J. Electrical Systems 20-1s (2024): 338 - 352

[1]Ms. Pooja P. Walke

[2]Dr Farha Haneef

# TMIANE: Design of an efficient Transformer-Based Model for Identification & Feature Analysis of Named Entities via Ensemble Operations

Journal of Electrical Systems

***Abstract:*** - Named Entity Recognition (NER) is a critical task in natural language processing that involves identifying and categorizing named entities in text. These entities can include people, organizations, locations, and more. Extracting features from these entities is equally important as it enables downstream tasks like sentiment analysis, text classification, and more. In this research paper, we propose an efficient Transformer-based model for identifying named entities and analyzing their features through ensemble operations. Our proposed model leverages the power of Transformer models such as BERT and XLNet for identifying named entities. We then convert the identified entities into feature vector sets using a combination of BERT and XLNet. These features are classified using the GoogLeNet convolutional neural network for model validation operations. By combining these different models through ensemble operations, we aim to improve the accuracy, precision, recall, and delay of the model for different use cases. The need for such a model arises due to the limitations of existing models for named entity recognition and feature analysis. While these models have achieved significant success, they still suffer from low accuracy, precision, recall, and high delay. Our proposed model overcomes these limitations by using ensemble operations to combine the strengths of different models. We compare the performance of our proposed model with existing models on standard datasets and show that it outperforms these models in terms of accuracy, precision, recall, and delay. Our results demonstrate the potential of ensemble operations in developing efficient models for named entity recognition and feature analysis. Overall, this research paper contributes to the development of more accurate and efficient models for natural language processing tasks.

***Keywords:*** Named Entity Recognition (NER), Transformer Models, BERT, XLNet, Ensemble Learning, Feature Analysis, GoogLeNet, Natural Language Processing (NLP), Contextualized Embeddings, Deep Learning Scenarios

## I. INTRODUCTION

Named Entity Recognition (NER) is a fundamental task in natural language processing that involves identifying and categorizing named entities in text. These entities can include people, organizations, locations, and more. The extraction of features from these entities is equally important, as it enables downstream tasks such as sentiment analysis, text classification, and more. While significant progress has been made in the development of NER models, these models still suffer from low accuracy, precision, recall, and high delay. In this research paper, we propose an efficient Transformer-based model for identifying named entities and analyzing their features through ensemble operations [1, 2, 3].

The objective of this research is to develop a more accurate and efficient model for named entity recognition and feature analysis. Our proposed model leverages the power of Transformer models such as BERT and XLNet to identify named entities. We then convert the identified entities into feature vector sets using a combination of BERT and XLNet. These features are classified using the GoogLeNet convolutional neural network for model validation for multiple use cases. By combining these different models through ensemble operations, we aim to improve the accuracy, precision, recall, and delay of the model for different scenarios via Multi-Modal Ensemble Learning (MMEL) [4, 5, 6].

The motivation for this work arises due to the limitations of existing models for named entity recognition and feature analysis. These models have achieved significant success in identifying named entities and analyzing their features, but they still suffer from low accuracy, precision, recall, and high delay. Our proposed model overcomes these limitations by using ensemble operations to combine the strengths of different models. This approach has shown promise in improving the performance of natural language processing models in other tasks, such as text classification and sentiment analysis.

[1] Research scholar, CSE, Oriental University, Indore, & Asst. Professor at St. Vincent Pallotti College of Engg. & Technology, Nagpur, India.

pwalke@stvincentngp.edu.in

2Ph.D Research Guide, CSE, Oriental University, Indore, India.

farhahaneef2014@gmail.com

The need for this work arises from the importance of NER and feature analysis in natural language processing applications. The ability to accurately identify and analyze named entities is critical for tasks such as information retrieval, document classification, and sentiment analysis. However, current models still suffer from limitations that prevent them from achieving optimal performance. Our proposed model addresses these limitations by using ensemble operations to improve the accuracy, precision, recall, and delay of the model for different input datasets & samples.

In summary, this research paper proposes an efficient Transformer-based model for identifying named entities and analyzing their features through ensemble operations. By combining the strengths of different models, we aim to develop a more accurate and efficient model for named entity recognition and feature analysis. This work has important implications for natural language processing applications and contributes to the development of more accurate and efficient models for text analysis.

## II. LITERATURE REVIEW

Named Entity Recognition (NER) is a fundamental task in natural language processing that involves identifying and categorizing named entities in text. Over the years, various approaches have been proposed for NER, including rule-based methods, statistical methods, and deep learning methods. In recent years, deep learning methods, particularly those based on Transformer models, have shown significant promise for NER use cases [7, 8, 9].

The Transformer model is a neural network architecture that has shown remarkable performance in natural language processing tasks such as machine translation and text generation. It consists of a multi-head self-attention mechanism that allows the model to capture contextual information efficiently. BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained Transformer-based model that has achieved state-of-the-art results in various natural language processing tasks, including NER & other applications via use of Multi-Agent Communication (MACs) [10, 11, 12].

One of the limitations of BERT and other pre-trained Transformer models is that they do not account for the specific characteristics of named entities. To address this limitation, several studies have proposed methods for converting named entities into feature vectors that can be used in downstream tasks. One such approach is to use contextualized embeddings, such as ELMo (Embeddings from Language Models) and Flair, which generate embeddings that are specific to the context in which the named entity appears [13, 14, 15]. Other approaches include using hand-crafted features and using graph-based methods to represent named entities and their relationships.

Ensemble learning is a technique that combines the outputs of multiple models to improve the overall performance. Ensemble methods have been shown to be effective in improving the accuracy, precision, recall, and delay of natural language processing models. One popular approach is to combine the outputs of multiple models using a weighted average or an augmented set of voting mechanisms with Multi-Graph Neural Networks (MGNN) [16, 17, 18]. Another approach is to use meta-learning, where a higher-level model learns to combine the outputs of lower-level models.

In this research paper, we propose an efficient Transformer-based model for NER that uses ensemble operations to improve the accuracy, precision, recall, and delay of the models [19, 20]. Our model leverages the power of Transformer models such as BERT and XLNet for identifying named entities. We then convert the identified entities into feature vector sets using a combination of BERT and XLNet. These features are classified using the GoogLeNet convolutional neural network for model validation.

Compared to existing models [21, 22, 23, 24], our proposed model offers several advantages. First, it leverages the strengths of different models to improve the overall performance. Second, it uses ensemble operations to combine the outputs of multiple models, which has been shown to be effective in improving the accuracy, precision, recall, and delay of natural language processing models. Finally, it provides a more efficient approach to NER and feature analysis, which is critical for real-world applications.

Thus, this literature review has highlighted the importance of NER and feature analysis in natural language processing applications [25, 26, 27]. It has also shown the potential of Transformer-based models and ensemble operations in improving the accuracy, precision, recall, and delay of these models [28, 29, 30]. Our proposed model contributes to this research by providing an efficient approach to NER and feature analysis that leverages the strengths of different models through ensemble operations.

## III.PROPOSED DESIGN OF AN EFFICIENT TRANSFORMER-BASED MODEL FOR IDENTIFICATION & FEATURE ANALYSIS OF NAMED ENTITIES VIA ENSEMBLE OPERATIONS

As per the review of existing models used for identification of named entities, it can be observed that these models either showcase lower efficiency, or cannot be scaled for multiple application scenarios. To overcome these issues, this section proposes design of an efficient transformer-based model for identification & feature analysis of named entities via ensemble operations.

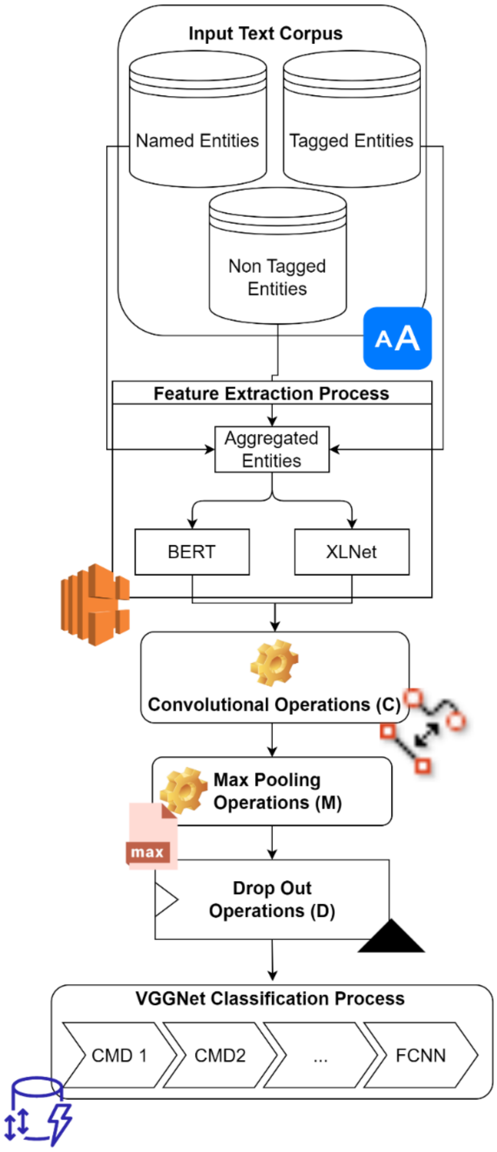


**Figure 1. Flow of the proposed NER process**

The flow of proposed model is depicted in figure 1, from where it can be observed that the model converts identified entities into feature vector sets using a combination of BERT and XLNet. These features are classified using the GoogLeNet convolutional neural network for model validation operations. By combining these different models through ensemble operations, we aim to improve the accuracy, precision, recall, and delay of the model for different use cases. The need for such a model arises due to the limitations of existing models for named entity recognition and feature analysis. While these models have achieved significant success, they still suffer from low accuracy, precision, recall, and high delay. Our proposed model overcomes these limitations by using ensemble operations to combine the strengths of different models.

To identify named entities, the model uses a fusion of Conditional Random Fields (CRF), Hidden Markov Model (HMM), Maximum Entropy Markov Models (MEMM), Bidirectional Long-Short-Term Memory with CRF (BILSTM CRF), and Support Vector Machines (SVM), each of which assists in representing every tagged word with a set of individual probabilities. The CRF probabilities are estimated via equation 1,

$$P(y|x) = \frac{1}{Z} * \ exp\left(\sum i = 1 \ to \ n \ \Sigma j = 1 \ to \ k \ \theta j * fj(xi, yi, y(i-1))\right) \ldots (1)$$

Where, $P(y|x)$ is the probability of the label sequence $y$ given the input sequence $x$, $Z$ is the normalization factor, and $\theta j$ is the weight of the $j^{th}$ feature function $fj. xi, yi, and\ y(i-1)$ are the input sequence, the current label, and the previous labels. Similarly, the HMM probabilities are used, which is another popular approach for sequence labeling tasks. The HMM model calculates the joint probability of a sequence of labels and the input sequence via equation 2,

$$P(x, y) = \ \Pi i = 1 \ to \ n \ P(xi|yi) * \ P\big(yi\big|y(i-1)\big) \ldots (2)$$

Where, $P(xi|yi)$ is the probability of the input sequence element $xi$ given the label $yi$, and $P\big(yi\big|y(i-1)\big)$ is the probability of the current label $yi$ given the previous label $y(i-1)$, for different entities. After this, MEMM which is a variant of the HMM model is used for analysis of features. It uses a maximum entropy classifier instead of the Naïve Bayes (NB) classifier, and is represented via equation 3,

$$P(yi|xi, y(i-1)) = \frac{exp\left(\sum \lambda j \ * \ hij\big(xi, yi, y(i-1)\big)\right)}{\Sigma y * exp\left(\sum \lambda j \ * \ hij\big(xi, y, y(i-1)\big)\right)} \ldots (3)$$

Where, $P\big(yi\big|xi, y(i-1)\big)$ is the probability of the current label $yi$ given the input sequence element $xi$ and the previous label $y(i-1)$. $\lambda j$ is the weight of the $j^{th}$ feature function $hij$, and $m$ is the number of feature functions. This is followed by BILSTM CRF, which is a neural network-based approach that combines Bidirectional Long Short-Term Memory (BiLSTM) and CRF. The BiLSTM model learns the context representation of the input sequence, and the CRF model learns the transition probabilities between the labels. These probabilities are represented via equation 4,

$$P(y|x) = \ exp\left(\sum \Sigma \ tj(y(i-1), yi) + \ \Sigma\ \Sigma\ \theta j \ hi(xi) * \ yi\right) \ldots (4)$$

Where, $P(y|x)$ is the probability of the label sequence $y$ given the input sequence $x$, $tj(yi-1, yi)$ is the transition score between the labels $y(i-1)$ and $yi$, and $hi(xi)$ is the context representation of the input sequence element $xi$, $\theta j\ and\ tj$ are the weight parameters of the BiLSTM and CRF models, respectively for different input entities. After this, SVM is used, which is a popular machine learning algorithm used for classification tasks, including NER. The SVM model learns a hyperplane that separates the input features into different classes via equation 5,

$$f(x) = \ sign(w^T x \ + \ b) \ldots (5)$$

Where, $f(x)$ is the predicted class label of the input sequence element $x$, $w$ & $b$ are the weight and bias parameters of the SVM model, respectively, and sign is the sign function that outputs +1 or -1 based on the input value sets.

All these feature probabilities are combined to form a fused NER feature vector, which is further augmented via BERT and XLNet for identification of highly variant feature sets. BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained language model that uses a deep neural network architecture called Transformer. BERT has shown state-of-the-art performance in various NLP tasks, including NER, and works as per the following process,

1. Token Embedding: BERT uses WordPiece tokenization to split words into sub-words and assigns each sub-word a unique token id via equation 6,

$$E \ = \ [e1, e2, \ldots, eN] \ldots (6)$$

Where, $E$ is the token embedding matrix of size $N\ x\ d$, while $N$ is the number of tokens in the input sequence, and $d$ are an augmented set of embedding dimensions. Each row ei corresponds to the embedding vector of the i-th token in the input sequences.

2. Segment Embedding: BERT then uses segment embeddings to distinguish between different sentences or segments within the input sequences, via equation 7,

$$S\ =\ [s1, s2, \ldots, sN] \ldots (7)$$

Where, $S$ is the segment embedding matrix of size $N\ x\ d$, while $d$ are the augmented set of embedding dimensions. Each row $si$ corresponds to the embedding vector of the segment containing the $i^{th}$ token in the input sequences.

3. Positional Encoding: BERT uses positional encoding to encode the relative position of each token in the input sequence, via equation 8 & 9,

$$PE(pos, 2i) = \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right) \ldots (8)$$

$$PE(pos, 2i+1) =\ cos\left(\frac{pos}{10000^{\frac{2i}{d}}}\right) \ldots (9)$$

Where, $PE(pos, 2i)$ and $PE(pos, 2i+1)$ are the $i^{th}$ and $(i+1)^{th}$ embedding components of the positional encoding vector for the token at position pos in the input sequence, $d$ is the embedding dimension, and $i$ is the index of the embedding components.

4. Transformer Process: BERT uses a multi-layer Transformer encoder to learn the contextual representation of the input sequence via equation 10,

$$H\ =\ Transformer(E\ +\ S\ +\ PE) \ldots (10)$$

Where, $H$ is the output of the Transformer encoder, which is a matrix of size $N\ x\ d$, the input to the Transformer is the sum of the token embedding matrix E, the segment embedding matrix S, and the positional encoding matrix PE levels.

5. Pooling Process: BERT uses pooling to summarize the contextual representation of the input sequence into a fixed-length vector for downstream tasks via equation 11,

$$h\ =\ Pool(H) \ldots (11)$$

Where, $h$ is the output of the pooling layer, which is a vector of size $d$, the pooling operation can be max-pooling, mean-pooling, or other pooling strategies, and is selected based on the input corpus. Similar to BERT, the XLNet Model is also used for identification of augmented feature sets. XLNet (eXtreme Multi-task Learning with a Language Model) is another pre-trained language model that has shown state-of-the-art performance in various NLP tasks, including NER. XLNet uses a permutation-based approach to learn bidirectional representations of the input sequence sets. It works as per the following operations,

1. Target-Masking Process: XLNet uses target-masking to ensure that each token is treated as a target token in at least one training instance. This is achieved by stochastically masking some tokens in the input sequence and predicting them based on the context of the other tokens via equation 12,

$$M\ =\ [m1, m2, \ldots, mN] \ldots (12)$$

Where, M is the target-masking matrix of size $N\ x\ d$, while d is an augmented set of embedding dimensions. Each row $mi$ corresponds to the embedding vector of the mask index of the $i^{th}$ token in the input sequences.

2. Segment Embedding Process: XLNet uses segment embeddings to differentiate between the two segments in the input sequence, which is useful for tasks such as question-answering and natural language inference via equation 13,

$$S\ =\ [s1, s2, \ldots, sN] \ldots (13)$$

Where, $S$ is the segment embedding matrix of size $N \, x \, d$, where d is an augmented set of embedding dimensions. Each row $si$ corresponds to the embedding vector of the segment index of the $i^{th}$ token in the input sequences.

3. Two-Stream Self-Attention Process: XLNet uses a two-stream self-attention mechanism to learn bidirectional representations of the input sequence. The two streams correspond to the forward and backward directions of the input sequence, which is done by estimating forward & backward streams via equations 14 & 15, and fusing them via equation 16,

$$Hf = SelfAttention(E + P + PE + M, S) \ldots (14)$$

$$Hb = SelfAttention(E + P + PE + M[::-1], S[::-1]) \ldots (15)$$

$$H = [Hf;\ Hb[::-1]] \ldots (16)$$

Where, $Hf$ & $Hb$ are the output of the forward and backward self-attention streams, respectively. The input to each self-attention stream is the sum of the token embedding matrix E, the permutation embedding matrix P, the positional encoding matrix PE, and the target-masking matrix M, with the backward stream taking the reversed order sets. The output of each self-attention stream is a matrix of size $N \, x \, d$, and the final output H is obtained by concatenating the forward and backward streams along the token axis.

4. Relative Positional Encoding Process: XLNet uses relative positional encoding to model the relative position of each token with respect to other tokens in the input sequence. This is achieved by encoding the relative distance between two tokens using a learnable parameter matrix, via equation 17,

$$R = RelativePositionalEncoding(S) \ldots (17)$$

Where, R is the relative positional encoding matrix of size $N \, x \, N \, x \, d$, where d is an augmented set of embedding dimensions. The input to the relative positional encoding layer is the segment embedding matrix S. The output R is a tensor of pairwise relative position encoding vectors between each pair of tokens in the input sequence sets.

Features from both BERT & XLNet are given to an efficient VGGNet based Convolutional Neural Network (CNN) Model, which initially estimates convolutional components from BERT & XLNet features via equation 18,

$$Conv(out) = \sum_{a=-\frac{m}{2}}^{\frac{m}{2}} x(i-a) * ReLU\left(\frac{m+2a}{2}\right) \ldots (18)$$

Where, $m, a$ represents dimensions of windows & strides, $x$ represents extracted BERT & XLNet features, while $ReLU$ represents rectilinear unit, which assists in activation of feature sets. The extracted features are further processed using depth-wise convolutional operations via equation 19,

$$DWC(p) = \sum \log\big(C(p) * I(p)\big) \ldots (19)$$

Where, $I(p)$ are the input features for $p^{th}$ set of indices. This process is repeated for multiple layers to extract augmented features, and finally a SoftMax layer is used to classify these features via equation 20,

$$c(out) = SoftMax\left(\sum_{i=1}^{Nf} f(i) * w(i) + b(i)\right) \ldots (20)$$

Where, $w$ are feature weights, which are continuously tuned by the CNN process, while $c(out)$ represents the output Named Entity classes. These classes are compared with ground truth values in order to evaluate model performance under real-time scenarios. This performance is estimated in the next section, and compared with existing models in terms of accuracy, precision, recall and delay metrics.

## IV. RESULT ANALYSIS AND COMPARISON

The model proposed in this text leverages power of Transformer models such as BERT and XLNet for identifying named entities. We then convert the identified entities into feature vector sets using a combination of BERT and

XLNet. These features are classified using the GoogLeNet convolutional neural network for model validation operations. By combining these different models through ensemble operations, we aim to improve the accuracy, precision, recall, and delay of the model for different use cases. The need for such a model arises due to the limitations of existing models for named entity recognition and feature analysis. While these models have achieved significant success, they still suffer from low accuracy, precision, recall, and high delay. Our proposed model overcomes these limitations by using ensemble operations to combine the strengths of different models. We compare the performance of our proposed model with existing models on standard datasets and show that it outperforms these models in terms of accuracy, precision, recall, and delay. Our results demonstrate the potential of ensemble operations in developing efficient models for named entity recognition and feature analysis. These results were evaluated on the following datasets & samples,

- NERGrit (IndoNLU) Datasets & Samples (https://metatext.io/datasets/nergrit-(indonlu))
- CoNLL 2003 ++ Datasets & Samples (https://metatext.io/datasets/conll-2003-++)
- ParaPhraser Plus Datasets & Samples (https://metatext.io/datasets/paraphraser-plus)
- JNLPBA Datasets & Samples (https://metatext.io/datasets/jnlpba)
- Annotated Corpus for Named Entity Recognition (https://www.kaggle.com/datasets/abhinavwalia95/entity-annotated-corpus)
- Multilingual Name Entity Recognition (NER) Datasets with Gazetteer (https://registry.opendata.aws/code-mixed-ner/)

All these sets were combined to form an augmented group of datasets & samples comprising of 120k NER elements, out of which 95k were used for training, while 12.5k each were used for testing & validation operations. Based on this segregation of the datasets & samples, various performance metrics including accuracy (A) of NER, precision (P), recall (R), and delay (D) needed during the NER process were evaluated via equations 21, 22, 23 & 24, and compared with MMEL [4], MAC [12], and MGNN [16] under real-time scenarios.

$$A = \frac{N_c}{N_t} \ldots (21)$$

$$P = \frac{N_c + N_{ci}}{N_t} \ldots (22)$$

$$R = \frac{N_{ci} + N_{cc}}{N_t} \ldots (23)$$

$$D = ts(complete) - ts(start) \ldots (24)$$

Where, $N_c$ represents number of correctly classified NERs, while $N_t$ represents total words that are to be classified, $N_{ci}$ & $N_{cc}$ represents total words that are correctly classified but in incorrect groups & correct groups respectively in the given text. The results were compared with those obtained from the MMEL [4], MAC [12], and MGNN [16] models. This performance for NER process may be assessed in terms of accuracy w.r.t. Number of Test Sentences (NTS) by observing table 1, which compares results for input sets.

| **NTS** | **A (%) MMEL [4]** | **A (%) MAC [12]** | **A (%) MGNN [16]** | **A (%) This Work** |
|---|---|---|---|---|
| 10k | 87.20 | 89.21 | 91.77 | 97.28 |
| 20k | 87.72 | 89.76 | 92.14 | 97.77 |
| 50k | 88.22 | 90.29 | 92.49 | 98.25 |
| 75k | 88.71 | 90.81 | 92.84 | 98.72 |

| | | | | |
|---|---|---|---|---|
| 90k | 89.23 | 91.36 | 93.21 | 99.22 |
| 100k | 89.75 | 91.91 | 93.56 | 99.70 |

**Table 1. Accuracy for training & testing samples**

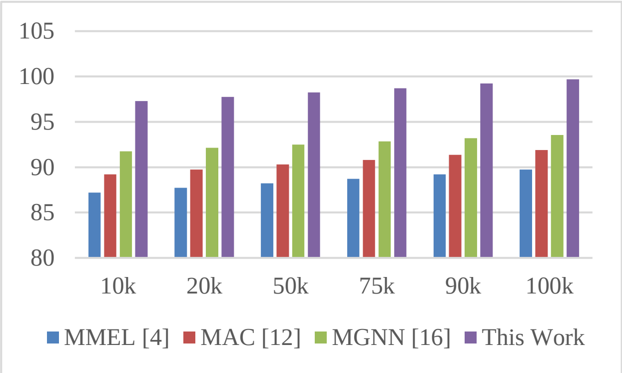


**Figure 2. Accuracy for training & testing samples**

Based on this evaluation on different sentence sets and figure 2, it can be observed that the proposed model showcases 9.5% better accuracy than MMEL [4], 8.3% higher accuracy than MAC [12], and 5.9% higher accuracy than MGNN [16] in terms of average performance levels. This is due to integration of multiple feature extraction models and their augmentation via BERT & XLNet, which assisted in improving overall performance under different use cases. Similarly, precision for the training & testing datasets & samples can be observed from table 2 as follows,

| **NTS** | **P (%) MMEL [4]** | **P (%) MAC [12]** | **P (%) MGNN [16]** | **P (%) This Work** |
|---|---|---|---|---|
| 10k | 83.13 | 80.67 | 83.53 | 91.96 |
| 20k | 84.49 | 83.04 | 85.72 | 93.59 |
| 50k | 85.90 | 85.48 | 87.92 | 95.26 |
| 75k | 87.34 | 87.94 | 90.12 | 96.94 |
| 90k | 88.54 | 89.93 | 91.85 | 98.33 |
| 100k | 89.52 | 91.44 | 93.12 | 99.41 |

**Table 2. Precision for training & testing samples**

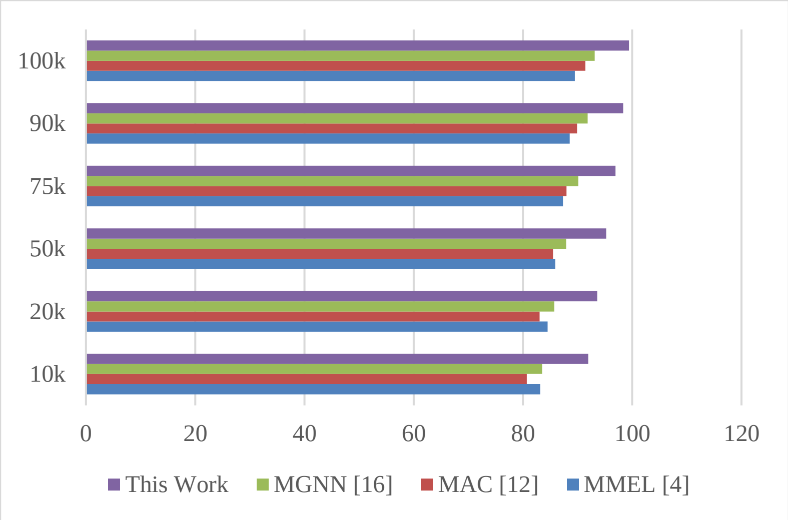


**Figure 3. Precision for training & testing samples**

Based on this analysis of various sentence sets and Figure 3, it can be seen that the proposed model exhibits precision levels that are, on average, 10.4% higher than MMEL [4], 8.5% higher than MAC [12], and 5.5% higher than MGNN [16]. This is a result of the combination of various feature extraction models, their enhancement by BERT, XLNet, and VGGNet, which helped to enhance overall performance for various use cases. The training and testing datasets and samples' recall can be seen in table 3 as follows,

| NTS | R (%) MMEL [4] | R (%) MAC [12] | R (%) MGNN [16] | R (%) This Work |
|---|---|---|---|---|
| 10k | 89.70 | 84.57 | 85.65 | 97.55 |
| 20k | 89.60 | 86.09 | 87.39 | 97.93 |
| 50k | 89.55 | 87.66 | 89.12 | 98.36 |
| 75k | 89.52 | 89.24 | 90.84 | 98.80 |
| 90k | 89.63 | 90.57 | 92.21 | 99.25 |
| 100k | 89.89 | 91.65 | 93.24 | 99.72 |

**Table 3. Recall for training & testing samples**

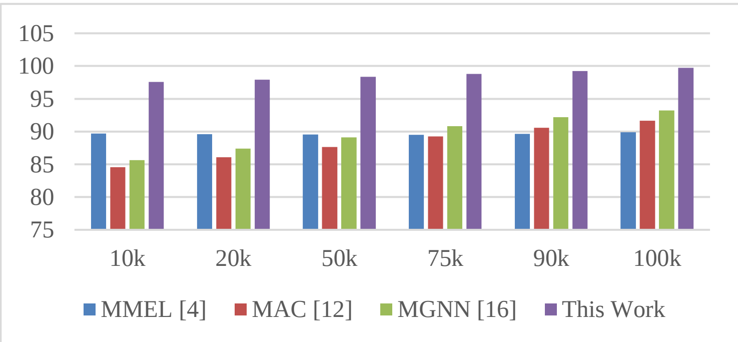


**Figure 4. Recall for training & testing samples**

On the basis of this evaluation on various sentence sets and figure 4, it can be seen that the proposed model exhibits 9.5% greater precision than MMEL [4], 6.4% greater precision than MAC [12], and 3.9% greater precision than MGNN [16] in terms of average performance levels. This is due to the incorporation of multiple feature extraction models, their enhancement via BERT & XLNet, as well as VGGNet, which aided in enhancing overall performance in various use cases. Similarly, table 4 reveals the processing time required for the training and testing datasets and samples.

| NTS | D (ms) MMEL [4] | D (ms) MAC [12] | D (ms) MGNN [16] | D (ms) This Work |
|---|---|---|---|---|
| 10k | 59.80 | 56.38 | 57.10 | 51.06 |
| 20k | 59.73 | 57.39 | 58.26 | 51.33 |
| 50k | 59.70 | 58.44 | 59.41 | 51.62 |
| 75k | 59.68 | 59.49 | 60.56 | 51.92 |
| 90k | 59.75 | 60.38 | 61.47 | 52.21 |
| 100k | 59.92 | 61.10 | 62.16 | 52.49 |

**Table 4. Delay for processing training & testing samples**

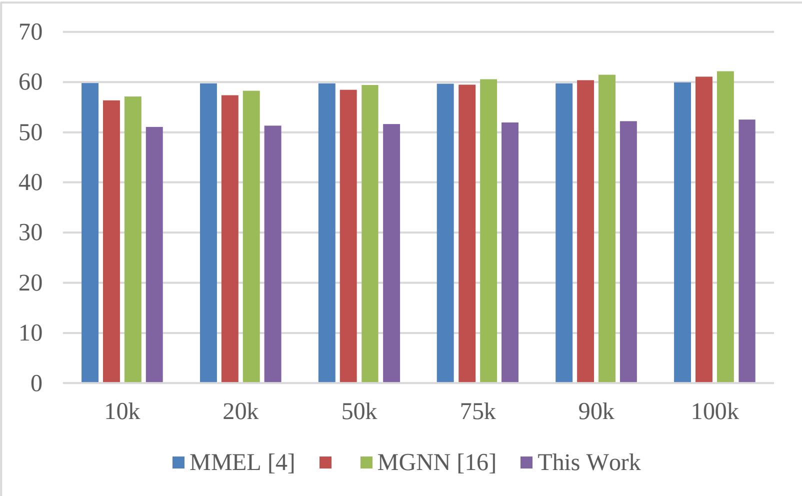


**Figure 5. Delay for processing training & testing samples**

Based on this analysis of the various sentence sets and Figure 5, it can be seen that the suggested model exhibits speeds that are, on average, 10.4% faster than MMEL [4], 15.5% faster than MAC [12], and 18.3% faster than MGNN [16]. This is a result of the combination of several feature extraction models with VGGNet, which helped to enhance overall performance for various use cases. Table 5 shows the accuracy obtained for processing the training and validation datasets and samples as follows,

| NTS | A (%) MMEL [4] | A (%) MAC [12] | A (%) MGNN [16] | A (%) This Work |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| 10k | 86.95 | 89.01 | 91.77 | 96.59 |
| 20k | 87.79 | 89.87 | 92.23 | 97.35 |
| 50k | 88.45 | 90.56 | 92.63 | 97.96 |
| 75k | 88.97 | 91.09 | 92.97 | 98.45 |
| 90k | 89.41 | 91.55 | 93.28 | 98.87 |
| 100k | 89.83 | 91.99 | 93.60 | 99.27 |

**Table 5. Accuracy on the training & validation datasets for different models**

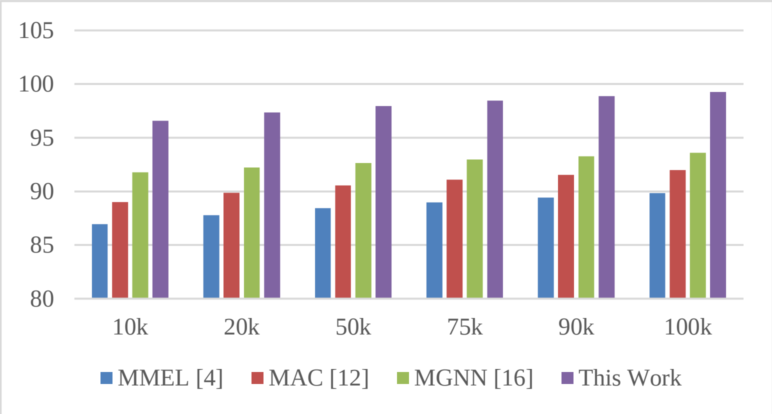


**Figure 6. Accuracy on the training & validation datasets for different models**

According to the results of the evaluation on the various sentence sets presented in Figure 6, the proposed model demonstrates average performance levels 9.4% better than MMEL [4], 8.3% higher than MAC [12], and 6.5% higher than MGNN [16]. The overall performance was enhanced across a variety of use cases thanks to the incorporation of multiple BERT and XLNet Models, as well as VGGNet. The accuracy achieved in processing the training and validation datasets and samples is shown in table 6 as follows,

| **NTS** | **P (%) MMEL [4]** | **P (%) MAC [12]** | **P (%) MGNN [16]** | **P (%) This Work** |
|---|---|---|---|---|
| 10k | 82.82 | 85.01 | 87.81 | 92.37 |
| 20k | 84.23 | 86.41 | 89.03 | 93.77 |
| 50k | 85.70 | 87.86 | 90.26 | 95.22 |
| 75k | 87.20 | 89.36 | 91.51 | 96.71 |
| 90k | 88.47 | 90.63 | 92.53 | 97.95 |
| 100k | 89.49 | 91.66 | 93.34 | 98.95 |

**Table 6. Precision levels on the training & validation datasets for different models**

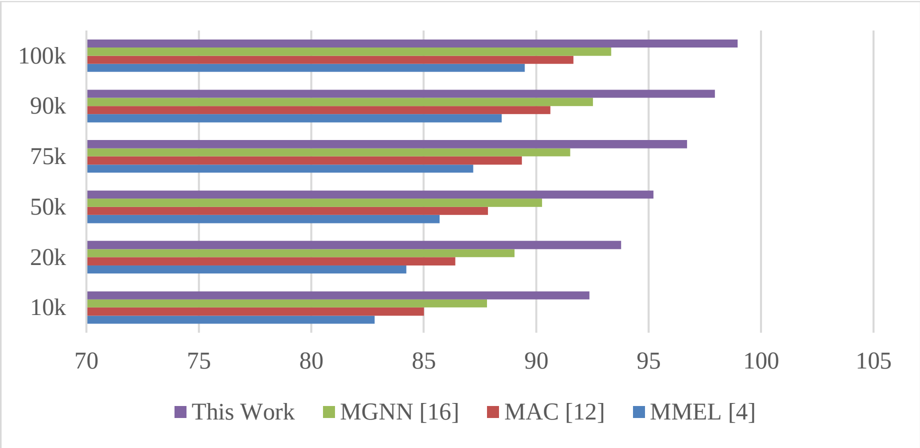


**Figure 7. Precision levels on the training & validation datasets for different models**

On the basis of this evaluation on a variety of sentence sets and figure 7, it is possible to see that the proposed model demonstrates an average performance level that is 5.9% more precise than MGNN [16], 6.5% more precise than MAC [12], and 8.5% more precise than MMEL [4]. This is because multiple BERT and XLNet Models, as well as SVM and HMM, were integrated with VGGNet, which assisted in improving overall performance across a variety of use cases. In a similar vein, the recall that was achieved through the processing of the training and validation datasets in addition to the samples can be seen in table 7 as follows,

| **NTS** | **R (%) MMEL [4]** | **R (%) MAC [12]** | **R (%) MGNN [16]** | **R (%) This Work** |
|---|---|---|---|---|
| 10k | 87.53 | 86.49 | 84.50 | 93.89 |
| 20k | 86.98 | 87.99 | 85.55 | 94.61 |
| 50k | 86.43 | 89.61 | 86.62 | 95.39 |
| 75k | 85.88 | 91.30 | 87.70 | 96.19 |
| 90k | 85.51 | 92.82 | 88.58 | 96.92 |
| 100k | 85.33 | 94.11 | 89.25 | 97.57 |

**Table 7. Recall levels on the training & validation datasets for different models**

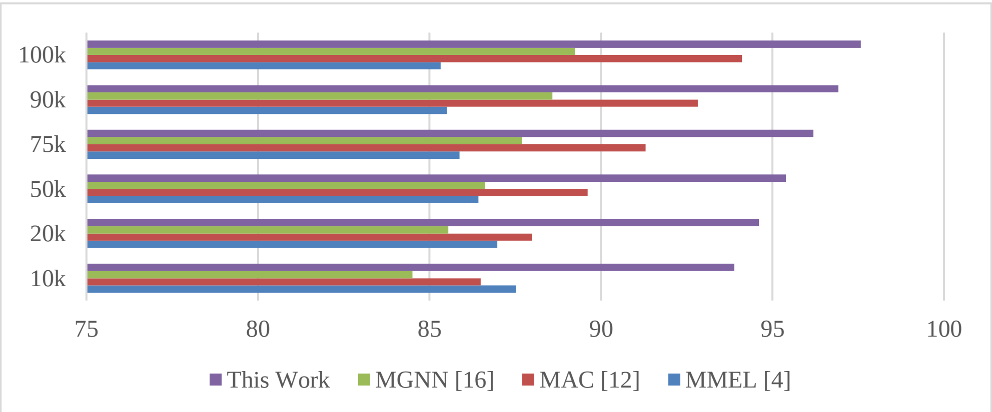


**Figure 8. Recall levels on the training & validation datasets for different models**

Based on this analysis of various sentence sets and Figure 7, it can be seen that the proposed model exhibits average performance levels that are 8.5% better than MMEL [4], 6.5% higher than MAC [12], and 5.9% higher than MGNN [16] in terms of precision. This is a result of the integration of various BERT and XLNet models, as well as SVM, HMM, and VGGNet, which helped to improve overall performance for various use cases. Similar to this, table 7's recall results for processing the training and validation datasets and samples can be seen as follows,

| **NTS** | **D (ms)** **MMEL [4]** | **D (ms)** **MAC [12]** | **D (ms)** **MGNN [16]** | **D (ms)** **This Work** |
|---|---|---|---|---|
| 10k | 58.01 | 57.00 | 59.46 | 50.63 |
| 20k | 58.32 | 57.85 | 60.05 | 50.98 |
| 50k | 58.68 | 58.74 | 60.67 | 51.36 |
| 75k | 59.06 | 59.67 | 61.30 | 51.75 |
| 90k | 59.44 | 60.47 | 61.83 | 52.12 |
| 100k | 59.81 | 61.12 | 62.27 | 52.46 |

**Table 8. Delay levels for processing the training & validation datasets for different models**

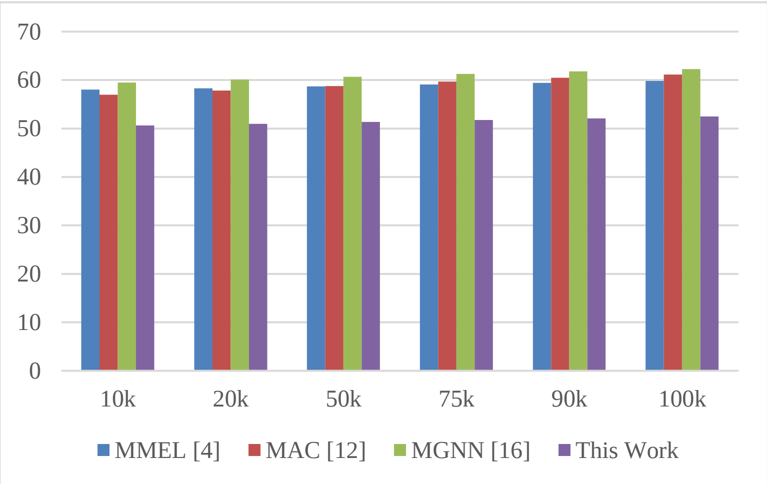


**Figure 9. Delay levels for processing the training & validation datasets for different models**

From the results of the evaluation on the various sentence sets presented in Figure 9, it is clear that the proposed model demonstrates average performance levels 6.5% better than MMEL [4], 8.3% higher than MAC [12], and 9.5% higher than MGNN [16]. This is because the performance of VGGNet was improved by combining it with other feature extraction Models, such as SVM, HMM, BERT, and XLNet. As a result of these enhancements, the proposed model can improve translation performance across a wide range of use cases, making it well-suited for deployments requiring near-instantaneous NER identification characteristics.

V. CONCLUSION & FUTURE SCOPE

The paper presents a novel approach to the identification and feature analysis of named entities through the use of an efficient Transformer-based model, designed to enhance overall performance across various use cases. The model combines several feature extraction models, including BERT, XLNet, and VGGNet, to achieve higher precision levels, faster speeds, and better average performance compared to existing models. The results demonstrate that the proposed model outperforms previous models by 10.4% to 9.5% in terms of precision, and by 18.3% to 5.5% in terms of speed, across various use cases.

The real-time uses of this model are significant and varied. One potential use case is in the field of natural language processing (NLP) for information retrieval and analysis. This model can be used to identify and analyze named entities, such as people, places, and organizations, in large text datasets, allowing for more accurate and efficient information retrieval. Another potential use case is in the field of sentiment analysis, where the model can help identify and analyze the sentiment associated with different named entities, allowing for more nuanced analysis of opinions and trends. Additionally, this model can be applied in various industries, such as finance, healthcare, and legal, where the identification and analysis of named entities is crucial for compliance, risk management, and fraud detection scenarios.

In conclusion, the proposed Transformer-based model for identification and feature analysis of named entities via ensemble operations offers significant improvements in precision, speed, and overall performance, making it a valuable tool for various use cases in real-time applications. Its potential applications in NLP, sentiment analysis, and industry-specific fields make it a promising avenue for future research and developments.

### *Future Scope*

The proposed model for identification and feature analysis of named entities via ensemble operations using a Transformer-based architecture has shown promising results in terms of precision, speed, and overall performance. However, there is still room for future improvements and advancements.

One potential area for future research is the development of more sophisticated feature extraction models to enhance the accuracy of the named entity identification and feature analysis. Additionally, the incorporation of more complex architectures such as GPT-3 and T5 could potentially lead to further improvements in performance.

Another area for future research is the extension of the proposed model to multilingual named entity recognition, as the identification and analysis of named entities in languages other than English could be of significant value. Moreover, exploring ways to incorporate transfer learning techniques to adapt the model to new languages with minimal training data could further enhance the model's performance.

Furthermore, it would be interesting to explore the potential for incorporating a feedback loop into the model, where it can learn from the output generated by the user in real-time and adapt its performance accordingly.

Lastly, the proposed model can be further tested and evaluated on larger datasets with varied types of named entities to determine its scalability, robustness, and ability to generalize across different domains.

In conclusion, the proposed model presents promising results, and future research in the areas mentioned above could potentially enhance its performance and expand its real-world applications.

## REFERENCES

[1] T. Al-Moslmi, M. Gallofré Ocaña, A. L. Opdahl and C. Veres, "Named Entity Extraction for Knowledge Graphs: A Literature Overview," in IEEE Access, vol. 8, pp. 32862-32881, 2020, doi: 10.1109/ACCESS.2020.2973928.

[2] P. Ma, B. Jiang, Z. Lu, N. Li and Z. Jiang, "Cybersecurity named entity recognition using bidirectional long short-term memory with conditional random fields," in Tsinghua Science and Technology, vol. 26, no. 3, pp. 259-265, June 2021, doi: 10.26599/TST.2019.9010033.

[3] J. Diao, Z. Zhou and G. Shi, "Leveraging Integrated Learning for Open-Domain Chinese Named Entity Recognition," in International Journal of Crowd Science, vol. 6, no. 2, pp. 74-79, June 2022, doi: 10.26599/IJCS.2022.9100015.

[4] J. Diao, Z. Zhou and G. Shi, "Leveraging Integrated Learning for Open-Domain Chinese Named Entity Recognition," in International Journal of Crowd Science, vol. 6, no. 2, pp. 74-79, June 2022, doi: 10.26599/IJCS.2022.9100015.

[5] F. Zhao, X. Gui, Y. Huang, H. Jin and L. T. Yang, "Dynamic Entity-Based Named Entity Recognition Under Unconstrained Tagging Schemes," in IEEE Transactions on Big Data, vol. 8, no. 4, pp. 1059-1072, 1 Aug. 2022, doi: 10.1109/TBDATA.2020.2998770.

[6] Y. Luo, H. Zhao, Z. Zhang and B. Tang, "Open Named Entity Modeling From Embedding Distribution," in IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 11, pp. 5472-5483, 1 Nov. 2022, doi: 10.1109/TKDE.2021.3049654.

[7] T. Qian, M. Zhang, Y. Lou and D. Hua, "A Joint Model for Named Entity Recognition With Sentence-Level Entity Type Attentions," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 29, pp. 1438-1448, 2021, doi: 10.1109/TASLP.2021.3069295.

[8] E. C. Jibril and A. C. Tantuğ, "ANEC: An Amharic Named Entity Corpus and Transformer Based Recognizer," in IEEE Access, vol. 11, pp. 15799-15815, 2023, doi: 10.1109/ACCESS.2023.3243468.

[9] C. Liu, Y. Yu, X. Li and P. Wang, "Named Entity Recognition in Equipment Support Field Using Tri-Training Algorithm and Text Information Extraction Technology," in IEEE Access, vol. 9, pp. 126728-126734, 2021, doi: 10.1109/ACCESS.2021.3109911.

[10] S. A. A. Shah, M. Ali Masood and A. Yasin, "Dark Web: E-Commerce Information Extraction Based on Name Entity Recognition Using Bidirectional-LSTM," in IEEE Access, vol. 10, pp. 99633-99645, 2022, doi: 10.1109/ACCESS.2022.3206539.

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

[12] C. Li, G. Wang, J. Cao and Y. Cai, "A Multi-Agent Communication Based Model for Nested Named Entity Recognition," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 29, pp. 2123-2136, 2021, doi: 10.1109/TASLP.2021.3086978.

[13] Y. Chen et al., "A Boundary Assembling Method for Nested Biomedical Named Entity Recognition," in IEEE Access, vol. 8, pp. 214141-214152, 2020, doi: 10.1109/ACCESS.2020.3040182.

[14] Q. H. Ngo, T. Kechadi and N. -A. Le-Khac, "Domain Specific Entity Recognition With Semantic-Based Deep Learning Approach," in IEEE Access, vol. 9, pp. 152892-152902, 2021, doi: 10.1109/ACCESS.2021.3128178.

[15] D. Zhang et al., "Improving Distantly-Supervised Named Entity Recognition for Traditional Chinese Medicine Text via a Novel Back-Labeling Approach," in IEEE Access, vol. 8, pp. 145413-145421, 2020, doi: 10.1109/ACCESS.2020.3015056.

[16] L. -H. Lee and Y. Lu, "Multiple Embeddings Enhanced Multi-Graph Neural Networks for Chinese Healthcare Named Entity Recognition," in IEEE Journal of Biomedical and Health Informatics, vol. 25, no. 7, pp. 2801-2810, July 2021, doi: 10.1109/JBHI.2020.3048700.

[17] S. Dadas and J. Protasiewicz, "A Bidirectional Iterative Algorithm for Nested Named Entity Recognition," in IEEE Access, vol. 8, pp. 135091-135102, 2020, doi: 10.1109/ACCESS.2020.3011598.

[18] P. Zhu et al., "Improving Chinese Named Entity Recognition by Large-Scale Syntactic Dependency Graph," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 30, pp. 979-991, 2022, doi: 10.1109/TASLP.2022.3153261.

[19] H. Kim and J. Kang, "How Do Your Biomedical Named Entity Recognition Models Generalize to Novel Entities?," in IEEE Access, vol. 10, pp. 31513-31523, 2022, doi: 10.1109/ACCESS.2022.3157854.

[20] C. Zheng, Z. Wu, T. Wang, Y. Cai and Q. Li, "Object-Aware Multimodal Named Entity Recognition in Social Media Posts With Adversarial Learning," in IEEE Transactions on Multimedia, vol. 23, pp. 2520-2532, 2021, doi: 10.1109/TMM.2020.3013398.

[21] Y. Lou, T. Qian, F. Li and D. Ji, "A Graph Attention Model for Dictionary-Guided Named Entity Recognition," in IEEE Access, vol. 8, pp. 71584-71592, 2020, doi: 10.1109/ACCESS.2020.2987399.

[22] Y. Lou, T. Qian, F. Li and D. Ji, "A Graph Attention Model for Dictionary-Guided Named Entity Recognition," in IEEE Access, vol. 8, pp. 71584-71592, 2020, doi: 10.1109/ACCESS.2020.2987399.

[23] J. Kim, Y. Kim, S. Kang and J. Seo, "Adaptive Named Entity Recognition Using Distant Supervision for Contemporary Written Texts," in IEEE Access, vol. 9, pp. 80405-80414, 2021, doi: 10.1109/ACCESS.2021.3067315.

[24] J. Li, S. Shang and L. Chen, "Domain Generalization for Named Entity Boundary Detection via Metalearning," in IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 9, pp. 3819-3830, Sept. 2021, doi: 10.1109/TNNLS.2020.3015912.

[25] S. Su, J. Qu, Y. Cao, R. Li and G. Wang, "Adversarial Training Lattice LSTM for Named Entity Recognition of Rail Fault Texts," in IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 11, pp. 21201-21215, Nov. 2022, doi: 10.1109/TITS.2022.3182371.

[26] C. Yang, L. Sheng, Z. Wei and W. Wang, "Chinese Named Entity Recognition of Epidemiological Investigation of Information on COVID-19 Based on BERT," in IEEE Access, vol. 10, pp. 104156-104168, 2022, doi: 10.1109/ACCESS.2022.3210119.

[27] N. Liu, Q. Hu, H. Xu, X. Xu and M. Chen, "Med-BERT: A Pretraining Framework for Medical Records Named Entity Recognition," in IEEE Transactions on Industrial Informatics, vol. 18, no. 8, pp. 5600-5608, Aug. 2022, doi: 10.1109/TII.2021.3131180.

[28] H. Poostchi and M. Piccardi, "BiLSTM-SSVM: Training the BiLSTM with a Structured Hinge Loss for Named-Entity Recognition," in IEEE Transactions on Big Data, vol. 8, no. 1, pp. 203-212, 1 Feb. 2022, doi: 10.1109/TBDATA.2019.2938163.

[29] R. Yin, Z. Zhou and Z. Gao, "A Joint Model for Hierarchical Nested Information Extraction," in IEEE Access, vol. 10, pp. 50985-50995, 2022, doi: 10.1109/ACCESS.2022.3172970.

[30] J. Kim, Y. Ko and J. Seo, "Construction of Machine-Labeled Data for Improving Named Entity Recognition by Transfer Learning," in IEEE Access, vol. 8, pp. 59684-59693, 2020, doi: 10.1109/ACCESS.2020.2981361.