Jurnal Sistem Informasi Bisnis 04(2024)
Copyright ©2024, JSINBIS, p-ISSN: 2502-2377, e-ISSN: 2088-3587
On-line: http://ejournal.undip.ac.id/index.php/jsinbis/article/view/65126

A Novel Fusion of Machine Learning Methods for Enhancing Named Entity Recognition in Indonesian Language Text

Widyawan [a,*], Bayu Prasetiyo Utomo[b], Muhammad Nur Rizal[a]

[a] Department of Electrical Engineering and Information Technology, Universitas Gadjah Mada Yogyakarta
[b] Directorate of Information Technology, Universitas Gadjah Mada Yogyakarta

Received: July 16th, 2024; Accepted: August 07 th, 2024
DOI: 10.21456/vol14iss4pp311-320

Abstract

One of the important implementations in machine learning is Named Entity Recognition (NER), which is used to process text and extract entities such as people, organizations, laws, religions, and locations. NER for the Indonesian language still faces significant challenges due to the lack of high-quality labelled datasets, which limits the development of more advanced models. To address this issue, we utilized several pre-trained BERT models (bert-base-uncased, indobenchmark/indobert-base-p1, indolem/indobert-base-uncased) and datasets (NERGRIT-IndoNLU, NERGRIT-Corpus, NERUGM, and NERUI). This study proposes a novel fusion approach by integrating deep learning architectures such as CNN, Bi-LSTM, Bi-GRU, and CRF to detect 19 entities. This approach enhances BERT's sequence modelling and feature extraction capabilities, while CRF improves entity prediction by enforcing global word-sequence constraints. Experimental results demonstrate that the fusion approach outperforms previous methods. On the bert-base-uncased dataset, accuracy reached 94.75%, while indobenchmark/indobert-base-p1 achieved 95.75%, and indolem/indobert-base-uncased achieved 95.85%. This study emphasizes the effectiveness of combining deep learning architectures with pre-trained transformers to improve NER performance in the Indonesian language. The proposed methodology offers significant advancements in entity extraction for languages with limited datasets, such as Indonesian.

Keywords: NER; BERT; Pre-training, Machine Learning

Abstrak

Salah satu implementasi penting dalam pembelajaran mesin adalah *Named Entity Recognition* (NER) yang digunakan untuk memproses teks dan mengekstrak entitas seperti orang, organisasi, hukum, agama, dan lokasi. NER untuk bahasa Indonesia masih menghadapi tantangan besar akibat kurangnya dataset berlabel yang berkualitas, yang membatasi pengembangan model yang lebih maju. Untuk mengatasi masalah ini, kami menggunakan beberapa model pra-pelatihan BERT (bert-base-uncased, indobenchmark/indobert-base-p1, indolem/indobert-base-uncased) dan dataset (NERGRIT-IndoNLU, NERGRIT-Corpus, NERUGM, serta NERUI. Studi ini mengusulkan pendekatan fusi baru dengan mengintegrasikan arsitektur deep learning seperti CNN, Bi-LSTM, Bi-GRU, dan CRF untuk mendeteksi 19 entitas. Pendekatan ini meningkatkan kemampuan pemodelan urutan dan ekstraksi fitur dari BERT, sementara CRF memperbaiki prediksi entitas dengan mengatur urutan kata secara keseluruhan. Hasil eksperimen menunjukkan bahwa pendekatan kami mampu mengungguli metode sebelumnya. Pada dataset bert-base-uncased, akurasi mencapai 94,75%, sedangkan indobenchmark/indobert-base-p1 mencapai 95,75% dan indolem/indobert-base-uncased mencapai 95,85%. Studi ini menekankan efektivitas kombinasi arsitektur deep learning dengan transformer pra-pelatihan untuk meningkatkan kinerja NER dalam bahasa Indonesia. Metodologi yang diusulkan menawarkan kemajuan signifikan dalam ekstraksi entitas pada bahasa dengan dataset terbatas, seperti bahasa Indonesia.

Kata kunci: NER; BERT; Pre-training; Pembelajaran Mesin

1. Introduction

The Internet users in Indonesia have an impact on the number of sites accessed, such as news portals and social media. In 2024 Twitter users in Indonesia were recorded as occupying the 4th largest position with a total of 24.85 million (Statista, 2024). Interaction between users on social media and the large number of daily news stories published online have resulted in availability of data in the form of text that becomes very large (Nasichuddin *et al.*, 2018).

This large amount of data can be analyzed to extract entities in the text, such as locations, organizations, health terms, finances, and people's names. One of the techniques to extract location entity is by matching text with a database containing gazetteer geographic information (Middleton *et al.*, 2018). The gazetteer-only approach has a drawback if the text found in the gazetteer refers to an entity rather than a location. Some studies combine several location extractions approaches to obtain better location recognition (Middleton *et al.*, 2018). Entity recognition from text can also be used for further

*) Corresponding author: widyawan@ugm.ac.id

analysis processes such as location prediction (Utomo *et al.*, 2018). So that better extraction will improve the results of the information provided or analysis further.

NER (Named Entity Recognition) is an application of Natural Language Processing (NLP) to extract entities from text that can gain benefits from abundant training data obtained from the Internet. NER can be carried out with various approaches. both rule-based (Eftimov *et al.*, 2017; Sinta & Sanjaya ER, 2021) or with ML. The rule-based approach classifies text using a set of linguistic rules. Meanwhile, the approach using ML uses probabilistic and statistics.

The ML researchers keep improving the performance solving NER. Deep learning, a sub-field of ML, excels by adeptly grasping intricate and abstract textual features. It gives better performance compared to shallow learning or traditional ML (Nasichuddin *et al.*, 2018). Several studies have been carried out using deep learning to solve NER such as bidirectional long short-term memory (Bi-LSTM) (Azzahra *et al.*, 2020; Nuranti & Yulianti, 2020; Sukardi *et al.*, 2020). Convolutional Neural Network (CNN) (Azzahra *et al.*, 2020; Nuranti & Yulianti, 2020), conditional random field (CRF) (Azzahra *et al.*, 2020; Situmeang, 2022) and bidirectional encoder from transformers (BERT) (Koto *et al.*, 2020; Wilie *et al.*, 2020). Some of the researchers combine several deep learning algorithms to improve the performance of NER (Azzahra *et al.*, 2020; Koto *et al.*, 2020).

Solving NER in specific language such as Indonesian. is still limited compared to English. Many studies related to English NER already have good performance, however Indonesian NER still needs to be improved (Alfina *et al.*, 2016, 2017; Ma & Hovy, 2016). Even though Indonesian is known as the fourth most widely used language on the internet (Wilie *et al.*, 2020), there are not many pre-trained language models in Indonesian. Several studies that have been carried out publish datasets and their hyperparameters to serve as a reference. The purpose is to enrich Indonesian NLP research and provide benchmarks for the models and an opportunity to develop libraries or models based on Indonesian language datasets (Budi & Suryono, 2023). Several research have been carried out in the case of NER where a pre-trained model using Indonesian language gives better results (Wilie *et al.*, 2020).

BERT is an example of a derivative of the Transformers model that Google uses in its search engine. Transformers models have become well known in the NLP process in recent years. BERT can describe the context of words or sentences. BERT is trained by completing NLP tasks such as Next Sentence Prediction (NSP) and Masked Language Modelling (MLM) (Devlin *et al.*, 2018). BERT can also perform specific NLP processing such as NER. However, the accuracy is not as good as if the model is fine-tuned for a particular task. The fine-tuning process can take form of adding layers or configuring hyperparameters when training with a labelled dataset. Employing a pre-trained model along with a combination of several techniques can enhance the accuracy of Indonesian NER. This study introduces a novel model for Named Entity Recognition in Indonesian text. The proposed model uses a fusion of BERT, CNN, Bi-LSTM, Bi-GRU and CRF. BERT, which had been previously trained with an Indonesian language corpus, was fine-tuned by combining CNN, Bi-LSTM, Bi-GRU and CRF. BERT is used as a pre-training model and comparisons will be made later.

The remainder of this paper is structured in the following manner: Section 2 explores related works in Named Entity Recognition (NER). Section 3 describes the methodology applied and introduces a novel model for Indonesian NER. Finally, Section 4 presents the outcomes and analysis, highlighting the strengths and weaknesses of the evaluated models.

2. Related Works

Study in Putra (2021) carried out location extraction by adopting NeuroNER method with Bi-LSTM and CRF. NeuroNER is a modification of the previous NER technique which uses the CRF algorithm by adding the recurrent neural network (RNN) algorithm to the NER model. NER results produce the average of all entities using NeuroNER precision 96.56%, recall 95.89%, and F1 96.21%.

Research in Wilie *et al.*, (2020) carried out 12 NLP tasks to provide a benchmark for Indonesian NLP regarding the pre-training IndoNLU model proposed. IndoNLU is BERT which is retrained with an Indonesian language corpus (OSCAR, CoNLLu Common Crawl, OpenSubtitles, Wikipedia, Dump, Wikipedia CoNLLu, Twitter Crawl, Twitter UI, OPUS JW300, Tempo, Kompas, TED, Parallel Corpus, TALPCo and Frog Story telling). It was reported F1 score of 67.42% for the NERGrit-IndoNLU dataset.

Koto *et al.*, (2020) created Pre-training BERT with an Indonesian language corpus (Indonesian Wikipedia, Kompas, Tempo, Liputan 6, Indonesian Web Corpus). As a benchmark, they carried out three NLP tasks divided into three categories (morpho-syntax/sequence labeling, semantics, and discourse coherence). They carried out Post Tagging and NER tasks for sequence labeling. Semantics tasks with sentiment analysis and summarization. Lastly, discourse coherence with the tasks of next tweet prediction and tweet ordering. Hyperparameters were differentiated for each task. Specifically for Post Tagging and NER with LR 5e-5, epoch of 100 with early stopping (patience = 5). The NER task with sequence of 512 produces F1 micro 74.9 % NER UGM and 90.1% NERUI.

Study in Azzahra *et al.*, (2020) conducted Indonesian NER from an unstructured dataset using a deep learning approach. The algorithms used are LSTM, Bi-LSTM, GRU, Bi-GRU, and CNN. The dataset used by NERGrit-Corpus is processed and divided into four types. The first is with no processing, the second type of dataset is processed with lowercase and without punctuation, the third type is with lowercase and punctuation, and the last is lowercase with a cleaned dataset. The lowercase process is carried out by making the words in the token dataset lowercase. The punctuation process carries out the process of removing punctuation marks, thereby reducing unnecessary tokens. The cleansing process removes lowercase words and labels without punctuation which only have the label O (not an entity). The F1 results show Bi-GRU with the highest score of 71.04% for the first dataset without processing, for the second and so on it produces 70.61%, 68.12%, 67.45%. The result shows that more complex DL algorithms do not give better performance than the simple once.

Study in Nuranti and Yulianti (2020) looked for the effectiveness of using deep learning to recognize entities in Indonesian language trial results documents. The algorithms used are CNN, LSTM, LSTM_CRF, and Bi-LSTM. The research also uses SVM and CRF to see comparisons as well as deep learning and ML approaches. The results show that the combination of Bi-LSTM and CRF produces the best F1 results of 83%. Other F1 algorithms are SVMfull 7%, SVMRemove_o_label 10%, CRF 42%, CNN 71%, LSTM 78%, Bi-LSTM 81%, LSTM_CRF 80%, Bi-LSTM_CRF 83%. Using Bi-LSTM alone produces good results but combining it with CRF can produce better understanding. The result shows that combined algorithms give better performance. However, the performance does not give a good F1 result if we compare it with the BERT Pre-Training model.

Table 1. summarizes the result of previous works. Deep Learning outperforms machine learning algorithms based on (Nuranti & Yulianti, 2020). The researchers Putra (2021) combine the DL algorithm to improve the performance. But we also considering, the model does not get complex, whereas the Bi-GRU can outperform LSTM and Bi-LSTM (Azzahra *et al.*, 2020). The others (Wilie *et al.*, 2020) , Koto *et al.*, (2020) using BERT Pre-Training to improve the performance solving Indonesian NER. The new researchers, keep improved the better performance to solve the Indonesia NER. Therefore, this study proposed the model that solve the Indonesia NER based on the literature review of previous study. We fusion the algorithms already purposed to solve NER with the Pre-Training in Indonesian to find new purposing model.

3. Methodology

The creation of a model for the Indonesian Named Entity Recognition (NER) task commences with the aggregation of a dataset intended for machine learning (ML) training. The dataset chosen is derived from a systematic literature review, providing a basis for comparing the proposed model against those established in prior studies.

Model evaluation is conducted using measures of accuracy, F1 macro, and F1 micro scores. The optimal average values from multiple trial datasets are benchmarked to ascertain the consistently superior model, which is then selected for implementation in location entity extraction. The holdout method is employed, where training data is used to develop the model, validation data to assess its accuracy, and test data to verify the validation of outcomes.

3.1. Datasets

Indonesia itself is in the top 4 with the largest population in the world (Devi *et al.*, 2016). Conversations using Indonesian expressed in cyberspace were the fourth largest in 2020 (Wilie *et al.*, 2020). Based on the SLRs already conducted, 1 out of 36 studies used an Indonesian text as dataset (Simanjuntak *et al.*, 2022).

The use of private datasets is another cause of the lack of benchmarks. The availability of private datasets generally requires permission first if you want to use them and access to the dataset is not even permitted. This makes it difficult to compare with the proposed model. The following is a list of available datasets. NERGrit IndoNLU (Genta Indra Winata, 2023): A collection of formal sentences, from wiki data sources. with a total of 2090 data. This dataset has been processed by IndoNLU compared with the source dataset NERGrit Corpus. NERGrit Corpus (Inovasi Teknologi, 2023): A collection of formal sentences with a total of 435.437 data. NER UI (Koto *et al.*, 2023): Collection of sentences from news data sources. with a total of 2.125 data. NER UGM (Koto *et al.*, 2023): Collection of sentences from news data sources, with a total of 2.243 data.

Datasets were used for training to detect 19 entities: cardinal, date, event, facility, geopolitical, language, law, location, money, ordinal, organization, percentage, person, political organization, product, quantity, religion, time and work of art.

Table 1. Summary of previous works

Research	Dataset	Methods	F1 Avg
Putra (2021)	Tweet crawling (private)	NeuroNER (Bi-LSTM_CRF)	96.21% (micro)
Wilie et al., (2020)	NERGrit-IndoNLU	BERT IndoNLU	67.42% (macro)
Koto et al., (2020)	NERUGM NERUI	BERT IndoLEM	74.9%. 90.1% (micro)
Azzahra et al., (2020)	NERGrit-Corpus	LSTM. Bi-LSTM. GRU. Bi-GRU. CNN	61.65%. 70.41%. 63.38%. 71.04%. 62.92% (micro)
Nuranti and Yulianti, (2020)	Legal Entity (private)	SVMfull. SVMRemove_o_label. CRF. CNN. LSTM. Bi-LSTM. LSTM_CRF. Bi-LSTM_CRF	7%. 10%. 42%. 71%. 78%. 81%. 80%. 83% (macro)

3.2. Data preprocessing

In this study preprocessing is realized through tokenizing, text to sequence, sentence segmentation and coding BERT input. Tokenizing involves breaking sentences into tokens that may represent a word or a sub word. In BERT, before being entered into the training process, the text is broken down into sub words that can be recognized by the BERT model. Text to sequence comprises organizing the dataset's words into a sequential index of words, each paired with its respective labels or attributes within the dataset. Sentence segmentation groups input words and labels in the dataset into sentences.

This process involves coding text into input for the BERT embedding layer in the form of numeric. The first stage is tokenization which has been carried out previously, adding CLS tokens at the beginning and SEP at the end. This process also requires dividing words into sub words with the marker "##" for words that are not found in the dictionary.

In this research, the coding of BERT input follow example given in IndoNLU module. The text under analysis comprises an array of words or sub words within a single sentence. Numerical data is transformed into vector values, also known as word vectors, which enables the machine to process the information for subsequent tasks like Named Entity Recognition (NER). The lexicon utilized for this conversion is derived from the pre-training vocabulary of the BERT model.

3.3. Hyperparameter

The hyperparameters used are based on research benchmarks conducted by IndoNLU. This study chooses the IndoNLU, because one of the hyperparameter they proposed to compare with, is suitable with the limitation of resources and times. The testing process involves carrying out a training process using the Adam optimization algorithm. The learning-rate value is 0.00004 (4e-5). Batch size 16, with repetition 25 times. To limit overfitting, Early Stopping Patience of 12 is set. This means that if the loss value during validation does not improve after 12 repetitions, the training process will be stopped. Apart from that, Max Norm 10 was also set to limit the model so that it does not overfit due to too large a number in the neural network weighing in the ML layers used. Seed 42 is set to produce randomization values that are always the same if the model is retrained. This research also limits the Sequence for fine-tuning on BERT to 512. The following Table 2 shows the hyper parameters used in this study.

Table 2. Hyperparameter

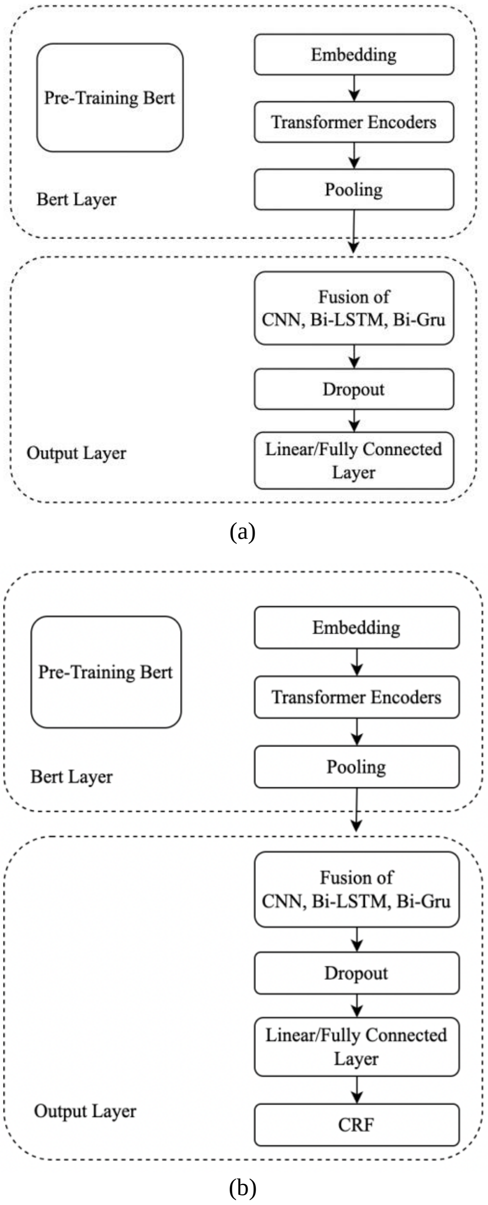
Hyperparameter	Value
Batch Size	16
Learning Rate	4e-5
Optimizer	Adam
Epoch	25
Early Stopping Patience	12
Max Norm	10
Seed	42
Max Sequence	512

3.4. Model Architecture

Enhancement of NER for Indonesian text is performed with fusion of 5 Machine Learning methods, namely BERT, CNN, Bi-LSTM, Bi-GRU and CRF. These algorithms will be fused and evaluated as follow: Bi-LSTM, Bi-GRU, CNN, CRF, CNN_Bi-LSTM, CNN_Bi-GRU, CNN_CRF, Bi-LSTM_CRF, Bi-GRU_CRF, Bi-LSTM_Bi-GRU, Bi-LSTM_Bi-GRU_CRF, CNN_Bi-LSTM_CRF, CNN_Bi-GRU_CRF, CNN_Bi-LSTM_Bi-GRU, and CNN_Bi-LSTM_Bi-GRU_CRF.

The fusion model discussed in this study is illustrated in Figure 1. This research assesses two variations of the model: one incorporating a

Conditional Random Field (CRF) and the other without it. Both models are composed of two primary layers: a pre-trained BERT layer and an output layer. Further specifics about these layers are provided in the subsequent sections as follow.



(a)

(b)

Figure 1. (a) Model without CRF (b) Model with CRF

3.4.1. BERT Layer

In this research, the BERT architecture utilizes an Input Embedding/Token Embedding Layer, which converts sub words into vector representations. This layer processes the input data that has already undergone embedding/tokenization. For the sake of fair comparison with previous studies, the pre-training employed an uncased model, specifically the 'bert-base-uncased', 'indobenchmark/indobert-base-p1', and 'indolem/indobert-base-uncased' models.

Transformers Encoders Layer. the transformers layer used depends on the Pre-training model used for the tuning process. In the base model the total number of layers is 12 and in the large model there are 24. Pooling Layer. the pooling layer used also depends on the Pre-training model that will be used. In the base model the total number of layers is 768 and in the large model there are 1024.

Output/Task Specification/Classification layer. this layer begins by averaging the tokens from the final layer in pooling. This is because at the embedding stage there are words that are tokenized per sub-word (according to the vocabulary of the Pre-training model). So, the model gets an understanding of the embeddings per word in the training labels. The next stage continues to dropout layer 0.1 (Devlin *et al.*, 2018) to prevent overfitting. The final stage is connected directly to the fully-connected-layer with several labels that will be classified according to each dataset. The evaluation results for each batch use the SoftMax loss function.

3.4.2. Output Layer

The output layer is mainly a fusion of various machine learning methods including CNN, Bi-LSTM, Bi-GRU and CRF. Each method will be discussed in detail in the following sections.

A. CNN

The research presents a CNN architecture that incorporates one-dimensional CNN layers. These layers are specifically chosen to handle text data, which inherently requires only one dimension. The convolution layers are designed with padding, utilizing a kernel size of 3 and a stride of 1, and are directly connected to a fully connected layer. The model uniquely integrates a CNN layer which precedes the token averaging step in the process.

B. Bi-LSTM

In this study, the Bi-LSTM method is configured with an input layer size matching BERT's output layer of 768. The model consists of two layers with a hidden size of 384. The Bi-LSTM layer processing follows the token averaging step, ensuring clarity in the model's comprehension of word context, and preventing any potential confusion or mix-up.

C. Bi-GRU

The Bi-GRU approach utilized in this study features an input layer of size 768, corresponding to the output layer of BERT. It includes a total of two layers, each with 384 hidden units. The Bi-GRU layer is applied after the token averaging step to preserve the model’s ability to discern the context of words and sub-words.

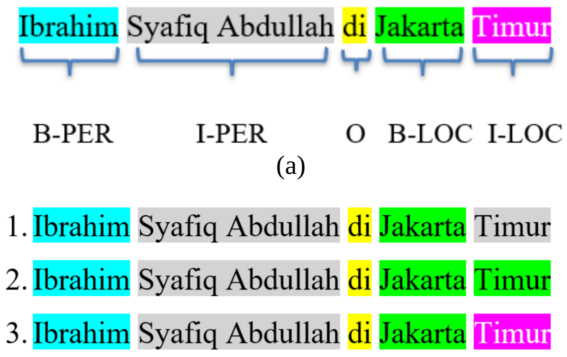
D. CRF

The Conditional Random Field (CRF) taking input from fully connected layer that corresponds to the number of labels present in each dataset. The CRF's role is to decode the tensor input it receives into a predicted sequence of labels. For models that do not incorporate a CRF layer, the approach involves identifying the index or the maximum value within the tensor along the last dimension, which is then interpreted as the label prediction made by the model.

4. Result Analysis

This research was carried out by comparing and analyzing the effect of the Pre-training model and a combination of several ML methods used in this research. For evaluation metrics, the study used F1-Score to evaluate model performance. Evaluation of model performance based on CONLL (Sang *et al.*, 2003) i.e., per entity. Figure 2 (a) shows some actual labels and Figure 2 (b) predicted labels in the dataset. In Prediction 1, the entity "Jakarta Timur" is incorrect, because I-LOC produces I-PER prediction. Moreover, the labels for prediction 2 and prediction 3 are correct to predict the entity LOC (location).

In this study, we did not use a strict approach where the IOB (Inside-Outside-Beginning) tag level had to be the same, entity LOC must predict tag B-LOC and I-LOC, following several studies from (Koto *et al.*, 2020; Wilie *et al.*, 2020). If a strict evaluation mechanism is used, the correct entity evaluation is only prediction 3. Figure 2 (b) shows how the evaluation pays attention to the correctness of the resulting entity, not to the strictness of each tag. F1-Score (Takahashi *et al.*, 2022) is calculated based on the harmonic average of the precision and recall of each entity as shown in Eq. 1 on a micro basis and macro in Eq. 2. F1 micro calculate metrics globally instead of macro calculate metrics for each label. So, it's suitable to use F1 macro if the datasets are imbalanced, where all classes are equally important. F1 Micro treat overall performance regardless of the balanced class, so the majority class would have a bigger impact toward performance result.


Ibrahim
Syafiq Abdullah
di
Jakarta
Timur
B-PER
I-PER
O
B-LOC
I-LOC
1. Ibrahim Syafiq Abdullah di Jakarta Timur
2. Ibrahim Syafiq Abdullah di Jakarta Timur
3. Ibrahim Syafiq Abdullah di Jakarta Timur

(a)

(b)

Figure 2. (a) Actual label. (b) Prediction

$$MiPrecision = \frac{\sum_{i=1}^{r} tp_i}{\sum_{i=1}^{r}(tp_i + fp_i)} \tag{1}$$

$$MiRecall = \frac{\sum_{i=1}^{r} tp_i}{\sum_{i=1}^{r}(tp_i + fn_i)}$$

$$MiF1 = 2 * \frac{MiPrecision * MiRecall}{MiPrecision + MiRecall}$$

$$MaF1 = \frac{1}{r}\sum_{1=1}^{r} F1\,i \tag{2}$$

In the case *r* is sum of class, with *i* refers to 1,2, 3, and so on.

4.1. Bert-base-uncased model performance

In Table 3. the highest average accuracy on the dataset with the models mentioned is 94.75% (CNN. Bi-GRU_CRF). For F1 micro and macro-CNN was highest with 77.5% and 73.25%.

Improvements observed in some fusion methods, demonstrate the model's understanding of complex representations from the data. However, it is evident that some fusions do not yield significant improvements. This can happen if a complex model occurs overfitting, when the model is unable to generalize well on data that has never been trained before. Limited training data is also a factor because complex models require a lot of training to optimize the model parameters properly. The mismatch of interactions between layers is also a consideration for future fusions in machine learning. In this combination of models during training, the addition of the Bi-LSTM layer reduces the average performance of the model.

4.2. Indobenchmark/ indobert-base-p1 model performance

The highest average accuracy on the dataset with the models mentioned is 95.75% in Tuning Fully Connected, CNN, CRF, CNN_CRF methods. For F1 micro, the highest is 83% on CNN_CRF, and the highest macro is on the Fully Connected Tuning model at 79.25%. When using pre-training with the Indonesian language corpus, there was no significant improvement in average performance; only the F1 micro score showed an increase. Pre-training with indobenchmark/indobert-base-p1 can represent the Indonesian dataset better, but the fusions of machine learning with pre-training does not show significant performance improvements.

4.3. Indolem/ indobert-base-uncased model performance

The research shows that the average performance of pre-training indolem/indobert-base-cased, as presented in Table 3, with several datasets produced the highest accuracy of 95.85% (Tuning Fully Connected, CNN_CRF). The highest micro F1 score was achieved in Tuning Fully Connected at 83.85%, and the highest macro F1 score was in CRF, Bi-GRU_CRF at 78.2%.

In this pre-training, the fusion with machine learning did not significantly improve performance. This could be because the model's understanding in BERT was already strong (as previous testing showed it produced the best F1 micro score). Other optimizations may need to be explored in future research, such as hyperparameter tuning, variations in the number of dropouts, and the use of activation functions like SoftMax and ReLU.

Table 3. Pretraining model performance

Method	Bert-base-uncased			Indobenchmark/indobert-base-p1			Indolem/indobert-base-uncased		
	Accuracy	$F1_{micro}$	$F1_{macro}$	Accuracy	$F1_{micro}$	$F1_{macro}$	Accuracy	$F1_{micro}$	$F1_{macro}$
Tuning fully connected	95.03%	79.03%	74.36%	95.75%	82.75%	79.25%	95.85%	83.85%	77.85%
Bi-LSTM	90.88%	55.60%	47.61%	85.75%	32%	26.25%	94.65%	73.80%	64.35%
Bi-GRU	95%	77.21%	71.18%	94.75%	74.75%	68.75%	95.75%	82.65%	76.05%
CNN	95.43%	80.93%	76.16%	95.75%	82.25%	78.25%	95.80%	83.05%	77%
CRF	95.35%	80.51%	75.90%	**95.75%**	82.75%	79%	95.8%	83.05%	**78.20%**
CNN_Bi-LSTM	88.23%	43.65%	34.71%	86.50%	36.25%	27%	86.95%	41.95%	34.65%
CNN_Bi-GRU	94.48%	74.66%	67.41%	93.75%	69%	60%	95.20%	78.50%	71%
CNN_CRF	94.45%	74.38%	69.03%	**95.75%**	**83%**	78.25%	**95.85%**	83.15%	76.60%
Bi-LSTM_CRF	89.95%	50.75%	40.26%	84.75%	30.25%	23%	92.60%	61%	46.80%
Bi-GRU_CRF	95.26%	80.21%	75.31%	95.25%	81%	76.75%	95.80%	83.65%	**78.20%**
Bi-LSTM_Bi-GRU	92.98%	68.80%	63.51%	90.25%	59.25%	55.75%	95.20%	78.40%	72.05%
Bi-LSTM_Bi-GRU_CRF	90.88%	55.68%	50.68%	87.25%	39.75%	37.50%	94.90%	77.05%	69.80%
CNN_Bi-LSTM_CRF	87.08%	37.68%	31.10%	86.75%	36.25%	32%	86%	38.30%	33.30%
CNN_Bi-GRU_CRF	93.41%	70.7%	64.76%	90.50%	59%	54%	95.25%	78.60%	72.05%
CNN_Bi-LSTM_Bi-GRU	89.76%	48.86%	40.65%	86.25%	34.25%	25.75%	90.80%	49.10%	39.20%
CNN_Bi-LSTM_Bi-GRU_CRF	90.08%	51.51%	43.33%	85.50%	31%	24.25%	94.25%	72.05%	61.75%

4.4. Average model performance

In the trials carried out, Table 4 shows that the average performance of several pre-training and BERT model datasets fused with CNN produced an accuracy of 95.43%, an F1 micro score of 80.93%, and the best macro F1 score of 76.16%. The machine learning fusion in the tests carried out can improve pre-training, which previously had poor performance. However, the research results showed that there was no significant increase in performance with pre-training.

Table 4. Average model performance

Method	Accuracy	$F1_{micro}$	$F1_{macro}$
Tuning fully connected	95.03%	79.03%	74.36%
Bi-LSTM	90.88%	55.60%	47.61%
Bi-GRU	95%	77.21%	71.18%
CNN	**95.43%**	**80.93%**	**76.16%**
CRF	95.35%	80.51%	75.90%
CNN_Bi-LSTM	88.23%	43.65%	34.71%
CNN_Bi-GRU	94.48%	74.66%	67.41%
CNN_CRF	94.45%	74.38%	69.03%
Bi-LSTM_CRF	89.95%	50.75%	40.26%
Bi-GRU_CRF	95.26%	80.21%	75.31%
Bi-LSTM_Bi-GRU	92.98%	68.80%	63.51%
Bi-LSTM_Bi-GRU_CRF	90.88%	55.68%	50.68%
CNN_Bi-LSTM_CRF	87.08%	37.68%	31.10%
CNN_Bi-GRU_CRF	93.41%	70.7%	64.76%
CNN_Bi-LSTM_Bi-GRU	89.76%	48.86%	40.65%
CNN_Bi-LSTM_Bi-GRU_CRF	90.08%	51.51%	43.33%

4.5. Comparison with prior work

The research results were carried out further by comparing previous studies with the best model from each dataset using either F1 micro or F1 macro evaluation. The hyperparameters used in this study are the same as those carried out by (Wilie *et al.*, 2020) listed in Table 2. Preprocessing and model design used are as described in the previous chapter.

The following are the comparison results of the NERUI F1 micro dataset in Table 5. Comparison with research by (Koto *et al.*, 2020) using the same Pre-training but using different hyperparameter configurations. Fine-tuning configuration by adding a classification layer, learning rate 5e-5, epoch 100, and early stopping (patience = 5). Koto *et al.*, (2020)'s model produced an average F1 micro of 90.1%, whereas in this study, the model produced an average F1 of 95%.

Table 5. Comparison (Koto *et al.*, 2020) dataset NERUI

Research	Dataset	Pre-training	Method	F1 AVG
This study	NERUI	Indolem/ indobert-base-uncased	BERT	95%
Koto et al.,(2020)	NERUI	Indolem/indobert-base-uncased	BERT	90.1%

The following are the comparison results of the NERUI F1 micro dataset in Table 6. Comparison with research by (Koto *et al.,* 2020)using the same Pre-training but using different hyperparameter configurations. Fine-tuning configuration by adding a classification layer, learning rate 5e-5, epoch 100, and early stopping (patience = 5). Koto *et al.,* (2020)'s model, produced an average F1 micro of 74.9%. In this study, the model produced an average F1 of 84%.

Table 6. Comparison (Koto *et al.,* 2020) with dataset NERUGM

Research	Dataset	Pre-training	Method	F1 AVG
This study	NERUGM	Indolem/indobert-base-uncased	BERT_Bi-GRU_CRF	84%
Koto et al., (2020)	NERUGM	Indolem/indobert-base-uncased	BERT	74.9%

The following are the F1 micro comparison results in Table 7, with research by Azzahra et al. In their research. Azzahra et al did not use Pre-training. The embedding process was obtained directly from the dataset. Meanwhile, in data preprocessing, a standard preprocessing assessment is chosen without using special data pre-processing such as lowercase, punctuation, and cleansing. Hyperparameters is used in LSTM, Bi-LSTM, Bi-GRU with 30 epochs, batch size 64, SoftMax activation function, ADAM optimization and dropout 0.5. Meanwhile, CNN uses filter of 39, kernel size of 4, epoch of 30, batch size of 8, SoftMax activation function, ADAM optimization and dropout of 0.5. Azzahra et al.'s model produced an F1 micro average of Bi-GRU of 71.4%, Bi-LSTM of 70.41%, GRU of 63.38%, CNN of 62.92%, and LSTM of 61.65%. In this study, the model produced an average F1 of 84%.

Table 7. Comparison with (Azzahra *et al.*, 2020)

Research	Data set	Pre-training	Method	F1 AVG
This study	NERGrit-Corpus	Indolem/indobert-base-uncased	BERT. BERT_CRF. BERT_CNN_CRF	84%. 84%. 84%
Azzahra *et al.,* (2020)	NERGrit-Corpus	-	Bi-GRU. Bi-LSTM. GRU. CNN. LSTM	71.04%. 70.41%. 63.38%. 62.92%. 61.65%

The F1 macro approach is used if it is necessary to consider the performance of each label. Table 8 displays a comparison of F1 macros. Comparison with research by Willie et al. carried out with Indolem/indobert-base-uncased Pre-training, it produces better performance compared to Indobenchmark/indobert-base-p1. Combined use of CNN, Bi-GRU and CRF can improve the performance of Pre-training Indobenchmark/ indobert-base-p1. Meanwhile, Indolem/indobert-base-uncased with direct fine-tuning with fully-connected-layer also produces performance that is as good as combining CRF and Bi-GRU (81%).

Table 8. Comparison with Wilie *et al.*, (2020)

Research	Dataset	Pre-training	Method	F1 AVG
This study	NERGrit-IndoNLU	Indolem/ indobert-base-uncased	BERT BERT_CRF BERT_ Bi-GRU_CRF	81%
	NERGrit-IndoNLU	Indobenchmark/ indobert-base-p1	BERT_CNN _Bi-GRU BERT_CNN _CRF	75%
Wilie *al.*,(2020)	NERGrit-IndoNLU	Indobenchmark/ indobert-base-p1	BERT	67.42%

5. Conclusion

The results of the fine-tuning carried out resulted in NER with better performance. F1 micro dataset NERUI 95% (BERT), F1 micro dataset NERUGM 84% (BERT_Bi-GRU_CRF), and F1 micro dataset NERGRIT-Corpus 84% (BERT, BERT_CRF, BERT_CNN_CRF), and F1 macro dataset NERGRIT-IndoNLU 81% (BERT, BERT_CRF, BERT_ Bi-GRU_CRF). This proves that in this study the model design, parameter configuration, hyperparameters and preprocessing carried out can produce a better model than the compared models. However, not all datasets give better results in the fusion, on the smallest dataset in this study NERUI, tuning with fully-connected-layer gave the best results. On other datasets, NERGRIT-Corpus and NERGRIT-IndoNLU, the ML fusion performs as well as direct tuning with the fully-connected-layer.

The enhancement in ML fusion occurred on the NERUGM dataset. Carrying out several ML fusions, BERT_CNN produced the best overall average with an accuracy of 95.43%. F1 micro 80.9% and F1 macro 76.16%. This shows that the BERT_CNN model produces stable improvements on the entire dataset. Even though it did not produce the best performance on one of the datasets selected in this study, adding a CNN layer provided insight into the overall good model. This NER model can be used for location extraction.

Acknowledgement

The authors gratefully acknowledge the research funding provided by the Ministry of Education, Culture, Research and Technology through the PUPT program, the National Research and Innovation Agency (BRIN) through RIIM scheme and extend their thanks to Pustakadata.id for providing the testbed.

References

Alfina, I., Manurung, R., Fanany, M.I., 2016. DBpedia Entities Expansion in Automatically Building

Dataset for Indonesian NER. *2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, 335-340. https://doi.org/10.1109/ICACSIS.2016.7872784

Azzahra, N.S., Ibrohim, M.O., Fahmi, J., Apriyanto, B.F., Riandi, O., 2020. Developing Name Entity Recognition for Structured and Unstructured Text Formatting Dataset. *2020 Fifth International Conference on Informatics and Computing (ICIC)*, 1-7. https://doi.org/10.1109/ICIC50835.2020.9288566

Budi, I., Suryono, R.R., 2023. Application of Named Entity Recognition Method for Indonesian Datasets: a Review. *Bulletin of Electrical Engineering and Informatics*, 12(2), 969-978. https://doi.org/10.11591/eei.v12i2.4529

Devi, S., Fatchiya, A., Susanto, D., 2016. Kapasitas Kader dalam Penyuluhan Keluarga Berencana di Kota Palembang, Provinsi Sumatera Selatan. *Jurnal Penyuluhan*, 12(2), 144-156. https://doi.org/10.25015/penyuluhan.v12i2.11223

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2018. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805*. https://doi.org/10.48550/arXiv.1810.04805

Eftimov, T., Seljak, B.K., Korošec, P. 2017. A rule-Based Named-Entity Recognition Method for Knowledge Extraction of Evidence-Based Dietary Recommendations. *PLOS ONE*, 12(6), e0179488. https://doi.org/10.1371/journal.pone.0179488

Teknologi, G.I., 2023. Dataset NERGrit. https://huggingface.co/datasets/grit-id/id_nergrit_corpus

Koto, F., Rahimi, A., Chandra, A., 2023. Dataset IndoLEM. https://github.com/indolem/indolem

Koto, F., Rahimi, A., Lau, J.H., Baldwin, T., 2020. IndoLEM and IndoBERT: A Benchmark Dataset and Pre-trained Language Model for Indonesian NLP. *Proceedings of the 28th International Conference on Computational Linguistics*, 757-770. https://doi.org/10.18653/v1/2020.coling-main.66

Ma, X., Hovy, E., 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 1, 1064-1074. https://doi.org/10.18653/v1/P16-1101

Middleton, S.E., Kordopatis-Zilos, G., Papadopoulos, S., Kompatsiaris, Y., 2018. Location Extraction from Social Media. *ACM Transactions on Information Systems*, 36(4), 1-27. https://doi.org/10.1145/3202662

Nasichuddin, M.A., Adji, T.B., Widyawan, 2018. Performance Improvement using CNN for Sentiment Analysis. *IJITEE (International Journal of Information Technology and Electrical Engineering)*, 2(1), 9-14. https://doi.org/10.22146/ijitee.36642

Nuranti, E.Q., Yulianti, E., 2020. Legal Entity Recognition in Indonesian Court Decision Documents using Bi-LSTM and CRF Approaches. *2020 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, 429-434. https://doi.org/10.1109/ICACSIS51025.2020.9263157

Putra, F.N., 2021. Ekstraksi Informasi Menggunakan Kombinasi Metode NeuroNER, Neural Relation Extraction, dan FASM pada Deteksi Kejadian dari Data Stream Twitter. *Tesis*. Surabaya: Institut Teknologi Sepuluh Nopember.

Sang, E.F.T.K, Meulder, F.D., 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, 142-147.

Simanjuntak, L.F., Mahendra, R., Yulianti, E., 2022. We Know You Are Living in Bali: Location Prediction of Twitter Users Using BERT Language Model. *Big Data and Cognitive Computing*, 6(3), 77. https://doi.org/10.3390/bdcc6030077

Situmeang, S., 2022. Impact of Text Preprocessing on Named Entity Recognition Based on Conditional Random Field in Indonesian Text. *Jurnal Manajemen, Teknologi Informasi dan Komputer (Mantik)*, 6(1), 423-430.

Statista, 2024. Twitter: Most Users by Country.

Sukardi, S., Susanty, M., Irawan, A., Putra, R.F., 2020. Low Complexity Named-Entity Recognition for Indonesian Language using BiLSTM-CNNs. *2020 3rd International Conference on Information and Communications Technology (ICOIACT)*, 137-142. https://doi.org/10.1109/ICOIACT50329.2020.9331989

Takahashi, K., Yamamoto, K., Kuchiba, A., Koyama, T., 2022. Confidence interval for micro-averaged F1 and macro-averaged F1 scores. *Applied Intelligence*, 52(5), 4961-4972. https://doi.org/10.1007/s10489-021-02635-5

Utomo, M.N.Y., Adji, T.B., Ardiyanto, I., 2018. Geolocation Prediction in Social Media Data using Text Analysis: A Review. *2018 International Conference on Information and Communications Technology (ICOIACT)*, 84-89. https://doi.org/10.1109/ICOIACT.2018.8350674

Wahyuni, N.M.S., Sanjaya ER, N.A., 2021. Rule-Based Named Entity Recognition (NER) to Determine Time Expression for Balinese Text Document. *JELIKU (Jurnal Elektronik Ilmu Komputer Udayana)*, 9(4), 555-562. https://doi.org/10.24843/JLK.2021.v09.i04.p14

Wilie, B., Vincentio, K., Winata, G.I., Cahyawijaya, S., Li, X., Lim, Z.Y., Soleman, S., Mahendra, R., Fung, P., Bahar, S., Purwarianti, A., 2020.

IndoNLU: Benchmark and Resources for Evaluating Indonesian Natural Language Understanding. *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*, 843-857.

Winata, G.I., 2023. Dataset IndoNLU. https://github.com/indobenchmark/indonlu/tree/master/dataset