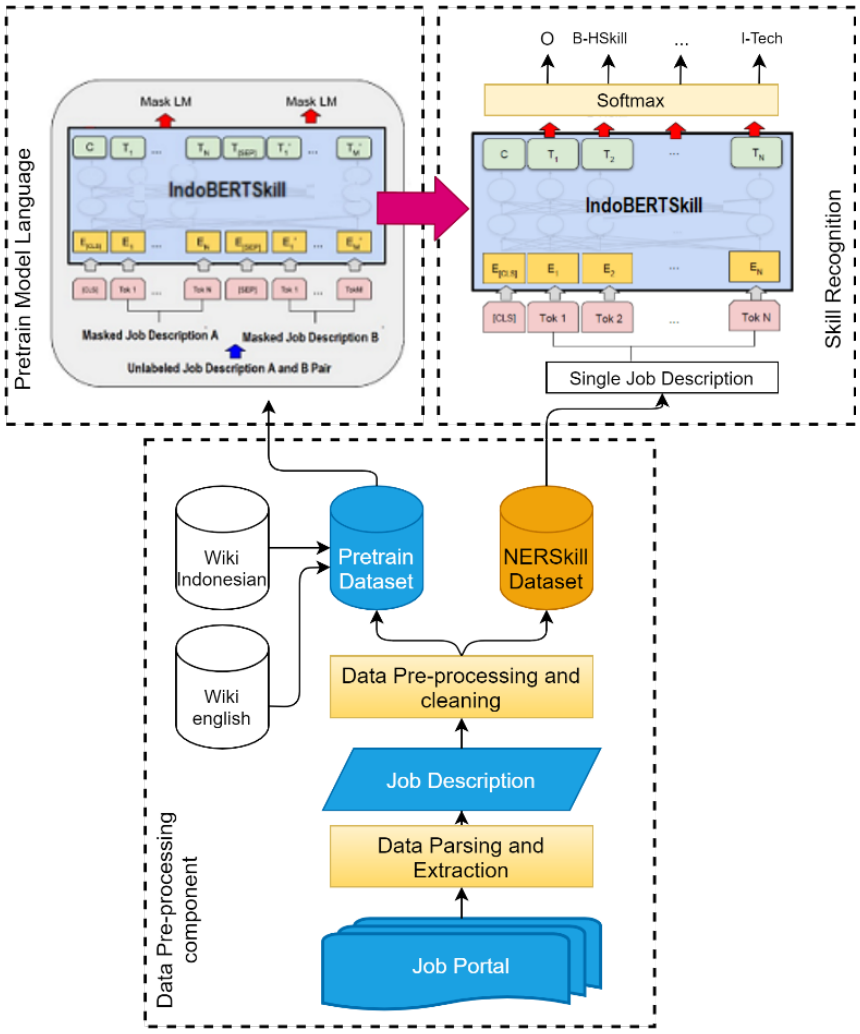


Fig. 3. System Architecture of Our Proposed.

0	1. menguasai basic pemotretan dan videography ...
1	Pengalaman Min 1 tahun di bidang quality. Kea...
2	Pengalaman 2 tahun. Fresh Graduate diperbolehk...
3	1. Membantu proses improvement di lingkungan ...
4	Min. 1 tahun dibidang yang sama Keahlian : M...

Basic knowledge of photography and videography.

Minimum 1 year of experience in quality.

2 years of experience. Fresh graduates are welcome to apply.

Assist in process improvement within the workplace.

Minimum 1 year of experience in the same field. Skills: ...

Fig. 4. Sample data of the job description.

The NER modeling dataset differs from the one used in the pre-trained language model IndoBERTSkill. Job Requirements for the dataset used in the NER-Skill modeling are annotated. This dataset is annotated in the BIO (Beginning, Inside, and Outside) tagging format [43]. The job requirements used in the NER-Skill modeling are 4,394 rows. We use the NLTK Python Library to process word tokenization in the job description. Word tokenization generated from pre-processing job descriptions is made into a data frame and then manually labeled by the annotator. Data annotation aims to identify hard skills (B-HSkill and I-HSkill), soft skills (B-SSkill and I-SSkill), and Technology entities (B-Tech and I-Tech). Writing the NERSkill dataset uses the CoNLL 2003 format, consisting of 3 columns: sentences, words, and tags. Table 1 shows sample annotated data written in CoNLL 2003 format.

Table 1. The annotated data sample in the NERSkill dataset.

Sentence#	Word	Tag
sentence 1	*mengerti* / proficient	O
	Jaringan / network	B-HSkill
	,	O
	hosting	B-HSkill
	Server	I-HSkill
	,	O
	program	B-Tech
	php	I-Tech
	,	O
	git	B-Tech

The NERSkill dataset has considerable class imbalance, with most tokens labelled as non-entities (O), representing 84.49% of the total data.. In contrast, skill-related entities are much less frequent, including B-Tech (4.13%), B-HSkill (4.06%), I-HSkill (2.75%), B-SSkill (2.60%), I-Tech (1.05%), and I-SSkill (0.92%).

2.2. Vocabulary

Before job descriptions are utilized as inputs for constructing the IndoBERTSkill language model, we must create a set of tokens for word-level tokenization in the job descriptions. The vocabulary (vocab) in IndoBERTSkill consists of a collection of tokens or subwords that are used to train and evaluate the IndoBERTSkill model. The vocab in IndoBERTSkill is established through WordPiece tokenization [1], a tokenization technique that maps words or phrases within text to smaller tokens or subwords. Each token present in the IndoBERTSkill vocab is assigned a distinct numeric index. The vocabulary structure in IndoBERTSkill consists of 31,923 tokens. The input text undergoes a conversion process into a sequence of numbers (word embeddings), presented in vector form, which can subsequently be input into the IndoBERTSkill model.

2.3. Pretain Model Language IndoBERTSkill

IndoBERTSkill is built on the Transformer architecture, an artificial neural network that processes sequential data, such as job descriptions, by paying attention to different parts of the input. In the context of Transformer, attention is a mechanism used to compute the relevance of various parts of the input sequence when output is generated. Word embeddings as input sequences are then given positional information using Positional Encodings. These encodings are calculated based on the position of tokens in the sequence and are designed to provide unique positional information for each token. This allows the model to understand the importance of token positions and their semantic meaning. Positional encoding is applied using Equations (1) and (2).

$$k_{(i,2p)} = \sin\left(\frac{i}{10000^{\frac{2p}{d}}}\right) \quad (1)$$

$$k_{(i,2p+1)} = \cos\left(\frac{i}{10000^{\frac{2p}{d}}}\right) \quad (2)$$

which k refers to 'Positional Encoding' , i stands for the position of the word , p refers to embedding of the feature, and d is the word's embedding size. Equation (3) is used for each token's positional encoding.

$$s_i = t_i + k_i \quad (3)$$

The attention mechanism in transformers computes weights for each input token based on its relation to the present output token [3]. These weights are then used to determine the weighted sum of the input tokens, which is used to create the output token.

The input sequence $\boldsymbol{s} = (s_1, s_2, \ldots, s_m)$ m is the length of the token (we use m=256). Each $s_i \in \mathbb{R}^d$ having dimension d=768. The input sequence s will projected through three matrices $W^Q \epsilon\, \mathbb{R}^{mxd_q}$, $W^K \epsilon\, \mathbb{R}^{mxd_k}$ and $W^V \epsilon\, \mathbb{R}^{mxd_v}$, to yield representations of features Q, K, and V, referred to as Query,

Key, and Value, respectively, with dimensions $d_k = d_q = 64$. The *Q*, *K*, and *V* matrices are computed utilizing equation (4).

$$Q = s_i W^Q , K = s_i W^K , V = s_i W^V \quad (4)$$

The sequence input will produce attention values $\boldsymbol{z} = (z_1, z_2, \ldots, z_n)$ of the length equal to input $z_i \in \mathbb{R}^{d_z}$. Each attention element is calculated using equation (5).

$$z_i = \sum_{j=1}^{n} \alpha_{ij}(s_j W^V) \quad (5)$$

Each weight coefficient is calculated using the softmax function in equation (6).

$$\alpha_{ij} = \frac{\exp e_{ij}}{\sum_{k=1}^{n} \exp e_{ik}} \quad (6)$$

Each e_{ij} is calculated using equation (7).

$$e_{ij} = \frac{(s_i W^Q)(s_j W^K)^T}{\sqrt{d_k}} \quad (7)$$

The key innovation of IndoBERTSkill is its capability to train language models on Pretrain datasets utilizing the Masked Language Model (MLM) objective [4].

In this study, we trained IndoBERTSkill models from the ground up using the earlier configuration. The training process for IndoBERTSkill employed the MLM objective, as done in previous research [4]. Specifically, we randomly selected 15% of tokens and then replaced 80% with "[MASK]", replaced another random 10% of tokens, and left 10% of the tokens unaltered. The IndoBERTSkill training approach is similar to that of IndoBERT [20]. We used a Transformer encoder with 12 hidden layers (dimension = 768), 12 attention heads, and three hidden feedforward layers (dimension = 3,072). The only distinction is the maximum sequence length, which was fixed at 256 tokens based on the average number of tokens in the job description. We used a learning rate of 1e-4, a batch size of 64, a dropout rate of 0.1, warmup steps of 10,000, weight decay of 10, and 300 epochs (448.500 steps). We apply gradient accumulation every 4 steps and do not use early stopping. Overfitting is monitored through validation loss.

2.4. Skill Recognition

To extract skills from job descriptions, a NER model is utilized. This NER model is fine-tuned on the pre-trained IndoBERTSkill language model. The training process uses the NERSkill dataset with a strategy of updating the IndoBERTSkill model weights and classification heads to optimise model performance. The NER Skill model adds a linear layer and softmax activation to identify entities. We selected one hyperparameter recommended by Devlin et al.'s study [4] in fine-tuning the model: a learning rate of 3e-5, dropout = 0.2, epoch = 5, and batch size = 32.

Training and evaluation of models in the fine-tuning phase used a maximum sequence length of 128 and 256 tokens. To handle job descriptions that exceeded this length, we used a truncation strategy. This was done because the average number of words in job postings in the NERSkill dataset was 133.48 words.

2.5. Evaluation

We evaluated the model's ability to predict the next word in a sequence of words in the IndoBERTSkill language model. NER model was evaluated at two levels: token level and entity level. At the token level, metrics are calculated as independent token classification and use micro-F1. At the entity level, the CoNLL 2003 evaluation protocol is used, where predictions are considered correct only if the label and entity span match exactly. Partial predictions are considered incorrect. The main F1 score reported is micro-F1 at the entity level, in accordance with NER evaluation standards.

We use a confusion matrix to obtain the micro-F1 value. The matrix will have four cells representing the possible outcomes of each prediction: true positive, false positive, true negative, and false negative. F5 shows a table of the confusion matrix.

		Prediction Values	
		Positive	Negative
Actual Value	Positive	true positive	false negative
	Negative	false positive	true negative

Fig. 5. The confusion matrix.

Based on this confusion matrix, each entity is evaluated by using Precision, Recall, and micro F1-Score [44]. The metrics used for the model's evaluation are calculated using equations 5, 6, and 7. All evaluations were performed on varying token lengths (128 and 256) to test the model's capability to generalize across different input sequences.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{8}$$

$$Recall = \frac{True\ Positive}{True\ Positive\ \_False\ negative} \tag{9}$$

$$F1\ score = \frac{2\ x\ Precision\ x\ Recall}{Precision + Recall} \tag{10}$$

3. Results and Discussion

In this section, we will explain the results of pre-trained IndoBERTSkill by doing [MASK] in the job requirements sentence. The NER model is evaluated by calculating precision, recall, and F1-score.

3.1. IndoBERTSkill

The IndoBERTSkill language models were trained from scratch on an NVIDIA A100-SXM machine. The IndoBERTSkill training model used hyperparameters including a learning rate of 1e-04, a dropout rate of 0.1, a batch size of 64, 300 epochs, and 448.500 steps, resulting in 110.591.667 parameters. The vocab size used in IndoBERTSkill is 31.923, the same as that used in IndoBERT [20]. As the baseline performance to evaluate the IndoBERTSkill pre-trained model language, we use Multilingual BERT [4] and IndoBERT [20]. Table 2 shows a sample result of tokenizing a job description using vocabulary in the IndoBERTSkill pre-trained model language and the baseline.

Table 2. A Sample result of the tokenized job description.

Comparison	Model	Sample Job description: *"mengerti* (Proficient) *jaringan* (network), hosting server, program PHP *dan* (and) git"
	Vocabulary	**Result**
Baseline	BERT Multilingual by Vaswani [4]	['[CLS]', 'men', '##gert', '##i', 'jaringan', ',', 'hosting', 'server', ',', 'program', 'php', 'dan', 'gi', '##t', '[SEP]']
	Koto et al. [20]	['[CLS]', *'mengerti'*, *'jaringan'*, ',', 'hosting', 'server', ',', 'program', 'php', 'dan', 'gi', '##t', '[SEP]']
Our Propose	**IndoBERTSkill**	**['[CLS]', *'mengerti'*, *'jaringan'*, ',', 'hosting', 'server', ',', 'program', 'php', 'dan', 'git', '[SEP]']**

Table 2 shows the word "git" when tokenized using IndoBERTSkill; it is recognized as a single word, "git". However, in Multilingual BERT [4] and IndoBERT [20], it is treated as an out-of-vocabulary word and tokenized into "gi" and "#t". This was the result of the word "git," which is a technology name

frequently appearing in job descriptions. These differences in tokenization can affect the performance of pre-trained language models when predicting words found in job description sentences.

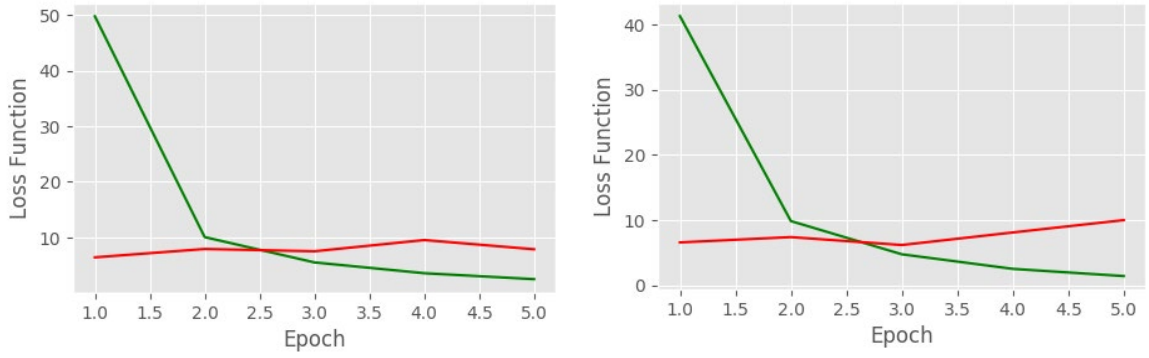
We evaluated the performance of the IndoBERTSkill model by performing '[MASK]' on the sample job description sentences. Furthermore, the IndoBERTSkill models will predict the covered word. Table 3 displays the hidden word prediction results for the example job description sentence, which can be performed very well by the IndoBERTSkill pre-trained models compared to the BERT Multilingual and IndoBERT pre-trained models. The pre-trained BERT Multilingual and IndoBERT models can recognize Indonesian sentences but cannot recognize specific words in job description sentences. Therefore, during IndoBERTSkill training, a vocabulary set contains specific words from job description sentences. This vocabulary set is then used for sentence tokenization during training.

Table 3. A Sample result of predicting [MASK] of the word.

Comparison	Model	Sample Job description: *"pengalaman* (Experience) *sebagai* (as) *designer*(designer) *[MASK]/ux selama* (for) *5 tahun* (year)"	Prob.	Result
Baseline	BERT Multilingual [4]	pengalaman sebagai designer U/ux selama 5 tahun	0.17	False
	IndoBERT [20]	pengalaman sebagai designer b/ux selama 5 tahun	0.04	False
Our Propose	**IndoBERTSkill**	**pengalaman sebagai designer ui/ux selama 5 tahun**	**0.959**	**True**

3.2. NER Skill Modelling

We performed fine-tuning on the pre-trained IndoBERTSkill on the NERSkill dataset for NER-Skill modelling. We used the Adam optimiser with a learning rate of 3e-5, five epochs, and a batch size of 32. Fig. 6 shows the training loss on IndoBERT fine-tuning. Because the job descriptions have token lengths between 128 and 256, we evaluated our model's performance through fine-tuning, employing two different token lengths: 128 and 256. Fig. 6 shows training loss and validation loss during fine-tuning IndoBERTSkill. Fig. 2 shows that in the third epoch, validation loss starts to increase, even though training loss continues to decrease. The decision to set the hyperparameter to 5 epochs for fine-tuning in IndoBERTskill suggests that the NER-Skill model maintains a favorable level of generalization.

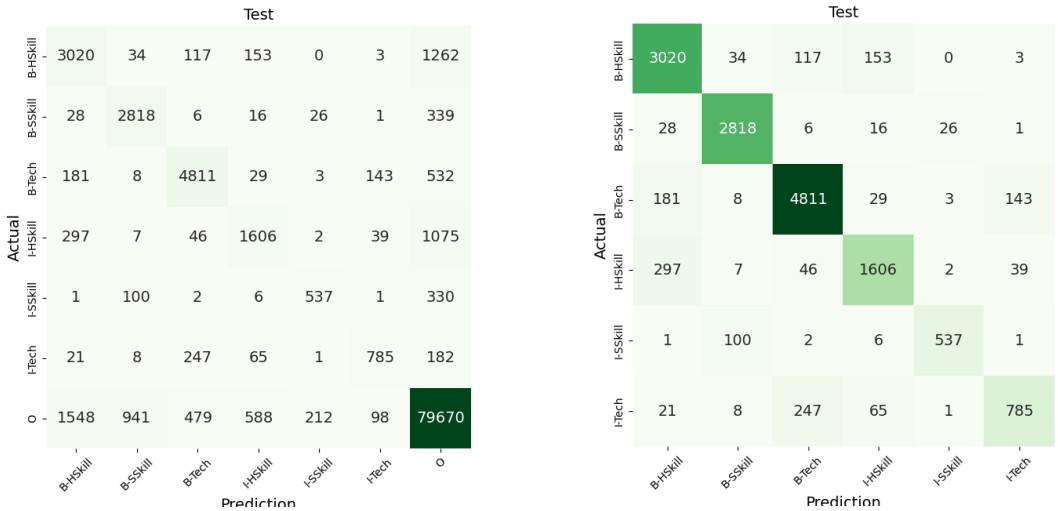


(a) **(b)**

Fig. 6. Training Loss and Validation Loss on Fine-Tuning IndoBERTSkill.

We use the confusion matrix to evaluate IndoBERTSkill's performance in recognizing Indonesian language skills. Model performance is evaluated at the token level and entity level. At the token level, model performance is calculated based on all entities. Meanwhile, at the entity level, model performance is calculated without including entity O. Fig. 7 illustrates the confusion matrix results for a maximum token length of 128, whereas Fig. 8 illustrates the results for a maximum token length of 256.

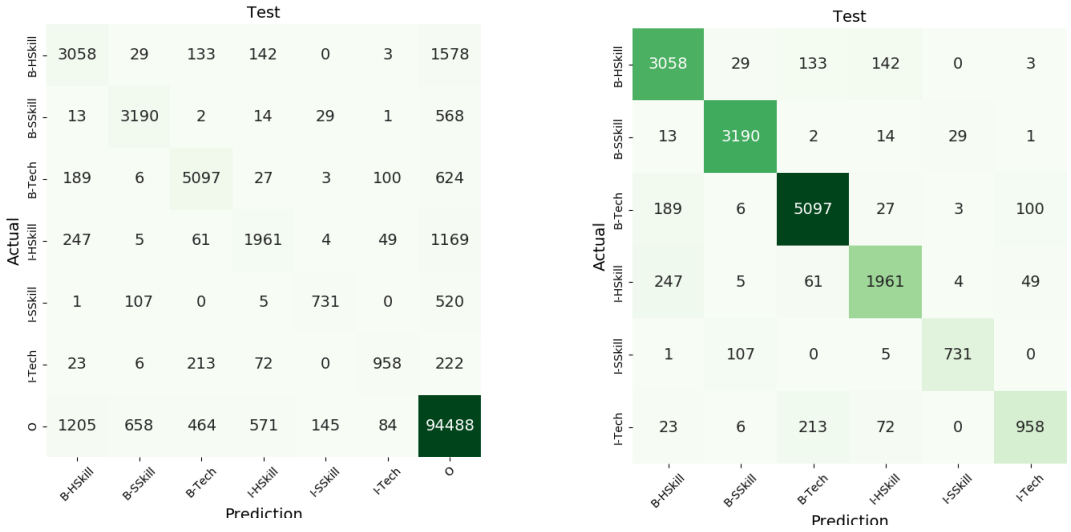
Fig. 7(a) shows that the model evaluated using a confusion matrix at the token level indicates that the model often predicts tag 'O' in the job description even when it should be tag 'B-HSkill' or tag 'I-HSkill', resulting in a false positive. On the opposite side, the model often fails to recognize the presence of tag 'B-HSkill', tag 'B-SSkill', or tag 'I-HSkills', even when the tag is present, resulting in a false negative. Fig.7(b) The result of the confusion matrix at the entity level indicates that the model often predicts the tag 'B-Tech' and tag 'I-Skill' in the job description even when it should be tag 'B-HSkill', resulting in a false positive. On the other hand, the model often fails to detect the presence of the tag 'B-Tech', even when the tag is present, resulting in a false negative.



(a) (b)

Fig. 7. The result of the confusion matrix in token lengths: 256. (a) at token level;(b) at entity level.

Fig. 8 illustrates a more comprehensive capture of tags compared to Fig. 7. This discrepancy can be attributed to the distinct token lengths 128 and 256. Despite their differing token lengths, the prediction outcomes are nearly identical. Notably, when considering entity-level predictions, the model with a token length of 256 tends to mistakenly predict the 'O' tag in job descriptions even when it should be tagged 'B-HSkill' or 'I-HSkill'. Similarly, at the entity level, the model often predicts 'B-Tech' and 'I-Skill' tags in job descriptions when they should be tagged as 'B-HSkill' tags. Additionally, the model frequently needs help to identify the presence of the 'B-Tech' tag correctly.



(a) (b)

Fig. 8. The result of the confusion matrix in token lengths: 256. (a) at token level;(b) at entity level.

By examining the confusion matrix values, we can derive various performance metrics that aid in comprehending the performance of IndoBERTSkill. Table 4 shows the results of the NERSkill Model evaluation at the token level.

Table 4. The evaluation results of the NERSkill Model at the token level.

Tag	m=128			m=256		
	P	R	F1	P	R	F1
B-HSkill	59.26%	65.81%	62.36%	64.57%	61.87%	63.19%
B-SSkill	71.96%	87.14%	78.83%	79.73%	83.57%	81.61%
B-Tech	84.29%	84.30%	84.29%	85.38%	84.30%	84.84%
I-HSkill	65.21%	52.28%	58.03%	70.24%	56.09%	62.37%
I-SSkill	68.76%	54.96%	61.09%	80.15%	53.59%	64.24%
I-Tech	73.36%	59.97%	65.99%	80.17%	64.12%	71.25%
O	95.54%	95.37%	95.46%	95.28%	96.80%	96.03%
Average	74.05%	71.40%	72.29%	79.36%	71.48%	74.79%

*m=token lengths, P= Precision, R= Recall, F1= Micro F1-Score

Table 4 indicates that the 'O' tag achieves the highest F1-Score for both token lengths, 128 and 256, with scores of 95,46% and 96,03%, respectively. This is because most tokens in the NERSkill dataset have O tag. Following the 'O' tag, the 'B-Tech' tag exhibits a notable F1-Score of 84.29% and 84.84%. The average F1-Score for token length 128 is 72,29%, while for token length 256 it is 74,79%.

We also evaluated the NERSkill model at the entity level because we want to focus on identifying the skill entities as a whole, including their boundaries and labels. In other words, entities are evaluated based on their spans and the correct label. Table 5 shows the evaluation results of the NERSkill Model at the entity level. The result shows that the 'B-SSkill' tag achieved an F1-score of 96,01% for token length 128 and 96,78% for token length 256, which is higher than any other tag. This phenomenon arises because a majority of 'B-SSkill' tags correspond to single words in the Indonesian Language, like '*ulet*' (diligent), '*teliti*' (precise), and '*cermat*' (careful). Consequently, the recognition of 'B-SSkill' is more consistent than other tags. The 'I-Tech' tag achieves the lowest F1-score of 74.8% due to the frequent confusion between identifying the 'I-Tech' and 'B-Tech' tags, which are often interchanged. For instance, 'framework' is labeled as 'B-Tech' in the term 'springboot framework' but as 'I-Tech' in the phrase 'php framework'. The average F1-score for the NERSkill Model is 87%. These results demonstrate the effective performance of our proposed model in recognising skill entities in job descriptions.

Table 5. The results of the NERSkill Model evaluation at the entity level.

Tag	m=128			m=256		
	P	R	F1	P	R	F1
B-HSkill	85.12%	90.77%	87.85%	86.60%	90.88%	88.69%
B-SSkill	94.72%	97.34%	96.01%	95.42%	98.18%	96.78%
B-Tech	92.01%	92.97%	92.48%	92.57%	94.01%	93.28%
I-HSkill	85.65%	80.42%	82.95%	88.29%	84.27%	86.24%
I-SSkill	94.38%	83.00%	88.32%	95.31%	86.61%	90.75%
I-Tech	80.76%	69.65%	74.80%	86.23%	75.31%	80.40%
Average	88.77%	85.69%	87.07%	90.74%	88.21%	89.36%

*m=token lengths, P= Precision, R= Recall, F1= Micro F1-Score

The results of the evaluation of the NER-Skill model compared to the baseline model on our NERSkill dataset are shown in Table 6. In the token-level evaluation, it can be seen that the NER-Skill

model adapted to IndoBERT achieved the highest micro F1 score of 78.59%. This outcome is attributed to the accurate predictions of the 'O' tag. However, our models achieved the highest F1 score on the skill entity recognition task, outperforming the previous state-of-the-art model by a significant margin. Specifically, NER-Skill model from fin-tuning IndoBERTSkill achieved an F1-score of 87%, 0.5% higher than the previous state-of-the-art model. This result demonstrates the superiority of IndoBERTSkill over the baseline model for recognizing skill entities in job requirements. The ablation study results show that the pre-training process is critical for improving the performance of the models, and training on domain-specific data further improves their performance.

Table 6. Comparison result for NER modeling.

	Pre-trained Language Model	Token level			Entity level		
		P	R	F1	P	R	F1
Baseline	BERT [4]	81.68%	75.03%	77.70%	87.9%	85.9%	86.3%
	IndoBERT [20]	81.65%	76.47%	78.59%	87.4%	85.7%	86.5%
Our Propose	IndoBERTSkill	74.05%	71.40%	72.29%	**88.77%**	85.69%	**87.07%**

* P= Precision, R= Recall, F1= Micro F1-Score

The limitation of this study is its reliance on a single annotated dataset (NERSkill) for evaluation. Although this dataset covers various types of jobs in the IT sector collected from various job portals and includes a mixture of Indonesian and English language formats, it may not cover the entire spectrum of job description structures used in various sectors. This may limit the generalisation of IndoBERTSkill to highly specific or underrepresented domains. As such, its long-term applicability may require continual updating with dynamic corpora and the exploration of newer pre-training objectives such as span prediction or contrastive learning.

The extracted skills from IndoBERTSkill are intended for downstream applications such as automated job matching, curriculum development, career recommendation systems, and labor market analytics. For example, recognized skills can be mapped to individual candidate profiles or academic syllabi to assess alignment with current industry demands. Furthermore, job portals and HR platforms can integrate IndoBERTSkill to enhance recruitment workflows by automatically tagging vacancies with structured skill metadata.

3.3. Error Analysis

Although IndoBERTSkill shows promising results, several challenges arose during development. First, there was high variation in the structure, length, and vocabulary of the job description dataset. Many entries had grammatical inconsistencies and followed non-standard formats, which made the annotation process difficult. In addition, the presence of multilingual content, especially a mixture of Indonesian and English, added complexity to the tokenisation and vocabulary representation processes. IndoBERTSkill consistently improves recognition of domain-specific terms such as "UI/UX", "Node.js", and "content writing", which are often split or misclassified by general models like IndoBERT and mBERT. This improvement is primarily due to the adapted vocabulary and pretraining on job-specific language, which enables the model to treat such tokens as cohesive entities. Although performance has improved, this model still exhibits some recurring errors, such as ambiguous tokens. Words like “framework” can be tagged inconsistently depending on context. For example, the word “framework” has the label B-Tech in the phrase ‘framework springboot’ or the label ‘I-Tech’ in the phrase ‘php framework.’.

4. Conclusion

This paper presents IndoBERTSkill, a domain-specific pre-trained language model built on the BERT architecture, specifically designed for recognizing skill entities in Indonesian job descriptions. IndoBERTSkill demonstrates competitive performance with an F1-score of 87.07%, outperforming existing Indonesian pre-trained models such as IndoBERT and Multilingual BERT. The key contributions of this work lie in (1) the development of a specialized language model tailored to the

structure and vocabulary of Indonesian job descriptions, (2) the creation and annotation of the NERSkill dataset using BIO tagging formats, and (3) a comprehensive evaluation framework including fine-tuning, hyperparameter optimization, and both token-level and entity-level performance analysis. The findings contribute significantly to the broader field of Natural Language Processing (NLP), particularly in the under-researched context of low-resource and mixed-language environments. IndoBERTSkill holds substantial potential for real-world applications such as automated job matching, intelligent recruitment systems, career recommendation engines, and talent management platforms, where accurate extraction of skill entities is essential. Despite its strengths, this study also recognizes several limitations, including the use of a single annotated dataset for evaluation and the challenges posed by mixed-language input. Future work should address these by incorporating more diverse job domains, adapting to emerging language trends, and exploring alternative pre-training strategies to enhance the model's generalizability and robustness.

Declarations

Author contribution. All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.
Funding statement. This research was supported by the Research Directorate of Universitas Gadjah Mada in the Rekognisi Tugas Akhir (RTA) 2021 scheme.
Conflict of interest. The authors declare no conflict of interest.
Additional information. No additional information is available for this paper.

References

[1] N. M. Gardazi, A. Daud, M. K. Malik, A. Bukhari, T. Alsahfi, and B. Alshemaimri, "BERT applications in natural language processing: a review," *Artif. Intell. Rev.*, vol. 58, no. 6, p. 166, Mar. 2025, doi: 10.1007/s10462-025-11162-5.

[2] J. Golec and T. Hachaj, "Ten Natural Language Processing Tasks with Generative Artificial Intelligence," *Appl. Sci.*, vol. 15, no. 16, p. 9057, Aug. 2025, doi: 10.3390/app15169057.

[3] A. Vaswani *et al.*, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, 2017, p. 11. doi: 10.48550/arXiv.1706.03762.

[4] 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*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 4171–4186. doi: 10.18653/v1/N19-1423.

[5] Y. Zhang *et al.*, "A Comprehensive Survey of Scientific Large Language Models and Their Applications in Scientific Discovery," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2024, pp. 8783–8817. doi: 10.18653/v1/2024.emnlp-main.498.

[6] I. D. Mienye, N. Jere, G. Obaido, O. O. Ogunruku, E. Esenogho, and C. Modisane, "Large language models: an overview of foundational architectures, recent trends, and a new taxonomy," *Discov. Appl. Sci.*, vol. 7, no. 9, p. 1027, Sep. 2025, doi: 10.1007/s42452-025-07668-w.

[7] X. Jiang, W. Wang, S. Tian, H. Wang, T. Lookman, and Y. Su, "Applications of natural language processing and large language models in materials discovery," *npj Comput. Mater.*, vol. 11, no. 1, p. 79, Mar. 2025, doi: 10.1038/s41524-025-01554-0.

[8] N. Abdel Samee, M. Alabdulhafith, S. Muhammad Ahmed Hassan Shah, and A. Rizwan, "JusticeAI: A Large Language Models Inspired Collaborative and Cross-Domain Multimodal System for Automatic Judicial Rulings in Smart Courts," *IEEE Access*, vol. 12, pp. 173091–173107, 2024, doi: 10.1109/ACCESS.2024.3491775.

[9] Y. Mao, Q. Liu, and Y. Zhang, "Sentiment analysis methods, applications, and challenges: A systematic literature review," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 36, no. 4, p. 102048, Apr. 2024, doi: 10.1016/j.jksuci.2024.102048.

[10] M. R. R. Rana, A. Nawaz, S. U. Rehman, M. A. Abid, M. Garayevi, and J. Kajanová, "BERT-BiGRU-Senti-GCN: An Advanced NLP Framework for Analyzing Customer Sentiments in E-Commerce," *Int. J. Comput. Intell. Syst.*, vol. 18, no. 1, p. 21, Feb. 2025, doi: 10.1007/s44196-025-00747-1.

[11] D. Herhausen, S. Ludwig, E. Abedin, N. U. Haque, and D. de Jong, "From words to insights: Text analysis in business research," *J. Bus. Res.*, vol. 198, p. 115491, Sep. 2025, doi: 10.1016/j.jbusres.2025.115491.

[12] X. Xu, Z. Li, H. Zhang, and K. Ma, "Named entity recognition for Chinese electronic medical records by integrating knowledge graph and ClinicalBERT," *Front. Artif. Intell.*, vol. 8, p. 1634774, Sep. 2025, doi: 10.3389/frai.2025.1634774.

[13] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: state of the art, current trends and challenges," *Multimed. Tools Appl.*, vol. 82, no. 3, pp. 3713–3744, Jan. 2023, doi: 10.1007/s11042-022-13428-4.

[14] T. Aggarwal, A. Salatino, F. Osborne, and E. Motta, "Large language models for scholarly ontology generation: An extensive analysis in the engineering field," *Inf. Process. Manag.*, vol. 63, no. 1, p. 104262, Jan. 2026, doi: 10.1016/j.ipm.2025.104262.

[15] K. Arai, "Design of On-Premises Version of RAG with AI Agent for Framework Selection Together with Dify and DSL as Well as Ollama for LLM," *Int. J. Adv. Comput. Sci. Appl.*, vol. 15, no. 12, pp. 117–124, Dec. 2024, doi: 10.14569/IJACSA.2024.0151212.

[16] D. Bzdok, A. Thieme, O. Levkovskyy, P. Wren, T. Ray, and S. Reddy, "Data science opportunities of large language models for neuroscience and biomedicine," *Neuron*, vol. 112, no. 5, pp. 698–717, Mar. 2024, doi: 10.1016/j.neuron.2024.01.016.

[17] B. Chen, Z. Zhang, N. Langrené, and S. Zhu, "Unleashing the potential of prompt engineering for large language models," *Patterns*, vol. 6, no. 6, p. 101260, Jun. 2025, doi: 10.1016/j.patter.2025.101260.

[18] A. Miftaroski, R. Zowalla, M. Wiesner, and M. Pobiruchin, "Leveraging Large Language Models to Improve the Readability of German Online Medical Texts: Evaluation Study," *JMIR AI*, vol. 5, no. 1, pp. e77149–e77149, Jan. 2026, doi: 10.2196/77149.

[19] Y. Wang, M. Lin, Q. Hu, S. Bai, Y. Li, and L. Bao, "A domain-specific cross-lingual semantic alignment learning model for low-resource languages," *Neural Networks*, vol. 194, no. Februari, p. 108114, Feb. 2026, doi: 10.1016/j.neunet.2025.108114.

[20] F. Koto, A. Rahimi, J. H. Lau, and T. Baldwin, "IndoLEM and IndoBERT: A Benchmark Dataset and Pre-trained Language Model for Indonesian NLP," in *Proceedings of the 28th International Conference on Computational Linguistics*, Stroudsburg, PA, USA: International Committee on Computational Linguistics, 2020, pp. 757–770. doi: 10.18653/v1/2020.coling-main.66.

[21] B. Wilie *et al.*, "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*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2020, pp. 843–857. doi: 10.18653/v1/2020.aacl-main.85.

[22] F. Koto, J. H. Lau, and T. Baldwin, "IndoBERTweet: A Pretrained Language Model for Indonesian Twitter with Effective Domain-Specific Vocabulary Initialization," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2021, pp. 10660–10668. doi: 10.18653/v1/2021.emnlp-main.833.

[23] S. M. Octafia, "IndoBERT-SupCon: A Supervised Contrastive Learning Model for Analyzing Public Perception on Halal Tourism," *J. Appl. Data Sci.*, vol. 7, no. 1, pp. 218–231, Jan. 2026, doi: 10.47738/jads.v7i1.1045.

[24] N. C. Mei, S. Tiun, and G. Sastria, "Multi-Label Aspect-Sentiment Classification on Indonesian Cosmetic Product Reviews with IndoBERT Model," *Int. J. Adv. Comput. Sci. Appl.*, vol. 15, no. 11, pp. 712–720, Nov. 2024, doi: 10.14569/IJACSA.2024.0151168.

[25] A. Riyadi, M. Kovacs, U. Serdült, and V. Kryssanov, "IndoGovBERT: A Domain-Specific Language Model for Processing Indonesian Government SDG Documents," *Big Data Cogn. Comput.*, vol. 8, no. 11, p. 153, Nov. 2024, doi: 10.3390/bdcc8110153.

[26] S. O. Khairunnisa, Z. Chen, and M. Komachi, "Improving Domain-Specific NER in the Indonesian Language Through Domain Transfer and Data Augmentation," *J. Adv. Comput. Intell. Intell. Informatics*, vol. 28, no. 6, pp. 1299–1312, Nov. 2024, doi: 10.20965/jaciii.2024.p1299.

[27] E. Yulianti, N. Bhary, J. Abdurrohman, F. W. Dwitilas, E. Q. Nuranti, and H. S. Husin, "Named entity recognition on Indonesian legal documents: a dataset and study using transformer-based models," *Int. J. Electr. Comput. Eng.*, vol. 14, no. 5, p. 5489, Oct. 2024, doi: 10.11591/ijece.v14i5.pp5489-5501.

[28] M. Laugiwa and E. Yulianti, "Question Answering through Transfer Learning on Closed-Domain Educational Websites," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 9, no. 1, pp. 104–110, Feb. 2025, doi: 10.29207/resti.v9i1.6163.

[29] S. Thavasi and T. Revathi, "A personalized machine learning–based system to evaluate students' skillset and analyze the gap between academia and industry for engineering students," *Kybernetes*, vol. 54, no. 9, pp. 4850–4865, Oct. 2025, doi: 10.1108/K-01-2024-0053.

[30] N. P, M. V, K. K. Ram, P. Vamsi, V. V. Vardhan Reddy, and D. Sridhar, "Sentimental Analysis of Job Classification using Machine Learning Algorithms," in *2023 8th International Conference on Communication and Electronics Systems (ICCES)*, IEEE, Jun. 2023, pp. 1279–1284. doi: 10.1109/ICCES57224.2023.10192717.

[31] M. A. Berawi, M. Sari, N. H. A. Amiri, suci indah Susilowati, S. R. Utami, and A. Kulachinskaya, "Developing a Machine Learning Model to Improve the Accuracy of Owner Estimate Cost in the Capital Expenditure Procurement Process," *Int. J. Technol.*, vol. 16, no. 4, p. 1179, Jul. 2025, doi: 10.14716/ijtech.v16i4.7409.

[32] K. Jian and M. A. A. D. Dizon, "Analysis of employment trends and prediction model for Chinese college graduates based on natural language processing," *Edelweiss Appl. Sci. Technol.*, vol. 9, no. 4, pp. 1145–1159, Apr. 2025, doi: 10.55214/25768484.v9i4.6188.

[33] W. Fang, Y. Xu, T. Zhang, Y. Wang, and L. Zheng, "Towards large language model for cognitive industrial mixed reality: A survey," *Adv. Eng. Informatics*, vol. 71, no. April, p. 104276, Apr. 2026, doi: 10.1016/j.aei.2025.104276.

[34] S. Fareri, R. Apreda, V. Mulas, and R. Alonso, "The worker profiler: Assessing the digital skill gaps for enhancing energy efficiency in manufacturing," *Technol. Forecast. Soc. Change*, vol. 196, no. November, p. 122844, Nov. 2023, doi: 10.1016/j.techfore.2023.122844.

[35] X. Q. Ong and K. Hui Lim, "SkillRec: A Data-Driven Approach to Job Skill Recommendation for Career Insights," in *2023 15th International Conference on Computer and Automation Engineering (ICCAE)*, IEEE, Mar. 2023, pp. 40–44. doi: 10.1109/ICCAE56788.2023.10111438.

[36] P. Achananuparp, Y. Xu, Y. Lu, X. J. S. Ashok, and E.-P. Lim, "Leveraging large language models for career mobility analysis: a study of gender, race, and job change using U.S. online resume profiles," *EPJ Data Sci.*, vol. 15, no. 1, p. 4, Jan. 2026, doi: 10.1140/epjds/s13688-025-00607-0.

[37] R. Alonso, D. Dessí, A. Meloni, and D. Reforgiato Recupero, "A novel approach for job matching and skill recommendation using transformers and the O*NET database," *Big Data Res.*, vol. 39, no. February, p. 100509, Feb. 2025, doi: 10.1016/j.bdr.2025.100509.

[38] N. Matkin *et al.*, "Correction to: Comparative Analysis of Encoder-Based NER and Large Language Models for Skill Extraction from Russian Job Vacancies," in *Communications in Computer and Information Science*, vol. 2364 CCIS, Springer Science and Business Media Deutschland GmbH, 2026, pp. C1–C1. doi: 10.1007/978-3-031-97019-1_13.

[39] R. Wang, Q. Chen, Y. Wang, L. Xiong, and B. Shen, "JobViz: Skill-driven visual exploration of job advertisements," *Vis. Informatics*, vol. 8, no. 3, pp. 18–28, Sep. 2024, doi: 10.1016/j.visinf.2024.07.001.

[40] K. Nguyen, M. Zhang, S. Montariol, and A. Bosselut, "Rethinking Skill Extraction in the Job Market Domain using Large Language Models," in *Proceedings of the First Workshop on Natural Language Processing for Human Resources (NLP4HR 2024)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2024, pp. 27–42. doi: 10.18653/v1/2024.nlp4hr-1.3.

[41] M. Hasan Nizami, A. Uddin, M. T. Salani, A. Saeed, F. Alvi, and A. Samad, "Enhancing Human Capital Management: AI Techniques for Candidate Matching and Skill Extraction Notebook for the TalentCLEF Lab at CLEF 2025," 2025, p. 11. Accessed: May 24, 2026. [Online]. Available: https://www.semanticscholar.org/paper/Enhancing-Human-Capital-Management%3A-AI-Techniques-Nizami-Uddin/28430020b7ad812bc59ff8555b036edee1bbd208.

[42] M. Lukauskas, V. Šarkauskaitė, V. Pilinkienė, A. Stundžienė, A. Grybauskas, and J. Bruneckienė, "Enhancing Skills Demand Understanding through Job Ad Segmentation Using NLP and Clustering Techniques," *Appl. Sci.*, vol. 13, no. 10, p. 6119, May 2023, doi: 10.3390/app13106119.

[43] M. N. Tentua, Suprapto, and Afiahayati, "NERSkill.Id: Annotated dataset of Indonesian's skill entity recognition," *Data Br.*, vol. 53, no. April, p. 110192, Apr. 2024, doi: 10.1016/j.dib.2024.110192.

[44] H. Kavas, M. Serra-Vidal, and L. Wanner, "Multilingual Skill Extraction for Job Vacancy–Job Seeker Matching in Knowledge Graphs," 2025. Accessed: May 24, 2026. [Online]. Available: https://aclanthology.org/2025.genaik-1.15/.