Received: May 11, 2023. Revised: June 16, 2023.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Improving Named Entity Recognition in Bahasa Indonesia with Transformer-Word2Vec-CNN-Attention Model

Warto[1,2] Muljono[1]* Purwanto[1] Edi Noersasongko[1]

[1]*Department of Computer Science, Dian Nuswantoro University, Semarang, Indonesia*
[2]*Department of Informatics, UIN Saizu, Purwokerto, Indonesia*
* Corresponding author's Email: muljono@dsn.dinus.ac.id

Abstract: Named entity recognition (NER) is one of the topics that get the attention of NLP (natural language processing) researchers. Most NER research uses English datasets in other languages, such as Bahasa Indonesia. However, NER is essential in recognizing corpus entities that can improve NLP performance. Deep learning approaches are currently a trend, including in NER research. In this article, we propose TWCAM (transformer-Word2Vec-CNN-attention model). Combining these models can improve NER performance in Bahasa Indonesia by obtaining better vector representations of words, extracting features in sentences, and notice to the surrounding context. The dataset in Bahasa Indonesia comes from several online news sites. Our annotation scheme is BIOLU (Begin, inside outside, last, unit). Using the learning-rate finder, the maximum F1-Score in the tests we conducted on the TWCAM was 0.8178, while the BiLSTM (Bidirectional long short-term memory) was only 0.7200. The following research opportunity is how to reduce computational complexity but not decrease overall NER performance.

Keywords: Named entity recognition, Bahasa Indonesia, Transformer with attention, Word2Vec, Convolutional neural network.

1. Introduction

Named entity recognition is one of the research topics in the field of NLP (natural language processing), tasked with identifying and classifying named entities in text. Commonly recognized entities are person (PER), location (LOC), and organisation (ORG). One of the focuses of NER's research development is to find more accurate and robust models that address multiple domains and languages. The majority of current NER research is on English. One of the reasons there is more NER research for English than any other language is the availability of quality English-language annotation datasets. Many corpora are extensive and available to the English-speaking public, such as, CoNLL-2003 [1], JNLPBA and GENIA [2], NCBI [3], etc. This data set allows researchers to train and evaluate NER models with high accuracy and consistency. Another reason is that English is the dominant language in many fields, including business, science, and technology, leading to a higher demand for the NER system in English. As a result, there is a more significant market and incentive for researchers to develop and improve English NER models.

However, in recent years, NER research has significantly increased in languages other than English. Bahasa Indonesia is one of the languages that has a predicate of few resources on the research topic of NER. In the early era, NER research in Bahasa Indonesia was conducted by [4, 5]. He uses a rule-based approach with the maximum entropy method with a news corpus in online national daily. In the following years, more and more researchers and academics developed NER in Bahasa Indonesia, including [4, 6–17].

Aryoyudanta used a semi-supervised approach with a co-training algorithm [6], while R. A. Leonandya proposed a semi-supervised algorithm for NER [10]. Gunawan [9] used a LSTM-CNN model, Azalia [7] used a Naïve Bayes classifier for name indexing in the Indonesian translation of hadith, and

Fu proposed a hierarchical structured-attention-based feature method for NER [8]. Various corpora developed from many sources, such as news articles [6], Wikipedia [9, 10], and religious texts [7]. However, some studies use more specialized datasets, such as hadith translations in Azalia [7], which limits their applicability to other domains.

Rachman [11], focuses on NER on Indonesian Twitter posts using LSTM networks. They utilized a dataset consisting of 480 tweets labeled with three entity types, such as Person, Location, and Organization. The experiment was conducted using a 90-10 train-test split, and the results showed that the LSTM network outperformed other baseline models, achieving a 77.08% F1 score. Next, Setiyoaji [14], proposed a hidden Markov model (HMM) and Viterbi algorithm-based approach for NER on medicinal plant texts in Indonesian language. The experiment was conducted using a 70-30 train-test split, and the results showed that the proposed approach achieved an F1 score of 72.55%, outperforming other baseline models.

D'Souza discuss the challenge of NER in Examining existing CS NER language resources and clarifying the problem of the lack of standardized entities therein [18]. The combination of BiLSTM + CRF shows an f1 score of 72.62, while BiLSTM + CNN + CRF is 75.18. Combining existing resources that fulfill the aim of contribution-centric extraction targets and further annotating additional data to create a large corpus which is made publicly available. Lai [19] proposed Transformer-based approach achieved competitive results in the leaderboard and ranked 12th out of 30 teams. The system achieved a macro F1 score of 72.50% on the held-out test set. The data augmentation approach using entity linking involved replacing named mentions in the training set with different entities that are also corporations. The replacement entities were obtained by using an entity linker to link the named mention to its corresponding entity in Wikidata, a large-scale knowledge graph. However, the approach did not improve the final performance of the system.

Cho [20], proposes a deep learning model for biomedical NER that incorporates a combinatorial feature embedding using BiLSTM with conditional random field (CRF), and integrates two different character-level representations extracted from a CNN and Bi-LSTM. The model also employs an attention mechanism to focus on relevant tokens, resulting in an F1-score of 75.31% on the JNLPBA dataset. Whereas, Che [21] proposes a temporal convolutional network with a conditional random field (TCN-CRF) for biomedical named entity recognition (Bio-NER), as a computationally efficient alternative to the state-of-the-art models based on BiLSTM and BERT. The proposed model achieves comparative performance with much less training time, as demonstrated through experiments on the GENIA with F1-score 76.45% with three and five kernel sizes. Deng [22] proposed model for NER that uses a self-attention-based bidirectional gated recurrent unit (BiGRU) and capsule network (CapsNet), and achieves improved performance about 78.16% in Diabetes dataset.

Each language has different characteristics from one another, including phonology [23], morphology [4], and grammatical structure [24]. Sentence structure in English is generally more straightforward compared to Bahasa Indonesia. In English, the subject is usually placed at the beginning of a sentence so that named entities, such as people or places, are easier to identify. However, in Bahasa Indonesia, sentence structure can be more complex and difficult to parse. English has more resources for NER model development, such as a more extensive and diverse corpus of text, well-structured dictionaries, and databases of named entities. On the other hand, resources for Bahasa Indonesia are still limited and in the development stage.

Differences in language structure and characteristics of each language can affect the NER process in English and Bahasa Indonesia. Although there are differences in the NER process between English and Bahasa Indonesia, the techniques and algorithms used in the NER process are essentially the same [25]. It means that effective NER techniques in English can also be applied to Bahasa Indonesia with certain adjustments. NER for Bahasa Indonesia is still in the development stage. Still, with the existence of machine learning and deep learning technology, it is expected that NER techniques for Bahasa Indonesia continue to develop and become more sophisticated.

The NER approach uses deep learning, widely developed worldwide, including recurrent neural network (RNN), LSTM, BiLSTM, and transformer. The advantages of NER using a deep learning approach are that it can handle large amounts of data [26], improve accuracy [27], reduce human interference in annotation [28, 29]. Many researchers also combine various deep learning approaches to get maximum performance. Deep learning-based NER research conducted by [9, 13, 17] using several deep learning algorithms, including RNN, BiLSTM, Transfer Learning, and CNN.

The various methods in the deep learning approach each have advantages and disadvantages. BiLSTM can handle problems in long or complex sentences by arranging sequences processed using

the RNN architecture [30]. However, it tends to keep the first words of a sentence than the words in the middle or end [31]. The advantages of the Word2Vec method are that it can learn better and more complex word representations from the given text to improve model performance [32]. Still, on the other hand, Word2Vec is less able to cope with out of vocab (OOV) [9]. CNN's character embedding solves the OOV problem because models can learn patterns from characters in never-before-seen words. This technique can be done by converting each word into a sequence of characters and embedding them in vector space using the convoltional neural network (CNN) character model [33]. CNN does not explicitly address OOV issues, so more specialized techniques, such as those used in Transformer models, improve performance in addressing OOV issues on NER. The combination of Attention and Transformer can help improve the performance of deep learning models by paying Attention to the context of more specific sentences, overcoming "long-tail" problems in entities, combining information from multiple sources, and obtaining better vector representations each word in the sentence. Therefore, the combination of Attention and Transformer can be an excellent choice to improve NER performance [34], Including in Bahasa Indonesia.

From some advantages and disadvantages in BiLSTM, Word2Vec, and CNN, we propose a combination of Transformer, Word2vec, CNN, and attention, especially in Bahasa Indonesia. We named this method TWCAM (Transformer-W2V-CNN-attention model) to handle the NER task in Bahasa Indonesia. This article is arranged systematically at the beginning, namely section 1 introduction, section 2 related work, section 3 proposed method, section 4 experiment setup, section 5 result and discussion, and section 6 conclusion.

2. Related work

Bahasa Indonesia is one of the languages with a large number of speakers. But NER research related to Bahasa Indonesia has been less. As stated in the introduction, NER research in Bahasa Indonesia since 2003 [4], [5]. They resurfaced NER research Bahasa Indonesia 2015 by [10] using a semi-supervised approach with datasets from Wikipedia Dump and DBPedia. While [16] use the ensemble supervised learning method. Along with the popularity of deep learning approaches, the most advanced technique in NER Bahasa Indonesia is Bi-LSTM combined with CNN [9, 17, 35, 36]. Combining the BiLSTM method with CNN in the Named Entity Recognition experiment has advantages because BiLSTM can overcome remote context dependencies. At the same time, CNN can extract local features useful for recognizing named entities.

Sequence labeling in NER involves the task of classifying each token in a sentence as a name entity or not. As for labeling sequences, the majority use CRF [10, 12, 15, 17, 36]. Leonandya [10] using CRF to explores morphological features, contextual features, and POS-Tags in Wikipedia datasets. By combining CRF and K-means methods, Santoso [12] provided NER performance improvements of up to 4.3% for Indonesian news datasets with person, location and organization entities. CRF has several advantages, including modeling dependencies between tokens in a sentence. CRF allows flexible feature engineering, which means that models can learn from various features such as word embedding, part-of-speech tags (POS-Tag), and context information.

One aspect of named entity research is datasets. Datasets can come from various sources, including Wikipedia, news, and Twitter. The advantages of the Wikipedia dataset are large volumes and neatly structured sentences, but sometimes they need to be completed and updated. Datasets from news portals have the advantage of being up-to-date and accurate, but they bias in meaning. While the Twitter dataset has the advantage of being up-to-date, most use slang. However, the NER researchers in Bahasa Indonesia used datasets sourced from Wikipedia [5, 6, 9, 10], news [4, 8, 12, 13, 16, 36], and Twitter [11, 15, 17, 37]. There are also some researchers who use specific datasets, for example, health datasets [35], and the religion field [7]. Many datasets in Bahasa Indonesia variants must indicate state-of-the-art NER research in Bahasa Indonesia. Starting in 2020, several researchers curated a public dataset under the name IndoBenchmark [38], such as NER-Grit[1] and NER-Pros[2]. The dataset is expected to be a benchmark for NER research in Bahasa Indonesia.

Current state-of-the-art deep learning transformers provide great opportunities for Bahasa Indonesia NER researchers. IndoBERT Transformer-based NER in Bahasa Indonesia began to be developed in 2020 by [38] using FastText, BERT models with various variations, and XLM-R. Until February 2023, NERGrit achieved 0.7997 and 0.8450

[1] https://github.com/IndoNLP/indonlu/tree/master/dataset/nergrit_ner-grit

[2] https://github.com/IndoNLP/indonlu/tree/master/dataset/nerp_ner-prosa

for the NERProsa dataset using the IndoBERT-large-p2 model[3].

BERT has many parameters and layers, so it can take a long time to train a model and apply it to more extensive data. As a dataset-dependent model [39], BERT requires a large, representative Bahasa Indonesia dataset to produce good results [38]. However, the Bahasa Indonesia datasets currently available are limited, including the dataset we used in this experiment. We can cover this BERT weakness with Word2Vec, which can provide good performance on dataset sizes that are not too large. FastText, in addition to providing lower performance than BERT, also has a weakness in entity recognition that rarely appears in training data, known as the "long-tail" problem.

The embedding layer in Xiao [40] is only based on word embedding, while in Zhai [33]use character embedding. To model the meaning and context of words in a sentence, we combine word and character embedding to capture significant morphological and syllable information.

As we mentioned in the introduction, this weakness we can overcome using Transformer-Attention. We propose a combination of Word2Vec, CNN, Transformer, and Attention to improving NER performance in Bahasa Indonesia, especially with smaller dataset collections.

3. Proposed method

In this section, we describe the proposed architecture of TWCAM (Transformer-Word2Vec-CNN-Attention Model) in Fig. 1. Our model consists of three parts, the first is the embedding layer, the second is the transformer-attention layer, and the third is the sequence labeling layer. The combination of Word2Vec, CNN, attention, and transformer can help improve the performance of deep learning models by heed to the more specific context of sentences, overcoming "long-tail" problems in entities, combining information from multiple sources, and obtaining better vector representations for each word in the sentence. Therefore, combining Word2Vec, CNN, Attention, and Transformer can be an excellent choice to improve NER performance in Bahasa Indonesia.

First, text sequences are fed into the model using tokenization, where each word is converted into a numerical representation, such as a word index or word embedding vector. Later, CNN extracted features from text sequences, paying Attention to

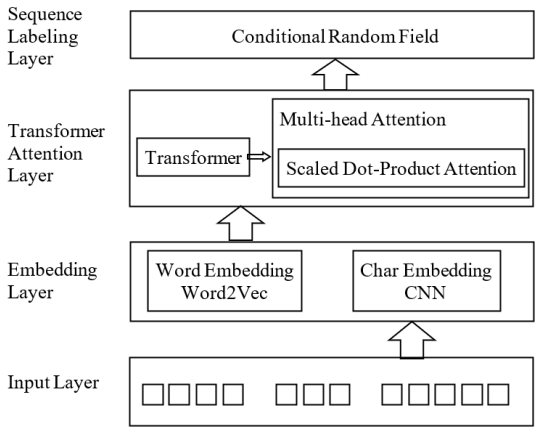

Figure. 1 NER architecture with Transformer-Word2Vec-CNN-Attention Model (TWCAM)

local information such as adjacent words. CNN generates a sequence of features representing a sequence of text, where each feature element represents information related to the words in the sequence. Furthermore, the feature extraction results from Word2Vec and CNN are given to transformer for advanced processing. In this stage, the attention mechanism is used to heed global information in the text sequence, where each word can pay attention to information from other words at a long distance. Each word in the CNN sequence feature is used as a key and value in the Attention mechanism. It derives a weight for each word in the text sequence. This weight indicates the importance of the information the word conveys to perform classification or prediction.

This process is done by calculating the dot product between the key and the value, and the results are processed using softmax to obtain weights that indicate the level of Attention given to each word in the text sequence. In the attention mechanism, a "key" is used to identify the parts of the input that are relevant to attention. Each key has a numerical representation known as a "value", representing the information associated with that key. The key is a word in a sentence, while the value represents the meaning or context associated with the word [41]. The transformer architecture we use is multi-head Attention [42] with scaled dot-product attention [43]. The output on the attention transformer layer goes to the sequence labeling layer using the CRF.

3.1 Word2Vec word embedding

As described in the dataset section, this study used a fairly small dataset, 4473 sentences. With limited datasets, training high-quality models from scratch is not easy. Transfer learning with Word2Vec

[3] https://www.indobenchmark.com/leaderboard.html

that has been trained can help overcome these shortcomings [40]. The Word2Vec model training process is carried out by utilizing the context of words in a text, where each word will be represented in the form of numerical vectors that reflect the context and meaning of the word [44]. The Word2Vec model then processes large amounts of text to learn the interrelationships between words in different contexts. During training, the model predicts the context of a given the word. In this process, the model will update existing words and create interrelated word representations in a high-dimensional vector space. Some Word2Vec embeddings available include GloVe [45], FastText [32], and Gensim [46]. By initiating a model using pre-trained embedding, we can leverage the knowledge captured in the embedding to improve model performance. In addition, embedding Word2Vec can also reduce the risk of overfitting [40]. Because pre-trained embedding that has been trained uses extensive data, it makes it less likely to overfit into smaller data.

3.2 Character embedding CNN

Convolutional neural network (CNN) was originally for image processing, but in its development can be used for natural language processing tasks, such as character embedding. Some researchers have implemented CNN for character embedding [47–50]. They successfully combined CNN with RNN, LSTM, BiLSTM, and CRF to improve NER performance. CNN for character embedding by treating characters as pixels in an image to extract various features of the text that word embedding cannot do, such as capital letter features.

CNN architecture typically consists of several convolution layers, a max-pooling layer, and a fully connected layer, as seen in Fig. 2. Input into the CNN network is in the form of characters represented in a one-hot vector. CNN for character embedding starts by converting input text into vector representations, then applying convolutional filters to extract features. The convolutional layer output is then passed to the pooling layer. Layer pooling helps simplify input representation and reduce subsequent layer computational costs. Thus, the model can learn to recognize patterns in text data and make predictions based on those patterns.

3.3 Transformer multi-head attention

As explained in the previous section, CNN generates a sequence of features that represent a sequence of text, where each feature element

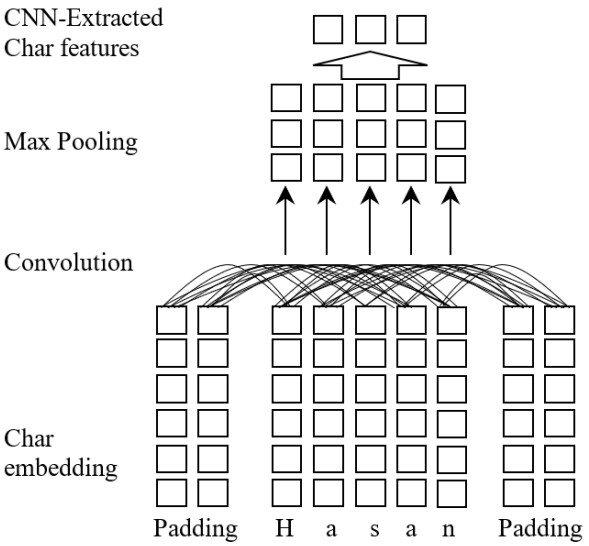

Figure. 2 CNN for character embedding

represents information related to the words in the sequence. The feature extraction results from CNN are given to Transformer for advanced processing, as seen in Fig. 1. In this stage, the Attention mechanism is used to observe the global information in the text sequence, where each word can heed information from other words at a longer distance.

The Multi-head Attention mechanism we used in this experiment uses formulas 1-3 [34], where each word in the sequence features of the CNN is used as the key *K*, and the value of *V* is used to derive the weight *W* for each word in the sequence of text. This weight indicates the importance of the information the word conveys to perform classification or prediction. This process is done through a set of *Q* queries in matrix form with dot-product calculations between the key *K* and the value of *V*. The results are processed using softmax to obtain weights that indicate the attention level given to each word in the text sequence.

$$\text{Attention} = (\text{Q}, \text{K}, \text{V}) = \text{Softmax}(\frac{\text{QK}^{\text{T}}}{\sqrt{\text{d}_\text{k}}})\text{V} \quad (1)$$

$$\text{MultiHead}(\text{Q}, \text{K}, \text{V}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_\text{h})\text{W}^{\text{O}} \quad (2)$$

where,

$$\text{head}_\text{i} = \text{Attention}(\text{QW}_\text{i}^{\text{Q}}, \text{KW}_\text{i}^{\text{k}}, \text{VW}_\text{i}^{\text{V}}) \quad (3)$$

where:

Q : query input vector
K : key input vector
V : value input vector
QW_i^Q : the linear weight matrix used to project Q
KW_i^k : the linear weight matrix used to project K
VW_i^V : the linear weight matrix used to project V
$\sqrt{d_k}$: dimensi dari setiap subruang linier

After obtaining the weight of Attention, the information of the entire sequence of text can be aggregated into a single vector of representation, which is used to perform classification or prediction [51]. This vector is generated by summing each value in the feature sequence of the CNN, multiplied by the corresponding attention weight. This process allows the model to observe the local and global information in text sequences for accurate classification or predictions.

3.4 Conditional random field

In the labeling sequence stage of the named entity recognition (NER) task, the conditional random field (CRF) algorithm works by predicting the sequence labeling of the input text. The CRF algorithm considers the global context of the entire text, so it can improve label predictions on certain words that depend on the words before or after it [12]. Probabilistic models are built to study the relationship between features and labels. At this stage, the CRF model considers the probability of the optimal label order based on the sequence of features in the text. The model can be learned from properly labeled training data. Probabilistic models are trained using training data. At this stage, the weights of the features are arranged so that the model can produce the most accurate predictions of labels based on the given features. Once the model is trained, it can predict labels on new data. At this stage, the CRF model selects the most likely order of labels based on the order of features in the text.

CRF algorithm proposed by [52] calculates the probability of the label sequence using factors related to the text and the previous label. The label sequence probability is expressed as the product of potential factors as functions of the previous word and label.

$$P(y_i|x, y_{i-1}) = \frac{\exp(\sum_k w_k f_k(y_i, y_{i-1}, x_i))}{\sum_{y'} \exp(\sum_k w_k f_k(y', y_{i-1}, x_i))} \quad (4)$$

Where

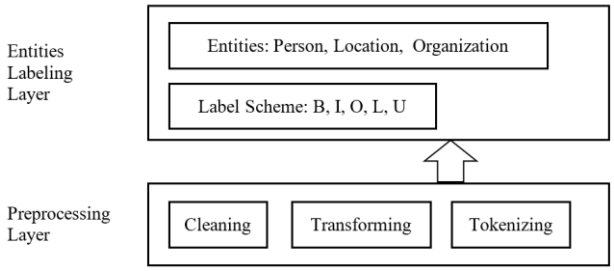

Figure. 3 Preprocessing stage and entity labeling

y_i : label for the i^{th} word in the text
x : input text
y_{i-1} : label for the previous word
w_k : weight for the k^{th} potential factor

CRF uses factors involving previous labels (y_{i-1}) and current labels (y_i), as well as features related to tokens (X) and positions (i or k) in the sentence. The parameters w_k and f_k are used to set the weight for each feature and influence its contribution to the label order score. The goal of the CRF is to estimate the probability $P(Y|X)$ for each sequence of Y labels based on the features and parameters found from the training data. The probabilities values are used to select the highest-probability label sequences as predictions for the tokens in the sentence.

4. Experiment setup

Our Experiments using Jupyter Notebook on Python 3.7 by utilizing several libraries, including Torch, Torchtext, Spacy, Gensim, Scikit-Learn, Matplotlib, Numpy, and Pytorch-crf. The experiment used a 2.6GB Core I7-9750H 9th Gen CPU machine, 12GB Nvidia GeForce GTX 1650 GPU, 16GB RAM, and Ubuntu 22.04 LTS operating system.

4.1 Dataset

The dataset in this experiment used the corpus NER Gultom[4] and Syaifudin[5], which Consists of 4473 sentences divided into training data (3535), validation (470), and test (468). We use this dataset because it is already labeled entities manually and has a diversity of entities, so the model can learn to recognize different entities more accurately. The dataset is publicly accessible and can be an alternative benchmark for representative NER research in Bahasa Indonesia. Preprocessing dataset, as shown in Fig. 3, using the BILOU (Begin, Inside, Outside, Last, Unit) annotation scheme. The *U* label is for entities that consist of only one word, while the

[4] https://github.com/yohanesgultom/nlp-experiments/tree/master/data/ner

[5] https://github.com/yusufsyaifudin/indonesia-ner/tree/master/resources/ner

O label indicates that the word is not an entity. Entities consisting of the first two words are labeled *B,* and the second word is labeled *L*. Furthermore, for entities consisting of more than equal three words, the first word is labeled *B,* the word in the middle uses the label *I,* and the last word in the entity is labeled *L*.

As stated by [53], BIOLU annotation scheme has several advantages over the BIO annotation scheme. Some of these advantages include: 1) calculating unit entities well, 2) being more expressive by using *L* and *U* tags, the BIOLU scheme can enrich the types of entities that can be identified, and 3) reducing tagging errors. The BIOLU schema minimizes the possibility of tagging errors in the BIO annotation scheme. However, the BIOLU scheme also holds a potential drawback, namely the difficulty in processing overlapping entities and covering all tokens in a sentence. In addition, poor understanding and use of the BIOLU scheme can lead to errors in entity tagging.

4.2 Hyperparameter

Parameters in deep learning experiments must be set before model training begins because hyperparameters can affect model performance and allow users to customize the model to suit the task or problem to be solved [54]. Hyperparameter settings in LSTM, CNN, Transformer, and Attention models, as shown in Table 1. After several experiments by changing the hyperparameter values, the optimal value is obtained as in table 1. Proper hyperparameter settings can help improve model performance and enable the model to produce the best performance in resolving NER issues. However, finding the right combination of parameters can take a lot of time and computational resources [54], [55]. So it needs to be done carefully, and the expansion of the model is done gradually by testing more data to see if the model still has good performance.

4.3 Evaluation metrics

We use the F-score evaluation method, as many other researchers use [56]. The F-score, as Eq. (7), is the harmonic mean of precision and recall. The F-score measures the balance between precision and recall and provides a single value that describes the overall performance of the NER system. Precision, as shown in Eq. (5), measures how accurately the system recognizes the relevant entities from all entities identified by the system. Precision is expressed as the ratio between the number of entities correctly identified with the number of entities identified by the system [51]. Recall, as shown in Eq. (6), measures how many relevant entities the system can identify from all the entities present in the data. A recall is expressed as the ratio between the number of correctly identified entities to the number of correctly identified and undetected entities by the system.

Table 1. Hyperparameter setting for proposed model.

LSTM	Hidden dimension: 64 Layer: 2 Dropout: 0.1 Optimizer: Adam Loss Function: Cross Entropy
CNN	Embedding dimension: 37 Embedding dropout: 0.25 Filter number: 4 Kernel size: 3 CNN dropout: 0.25
Attn	Attn Head: 16 Attn Dropout: 0.25 Attn type: dot-product attn
Transformer	Trf Layer: 2 Fc Hidden: 256

$$Precision = \frac{TruePositives}{TruePositives+FalsePositives} \quad (5)$$

$$Recall = \frac{TruePositives}{TruePositives+FalseNegatives} \quad (6)$$

$$F\text{-}score = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (7)$$

The F-score value will always be between the precision and recall values. *True positive* (TP) is an entity that the system has successfully identified correctly, *false positive* (FP) is an entity that is incorrectly identified by the system, and *False Negative* (FN) is an entity that the system fails to identify [57].

5. Result and discussion

This study aims to produce NER model that recognizes named entities in Bahasa Indonesia. The results of our experiments show that deep learning-based NER models perform better than NER models that only use conventional machine learning methods. In addition, using techniques such as transformer and attention combined with Word2Vec, and CNN also provides significant performance improvements to NER models.

CNN's character embedding is responsible for identifying capital letters, as we know that the capital letter feature indicates that the word containing capital letters is a certain entity [57]. As a general rule

Table 2. Number of parameters and computational time required to complete various models

Model	Number of Parameters	Training time
BiLSTM	9.305.086	4m 48s
Tranformer+W2V (TW)	9.305.086	7m 20s
Transformer+CNN (TC)	9.384.599	12m 3s
Transformer+W2V+CNN (TWC)	9.384.599	11m 36s
Transformer+W2V+Attn (TWA)	9.371.134	7m 50s
Transformer+W2V+ CNN+Attn (TWCAM)	14.428.183	8m 26s

Table 3. The optimal number of epochs in each model and the respective f1 score

Model	Epoch	F1 Train	F1 Valid
BiLSTM	16	0.9037	0.5981
Tranformer+W2V	32	0.7607	0.6700
Transformer+CNN	**49**	0.5518	**0.5789**
Transformer+W2V+CNN	44	0.5935	0.6233
Transformer+W2V+Attn	48	0.6316	0.6129
Transformer+W2V+ CNN+Attn	**20**	**0.8178**	0.7267

of language, writing the name of a person, the name of an organization, or the name of a place begins with a capital letter at the beginning of each word.

Based on the number of parameters and processing time, as shown in Table 2, the TWCAM method with 14 million parameters can be completed in 8 minutes and 26 seconds. In contrast, TWC, with only 9 million parameters, takes 11 minutes and 36 seconds. In the Word2Vec method, the model only considers the local context of a word, so it cannot take into account the broader context of a sentence. It can result in less accurate results in recognizing named entities in text. As for the CNN method, despite extracting features from all sentences, there are still limitations in the model's ability to account for the broader context of a document. Vaswani [34] and [41] show that the Transformer method can produce better results than the RNN method with faster computation times. In addition, Transformers can also process longer sentences in a relatively shorter time compared to RNN.

We used the learning rate finder in the experiment to determine the optimal learning rate when training the proposed model. The main purpose of the learning rate finder is to improve model accuracy by avoiding overfitting or underfitting the training data [58][6]. The learning rate finder method used (Smith, 2017) Cyclical learning rate by increasing the learning rate at the beginning and then decreasing the learning rate by following a certain pattern as the iteration progresses. This method uses one learning cycle, and the optimal learning rate finder is selected based on the most significant decrease in loss value during the learning cycle [58]. The results of the learning rate finder can find the maximum number of epochs with the best F1 score, as shown in Table 3, with graph illustration in Fig. 5. Our proposed TWCAM method produces an F1-score of 0.7267 at epoch 20. These results are far different from the Transformer method, which was only combined with CNN, which produced an F1-score of 0.5789 at epoch 49, with an illustrative graph of learning rate achievement as shown in Fig. 4.

In these experiments, techniques such as BiLSTM, Word2Vec, CNN, Transformer, and Attention were used to improve the accuracy of NER. Learning rate analysis aims to determine the best learning rate that can be used in training the model. The learning rate is the quantity used to regulate how much the parameters change in each iteration during model training [48]. If the learning rate is too low, then the convergence of the model will be very slow and take a long time to reach the desired level of accuracy. On the other hand, if the learning rate is too large, the model can experience an exploding gradient or divergent gradient, making the model training unstable, and the accuracy is not optimal.

To determine the best learning rate, a learning rate analysis can be performed by observing changes in loss values against variations in learning rate [41]. The trick is to run model training at different learning rate values and record loss values in each training iteration. Then, the loss value can be represented as a learning rate graph. As shown in Fig. 3, the optimal learning rate graph in each method shows different numbers. One reason to use the learning rate finder is that it controls how much of a step is taken when updating weights and biases on each iteration during training [57]. If the learning rate is too low, the model takes longer to reach convergence, while if it is too large, the training becomes unstable or even diverges.

As shown in Table 4, the performance is quite good in the TWCAM method by producing an F1-Score of 0.8178 compared to other models. The TWCAM method outperforms the other models,

[6] Learning rate finder implemented in PyTorch can be found at https://github.com/davidtvs/pytorch-lr-finder

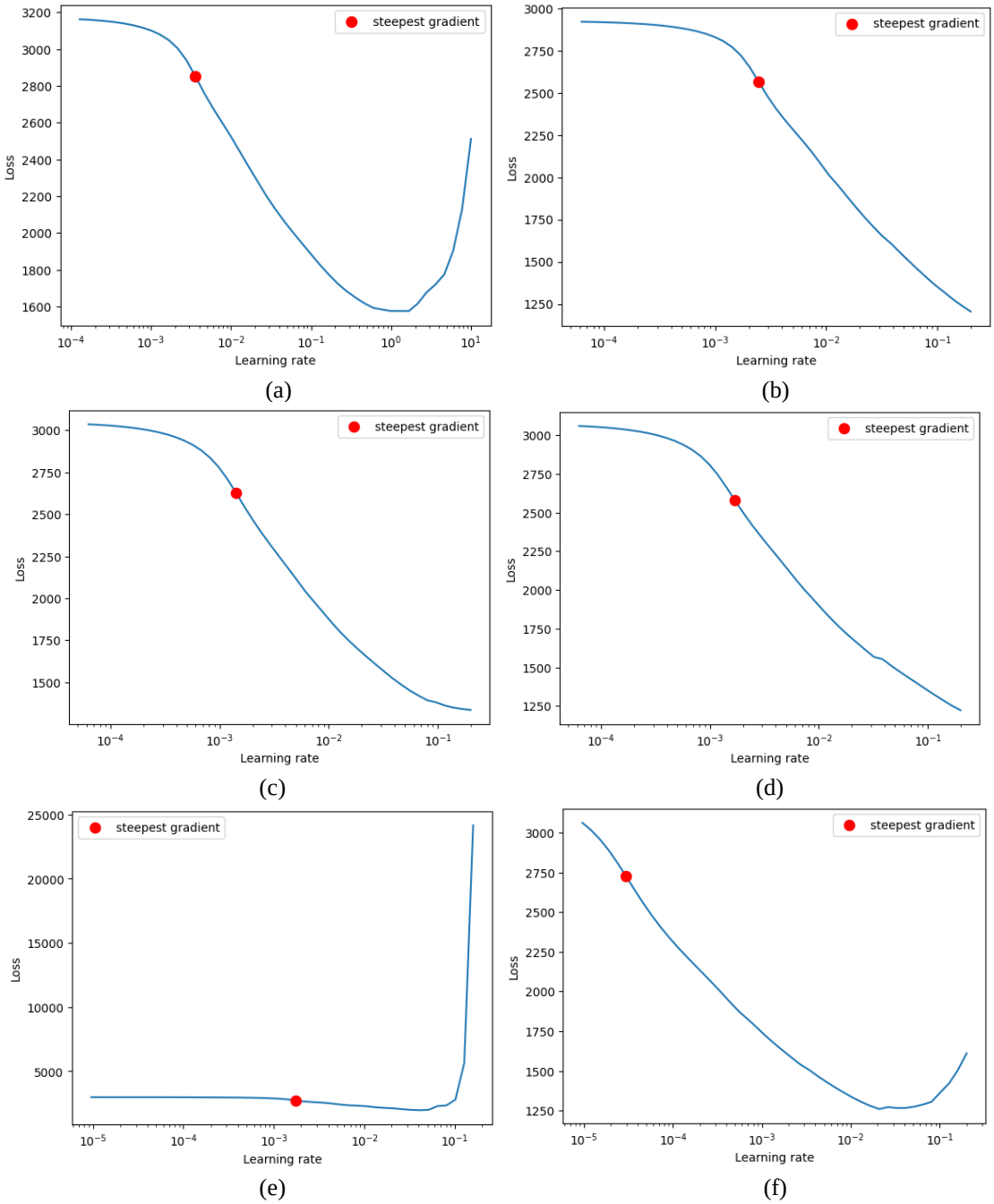

Figure. 4 Learning rate graph with loss values in several experimental schemes: (a) BiLSTM's suggested learning rate is 3.59E-03, (b) transformer+w2v is 2.45E-03. On the second row, (c) transformer+cnn is 1.41E-03, (d) transformer+w2v+cnn 1.70E-03. While in the third row, (e) transformer+w2v+attn 1.74E-03, and (f) transformer+w2v+cnn+attn 2.97E-05. The learning rate changes with each repetition of the experiment, but the change from one experiment to the next is not very significant

which is higher than the F1-Scores achieved by the other models. Specifically, the Transformer+W2V model has the lowest F1-Score at 72.50, while the BiLSTM and Transformer + CNN models have F1-Scores of 75.18 and 75.31, respectively. The Transformer + W2V + CNN and Transformer + W2V + Attn models have F1-Scores of 76.45 and 78.16, respectively, which are still lower than the F1-Score achieved by the TWCAM method.

Combination of transformer and attention provides various advantages to NER, including: 1) capturing long-range dependencies, 2) handling long variable inputs, 3) learning contextual representations, and 4) handling multiple entities in a single sentence. Transformer architecture that uses self-attention allows the model to consider the context of each word in a sentence, including those that have the position farthest from the position of the

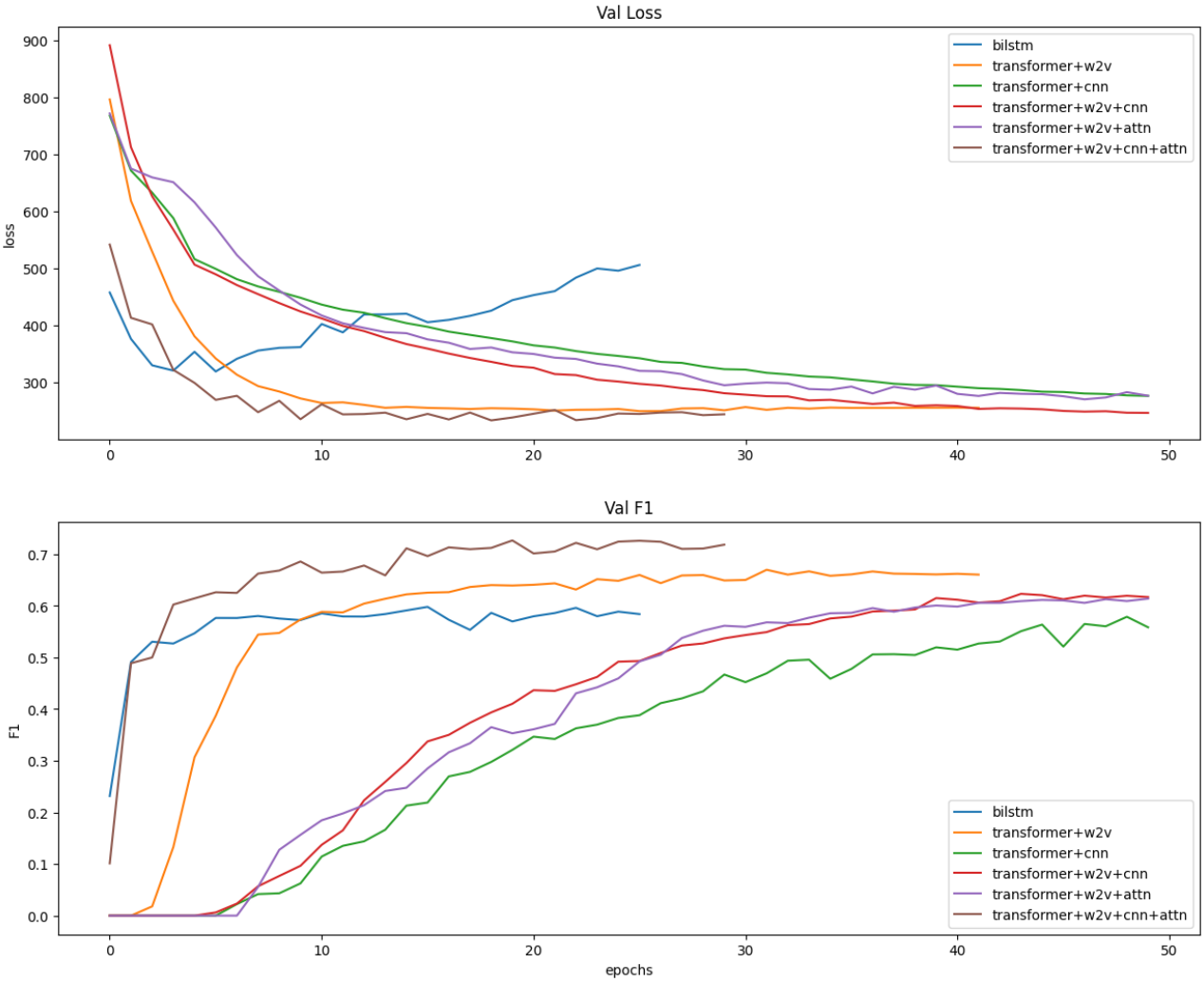

Figure. 5 Graph illustrating the relationship between the number of epochs on the x-axis and loss on the y-axis in the NER BiLSTM, Word2Vec, CNN, Transformer, and Attention experiments. The lowest loss value is found in the TWCAM model scheme

Table 4. Proposed TWCAM method is compared with others

Method	F1-Score
BiLSTM [18]	75.18
Transformer+W2V [19]	72.50
Transformer+CNN [20]	75.31
Transformer+W2V+CNN [21]	76.45
Transformer+W2V+Attn [22]	78.16
TWCAM (our model)	81.78

tested word. Transformer models can learn contextual representations for each word in a sentence based on the context in which they appear. It allows the model to capture different meanings of words in different contexts, which is important for NER, as the same word can refer to different types of entities depending on the context.

6. Conclusion and future works

As we have experimented, the implementation of Word2vec, CNN, transformer, and attention are powerful deep learning techniques to improve the performance of Indonesian-based named entity recognition (NER). Word2vec to obtain a vector representation of words useful for solving the "curse of dimensionality" problem in NER. Word vector representations obtained through Word2vec can help deep learning models understand relationships between words in sentences. Transformers and Attention are used to solve the problem of "long-term dependency" on sequential data such as sentences. Both can help deep learning models acquire relevant information and focus on the surrounding context to recognize entities. Implementing deep learning-based NER in the Bahasa Indonesia corpus, using these

techniques, can help improve the performance of deep learning models by obtaining better vector representations of words and characters, extracting features in sentences, and observe to the surrounding context. Therefore, implementing Word2vec, CNN, transformer, and attention can help improve NER performance in Bahasa Indonesia. The learning rate finder mechanism that we implement can also provide a learning rate value that can add to the system's optimal training process. One of the limitations of our experiment is the use of small datasets. Using larger Indonesian datasets can improve model accuracy and recognize more various entities—advanced research opportunities, e.g., fine-tuning models that have been trained. The fine-tuning technique has been proven effective in improving NER performance in English and can help improve the model's accuracy in Bahasa Indonesia. Most sequence labeling uses the CRF method; therefore, there is an opportunity to develop a hybrid CRF with several other sequence labels, such as HMM.

Conflicts of interest

The authors declare that there is no conflict of interest in the research process, data collection, manuscript preparation, authorship, and citations.

Author contributions

This article was compiled by four authors, each with the following contributions: conceptualization and methodology by Warto and Muljono, software by Warto; formal investigation and analysis by Warto, Muljono, Purwanto, and Edi Noersasongko; resources by Muljono and Edi Noersasongko; data preparation and curation by Warto and Muljono; drafting of articles by Warto and Muljono; manuscript review and editing by Purwanto and Edi Noersasongko; supervision by Purwanto, Muljono and Edi Noersasongko.

Acknowledgment

The author would like to thank all parties who have supported research, experiment, and write this manuscript, especially the Data Mining Laboratory of Dian Nuswantoro University, Semarang, Indonesia, and the Institute for Research and Community Service, UIN Saizu, Purwokerto, Indonesia.

References

[1] E. F. Sang and F. D. Meulder, “Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition”, *arXiv preprint cs/0306050*, 2003.

[2] P. T. A. Tsujii, “GENIA Corpus: Annotation Levels and Applications, The”, *Handbook of Linguistic Annotation*, 2017, doi: 10.1007/978-94-024-0881-2_54.

[3] R. I. Doğan, R. Leaman, and Z. Lu, “NCBI disease corpus: A resource for disease name recognition and concept normalization”, *J Biomed Inform*, Vol. 47, pp. 1–10, 2014, doi: https://doi.org/10.1016/j.jbi.2013.12.006.

[4] I. Budi, S. Bressan, G. Wahyudi, Z. A. Hasibuan, and B. A. A. Nazief, “Named entity recognition for the Indonesian language: combining contextual, morphological and part-of-speech features into a knowledge engineering approach”, In: *Discovery Science: 8th International Conference, DS 2005, Singapore, October 8–11, 2005. Proceedings 8*, pp. 57–69, 2005.

[5] I. Budi and S. Bressan, “Association rules mining for name entity recognition”, In: *Proc. of the Fourth International Conference on Web Information Systems Engineering, 2003. WISE 2003.*, *IEEE*, pp. 325–328, 2003.

[6] B. Aryoyudanta, T. B. Adji, and I. Hidayah, “Semi-supervised learning approach for Indonesian Named Entity Recognition (NER) using co-training algorithm”, In: *Proc. of 2016 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, pp. 7–12, 2016.

[7] F. Y. Azalia, M. A. Bijaksana, and A. F. Huda, “Name indexing in Indonesian translation of hadith using named entity recognition with naïve Bayes classifier”, *Procedia Comput Sci*, Vol. 157, pp. 142–149, 2019.

[8] Y. Fu, N. Lin, X. Lin, and S. Jiang, “Towards corpus and model: Hierarchical structured-attention-based features for Indonesian named entity recognition”, *Journal of Intelligent and Fuzzy Systems*, Vol. 41, No. 1, pp. 563–574, 2021, doi: 10.3233/JIFS-202286.

[9] W. Gunawan, D. Suhartono, F. Purnomo, and A. Ongko, “Named-Entity Recognition for Indonesian Language using Bidirectional LSTM-CNNs”, *Procedia Comput Sci*, Vol. 135, pp. 425–432, 2018, doi: https://doi.org/10.1016/j.procs.2018.08.193.

[10] R. A. Leonandya, B. Distiawan, and N. H. Praptono, “A semi-supervised algorithm for Indonesian named entity recognition”, In: Proc. of *2015 3rd international symposium on computational and business intelligence (ISCBI)*, pp. 45–50, 2015.

[11] V. Rachman, S. Savitri, F. Augustianti, and R. Mahendra, "Named entity recognition on Indonesian Twitter posts using long short-term memory networks", In: *Proc. of 2017 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, pp. 228–232, 2017.

[12] J. Santoso, E. I. Setiawan, E. M. Yuniarno, M. Hariadi, and M. H. Purnomo, "Hybrid conditional random fields and k-means for named entity recognition on indonesian news documents", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 3, pp. 233–245, 2020, doi: 10.22266/ijies2020.0630.22.

[13] J. Santoso, E. I. Setiawan, C. N. Purwanto, E. M. Yuniarno, M. Hariadi, and M. H. Purnomo, "Named entity recognition for extracting concept in ontology building on Indonesian language using end-to-end bidirectional long short term memory", *Expert Syst Appl*, Vol. 176, p. 114856, 2021.

[14] A. Setiyoaji, L. Muflikhah, and M. A. Fauzi, "Named entity recognition menggunakan hidden markov model dan algoritma viterbi pada teks tanaman obat", *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN*, Vol. 2548, p. 964X, 2017.

[15] N. Taufik, A. F. Wicaksono, and M. Adriani, "Named entity recognition on Indonesian microblog messages", In: *Proc. of 2016 International Conference on Asian Language Processing (IALP), IEEE*, pp. 358–361, 2016.

[16] A. S. Wibawa and A. Purwarianti, "Indonesian Named-entity Recognition for 15 Classes Using Ensemble Supervised Learning", *Procedia Comput Sci*, Vol. 81, pp. 221–228, 2016, doi: https://doi.org/10.1016/j.procs.2016.04.053.

[17] D. C. Wintaka, M. A. Bijaksana, and I. Asror, "Named-Entity Recognition on Indonesian Tweets using Bidirectional LSTM-CRF", *Procedia Comput. Sci.*, Vol. 157, pp. 221–228, 2019, doi: https://doi.org/10.1016/j.procs.2019.08.161.

[18] J. D. Souza and S. Auer, "Computer Science Named Entity Recognition in the Open Research Knowledge Graph", In: *Proc. of From Born-Physical to Born-Virtual: Augmenting Intelligence in Digital Libraries: 24th International Conference on Asian Digital Libraries, ICADL 2022, Hanoi, Vietnam, November 30–December 2, 2022, Proceedings*, Springer, pp. 35–45, 2022, Accessed: Jun. 11, 2023. [Online]. Available: https://arxiv.org/pdf/2203.14579.pdf

[19] N. M. Lai, "LMN at SemEval-2022 Task 11: A Transformer-based System for English Named Entity Recognition", *arXiv Preprint arXiv:2203.03546*, 2022, Accessed: Jun. 11, 2023. [Online]. Available: https://arxiv.org/abs/2203.03546

[20] M. Cho, J. Ha, C. Park, and S. Park, "Combinatorial feature embedding based on CNN and LSTM for biomedical named entity recognition", *J. Biomed Inform.*, Vol. 103, p. 103381, 2020, doi: https://doi.org/10.1016/j.jbi.2020.103381.

[21] C. Che, C. Zhou, H. Zhao, B. Jin, and Z. Gao, "Fast and effective biomedical named entity recognition using temporal convolutional network with conditional random field", *Math. Biosci. Eng*, Vol. 17, pp. 3553–3566, 2020.

[22] J. Deng, L. Cheng, and Z. Wang, "Self-attention-based BiGRU and capsule network for named entity recognition", *arXiv Preprint arXiv:2002.00735*, 2020.

[23] G. H. Ngo, M. Nguyen, and N. F. Chen, "Phonology-Augmented Statistical Framework for Machine Transliteration Using Limited Linguistic Resources", *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, Vol. 27, No. 1, pp. 199–211, 2019, doi: 10.1109/TASLP.2018.2875269.

[24] J. Shi, M. Sun, Z. Sun, M. Li, Y. Gu, and W. Zhang, "Multi-level semantic fusion network for Chinese medical named entity recognition", *J Biomed Inform*, Vol. 133, 2022, doi: 10.1016/j.jbi.2022.104144.

[25] I. Budi and R. R. Suryono, "Application of named entity recognition method for Indonesian datasets: a review", *Bulletin of Electrical Engineering and Informatics*, Vol. 12, No. 2, pp. 969–978, 2023, doi: 10.11591/eei.v12i2.4529.

[26] C. Shorten, T. M. Khoshgoftaar, and B. Furht, "Deep Learning applications for COVID-19", *J. Big Data*, Vol. 8, No. 1, 2021, doi: 10.1186/s40537-020-00392-9.

[27] C. Sun, Z. Yang, L. Wang, Y. Zhang, H. Lin, and J. Wang, "Deep learning with language models improves named entity recognition for PharmaCoNER", *BMC Bioinformatics*, Vol. 22, 2021, doi: 10.1186/s12859-021-04260-y.

[28] S. Moon, S. Chung, and S. Chi, "Bridge damage recognition from inspection reports using NER based on recurrent neural network with active learning", *Journal of Performance of Constructed Facilities*, Vol. 34, No. 6, p. 04020119, 2020.

[29] Y. Wu, M. Jiang, J. Lei, and H. Xu, "Named entity recognition in Chinese clinical text using

deep neural network", *Stud. Health Technol. Inform.*, Vol. 216, p. 624, 2015.

[30] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition", In: *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference*, pp. 260–270, 2016, doi: 10.18653/v1/n16-1030.

[31] V. Sornlertlamvanich and S. Yuenyong, "Thai Named Entity Recognition Using BiLSTM-CNN-CRF Enhanced by TCC", *IEEE Access*, Vol. 10, pp. 53043–53052, 2022, doi: 10.1109/ACCESS.2022.3175201.

[32] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, "Fasttext. zip: Compressing text classification models", *arXiv Preprint arXiv:1612.03651*, 2016.

[33] Z. Zhai, D. Q. Nguyen, and K. Verspoor, "Comparing CNN and LSTM character-level embeddings in BiLSTM-CRF models for chemical and disease named entity recognition", In: *Proc. of the Ninth International Workshop on Health Text Mining and Information Analysis*, Brussels, Belgium, pp. 38–43, 2018, doi: 10.18653/v1/W18-5605.

[34] A. Vaswani *et al.*, "Attention is all you need", *Adv Neural Inf Process Syst*, Vol. 30, 2017.

[35] D. H. Fudholi, R. A. N. Nayoan, A. F. Hidayatullah, and D. B. Arianto, "A Hybrid CNN-BiLSTM Model for Drug Named Entity Recognition", *Journal of Engineering Science and Technology*, Vol. 17, No. 1, pp. 730–744, 2022, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85124628532&partnerID=40&md5=6ad3355a09367e9606d12bfcdcc3dedf

[36] S. O. Khairunnisa, A. Imankulova, and M. Komachi, "Towards a standardized dataset on Indonesian named entity recognition", In: *Proc. 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: Student Research Workshop*, pp. 64–71, 2020.

[37] G. B. Herwanto and D. P. Dewantara, "Traffic Condition Information Extraction from Twitter Data", In: *Proc. of 2nd International Conference on Electrical Engineering and Informatics, ICELTICs 2018*, pp. 95–100, 2018, doi: 10.1109/ICELTICS.2018.8548921.

[38] B. Wilie *et al.*, "IndoNLU: Benchmark and resources for evaluating Indonesian natural language understanding", *arXiv Preprint arXiv:2009.05387*, 2020.

[39] X. Gao and Q. Li, "Named entity recognition in material field based on Bert-BILSTM-Attention-CRF", In: *Proc. of 2021 IEEE Conference on Telecommunications, Optics and Computer Science (TOCS)*, pp. 955–958, 2021, doi: 10.1109/TOCS53301.2021.9688665.

[40] L. Xiao, G. Wang, and Y. Zuo, "Research on patent text classification based on word2vec and LSTM", In: *Proc. of 2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, pp. 71–74, 2018

[41] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification", In: *Proc. of the 2016 conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1480–1489, 2016.

[42] L. L. G. Q. Zhou, "An End-to-End Entity and Relation Extraction Network with Multi-Head Attention", *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, 2018, doi: 10.1007/978-3-030-01716-3_12.

[43] Y. Du, B. Pei, X. Zhao, and J. Ji, "Deep scaled dot-product attention based domain adaptation model for biomedical question answering", *Methods*, Vol. 173, pp. 69–74, 2020, doi: https://doi.org/10.1016/j.ymeth.2019.06.024.

[44] D. A. K. Khotimah and R. Sarno, "Sentiment analysis of hotel aspect using probabilistic latent semantic analysis, word embedding and LSTM", *International Journal of Intelligent Engineering and Systems*, Vol. 12, No. 4, pp. 275–290, 2019, doi: 10.22266/ijies2019.0831.26.

[45] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation", In: *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.

[46] R. Řehůřek and P. Sojka, "Gensim—statistical semantics in python", *Retrieved from Genism. Org*, 2011.

[47] F. Dugas and E. Nichols, "DeepNNNER: Applying BLSTM-CNNs and Extended Lexicons to Named Entity Recognition in Tweets", In : *Proc. of the 2nd Workshop on Noisy User-generated Text ({WNUT})*, pp. 178–187, 2016, [Online]. Available: https://www.aclweb.org/anthology/W16-3924

[48] X. Ma and E. Hovy, "End-to-end sequence labeling via bi-directional lstm-cnns-crf", *arXiv Preprint arXiv:1603.01354*, 2016.

[49] J. Wang, W. Xu, X. Fu, G. Xu, and Y. Wu, "ASTRAL: Adversarial Trained LSTM-CNN for Named Entity Recognition", *Knowl Based Syst*, Vol. 197, p. 105842, 2020, doi: https://doi.org/10.1016/j.knosys.2020.105842.

[50] Z. H., H. W., and Z. Y, "FlexNER: A Flexible LSTM-CNN Stack Framework for Named Entity Recognition", *Natural Language Processing and Chinese Computing. NLPCC 2019. Lecture Notes in Computer Science*, T. J., K. MY., Z. D., L. S., and Z. H, Eds., Cham: Springer. doi: 10.1007/978-3-030-32236-6_14.

[51] S. M. E. Abyad, M. M. Soliman, and K. M. E. Sayed, "Deep Video Hashing Using 3DCNN with BERT", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 5, pp. 113–127, Oct. 2022, doi: 10.22266/ijies2022.1031.11.

[52] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data", In: *Proc. of the Eighteenth International Conference on Machine Learning*, San Francisco, CA, USA, pp. 282–289, 2011.

[53] L. Ratinov and D. Roth, "Design challenges and misconceptions in named entity recognition", In: *Proc. of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pp. 147–155, 2009.

[54] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization", *Journal of Machine Learning Research*, Vol. 13, No. 2, 2012.

[55] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms", *Adv Neural Inf Process Syst*, Vol. 25, 2012.

[56] G. Popovski, B. K. Seljak, and T. Eftimov, "A Survey of Named-Entity Recognition Methods for Food Information Extraction", *IEEE Access*, Vol. 8, pp. 31586–31594, 2020, doi: 10.1109/ACCESS.2020.2973502.

[57] Warto, Muljono, Purwanto, and E. Noersasongko, "Capitalization Feature and Learning Rate for Improving NER Based on RNN BiLSTM-CRF", In: *Proc. of 2022 IEEE International Conference on Cybernetics and Computational Intelligence, CyberneticsCom 2022*, 2022, doi: 10.1109/CyberneticsCom55287.2022.9865660.

[58] L. N. Smith, "Cyclical learning rates for training neural networks", In: *Proc. of 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE*, pp. 464–472, 2017.