A Performance Comparison for Vietnamese Named Entity Recognition

Lai Le Dinh Duc
lailedinhduc@gmail.com
SE192137

Le Nguyen Quoc Anh
ntt.dia@gmail.com
SE192149

Nguyen Trung Kien
ntk.241205@gmail.com
SE192209

***Abstract*—Named Entity Recognition (NER) is a key task in Information Extraction systems that automatically identifies key entities in text, and supports advanced NLP applications. This study investigates the performance of PhoBERT, a pre-trained Vietnamese language model combined with CRF layers. We compare this architecture against classification models such as Softmax, Random Forest. Experiments on the VLSP 2016 dataset show that the PhoBERT + CRF model achieves the highest F1-Score at 93.0%, outperforming Random Forest (78.0%) and Softmax (89.0%). The results demonstrate the potential of Vietnamese NLP for future research and real-world use.**

***Index Terms*—NER, PhoBERT, NLP.**

I. Introduction

Named Entity Recognition (NER) is a crucial task in Natural Language Processing (NLP) aimed at the automatic identification and classification of entities, including names of individuals, organizations, locations, and other proper nouns within unstructured text. As a core component of Information Extraction systems, NER plays an important role in enabling various downstream NLP applications such as question answering, information retrieval, and machine translation.

NER is defined as a sequence labeling task. The input text is first segmented into smaller units known as tokens through a tokenization process. These tokens are then passed through an embedding layer, typically derived from a pre-trained language model, to generate contextualized vector representations for each token. Subsequently, these contextual embeddings are fed into a classification module, which assigns a label to each token, producing a sequence of predicted entity tags corresponding to the original token sequence.

In recent years, deep learning approaches—especially those leveraging pre-trained language models—have significantly advanced the state of NER systems across multiple languages. However, while multilingual models such as XLM-R [1], M-BERT [2] have achieved competitive performance, their effectiveness can be limited when applied to language-specific nuances. This is particularly evident in languages with rich morphological complexity, such as Vietnamese.

In order to address this limitation, our study explores the integration of PhoBERT [3], a pre-trained language model for Vietnamese with a 20GB corpus, as a contextualized embedding model, and systematically compares the performance of three different labeling approaches: Softmax, Conditional Random Fields (CRF), and Random Forest. While Softmax provides a simpler, token-level classification mechanism that treats each label prediction independently , CRF captures dependencies between adjacent labels, offering a more structured prediction at the cost of added complexity. In contrast, Random Forest classifies each token's embedding independently, without modeling label dependencies. This comparison allows us to assess whether the additional modeling capacity of CRF [4] yields significant improvements over the simpler approach.

According to the data set, we use the VLSP 2016 NER corpus [1](16,858 annotated sentences from online news) to train and evaluate the mentioned models. As the first official benchmarking effort for Vietnamese NER, VLSP 2016 provides a structured and standardized dataset, inspired by the CoNLL 2003 shared task [5]. It includes annotations for four named entity categories: person (PER), organization (ORG), location (LOC), and miscellaneous (MISC), allowing for consistent evaluation across systems.

II. Related Work

A. Traditional Models

Before the emergence of deep learning models, Support Vector Machines (SVMs) were widely employed as a traditional approach for Named Entity Recognition (NER) [6]–[8]. The method presented in [6] leverages handcrafted features and kernel functions to classify tokens. However, its performance heavily relies on feature engineering and lacks the capacity to model contextual dependencies effectively, limiting its scalability and adaptability to diverse domains..

Despite the emergence of deep learning models, several feature-based methods remained popular in Vietnamese NER, including a feature-rich Conditional Random Fields (CRF) model that integrates word features, word shapes, part-of-speech (PoS) tags, chunk tags, Brown clusters, and word embeddings [9]. This model achieved promising results on the VLSP 2016 dataset, demonstrating the effectiveness of CRF in leveraging hand-crafted linguistic features.

B. Deep Learning Models with BiLSTM Layer

Two influential deep learning architectures for NER are the BiLSTM-CRF [10] and BiLSTM-CNN-CRF [11] models. These architectures combine the bidirectional memory capability of BiLSTM with the CRF's ability to model the entire sequence effectively. They significantly outperform traditional

[1]https://vlsp.org.vn/vlsp2016/eval/ner

CRF models and serve as a foundation for later BERT-based systems.

In Vietnamese context, the VnCoreNLP toolkit [12] provides a complete pipeline for Vietnamese NLP, including word segmentation, part-of-speech tagging, and named entity recognition. VnCoreNLP also uses BiLSTM-CRF-based models for its NER module and has been widely used by the Vietnamese NLP research community, including in the preprocessing stages of state-of-the-art models like PhoBERT [3] and the COVID-19 NER [13] corpus.

C. Pretrained BERT-based Models

The introduction of BERT [14] revolutionized many NLP tasks, including NER. For Vietnamese, PhoBERT [3] represents the first large-scale monolingual pretrained language model. It adopts the BERT [14] architecture and was trained on a 20GB Vietnamese corpus using RoBERTa's optimized pretraining strategies. This version gives a strong foundation for future studies and practical applications in Vietnamese NLP, particularly in NER. Notably, PhoBERT employs the RDRSegmenter from VnCoreNLP for text preprocessing, showing how traditional tools and transformer models can work well together.

PhoNLP [15] is the first multi-task learning model for Vietnamese that simultaneously performs POS tagging, NER, and dependency parsing. It outperforms PhoBERT-base on NER while offering significant resource savings.

D. Large Language Models

Recent work has evaluated the capabilities of large language models (LLMs) such as GPT-4o, LLaMA, and Gemini in NER tasks [16]. These studies found that although LLMs show potential in chain-of-thought or few-shot prompt methods, they still fall short of the performance of fine-tuned models like BERT [14] and RoBERTa [17], especially in specialized domains such as finance.

III. Method

A. Input and Output

The overall pipeline of our method is illustrated in Figure 1. Our NER model takes a sentence as input in the form of a series of tokens, and outputs a matching series of labels, each of which specifies whether the token belongs to a named entity and, if so, what kind of entity it is. The IOB (Inside-Outside-Beginning) tagging scheme is frequently used for these labels. For example, given the input sentence "Nguyễn Văn A học ở Hà Nội", the expected output might be [B-PER, I-PER, I-PER, O, O, B-LOC], indicating that "Nguyễn Văn A" is a person entity and "Hà Nội" is a location entity.

B. Dataset

The dataset used in this project is derived from the Named Entity Recognition (NER) shared task at the VLSP 2016 workshop. The data was collected from *online news articles* published on the web. Prior to annotation, the text underwent

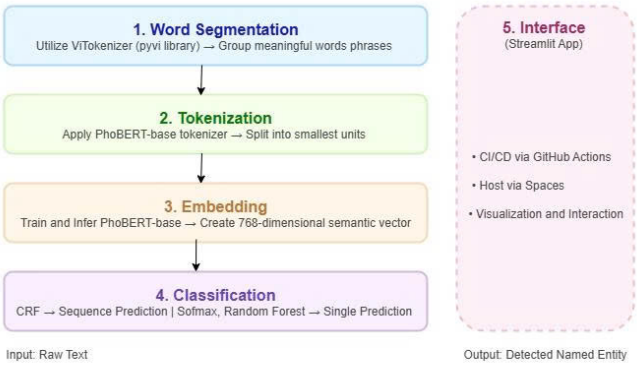

Figure 1. Vietnamese Named Entity Recognition Pipeline using PhoBERT

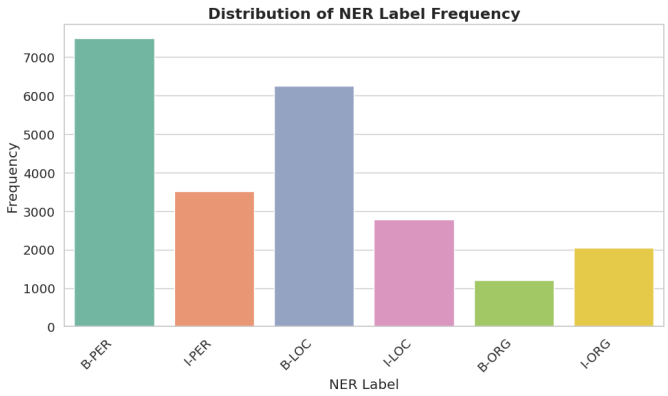

Figure 2. Distribution of NER Label Frequency

preprocessing, including word segmentation, to tokenize the sentences into discrete lexical units.

The dataset is structured into two columns: `tokens`, which represent the word-segmented sentences, and `labels`, which correspond to the named entity (NE) tags for each token. We used the standard IOB tagging system that CoNLL-2003 made popular. Specifically, the label `B-XXX` denotes the beginning of an entity of type `XXX`, `I-XXX` indicates subsequent tokens within the same entity, and `O` denotes tokens that are not part of any named entity. The distribution of NER label is shown in Figure 2.

As illustrated in Figure 3, The majority of sentences have lengths ranging from 8 to 36 tokens. Additionally, in Figure 4, more than 50% of the sentences in the dataset do not contain any named entities. Sentences with exactly one named entity account for 26.1%, while those with three named entities make up 11.7%.

C. Data Processing

As shown in Figure 1, for each new text data, the entire document first undergoes a word segmentation step to group individual words into meaningful phrases. Subsequently, the

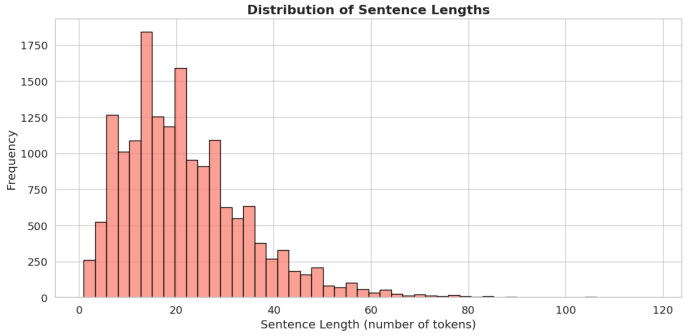

Figure 3. Distribution of sentence lengths

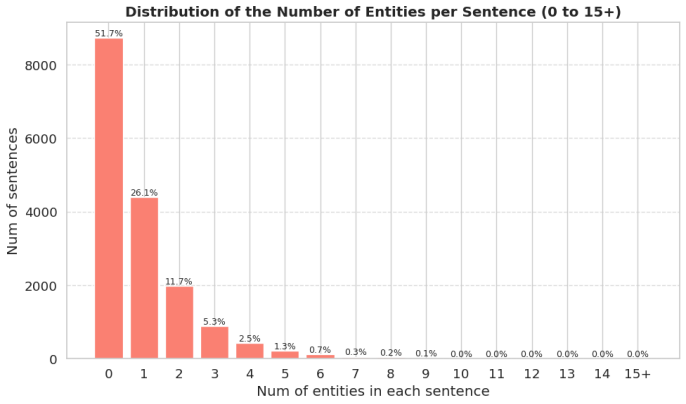

Figure 4. Distribution of the number of named entities per sentence.

segmented text is processed by a tokenizer to split it into the smallest meaningful units, or tokens, in preparation for generating embedding vectors. In cases where a word does not appear fully in the vocabulary and is split into multiple subwords, the embeddings of these subwords are aggregated using an average pooling method.

D. CRF Model

Conditional Random Fields (CRFs) are probabilistic models widely used for sequence labeling tasks. In this project, we employ a CRF to predict a sequence of labels from a sequence of embedding vectors generated by the PhoBERT [3] model. The core idea is to model the conditional probability of a label sequence $y = (y_1, y_2, \ldots, y_n)$ conditioned on an input sequence $x = (x_1, x_2, \ldots, x_n)$ Figure 5.

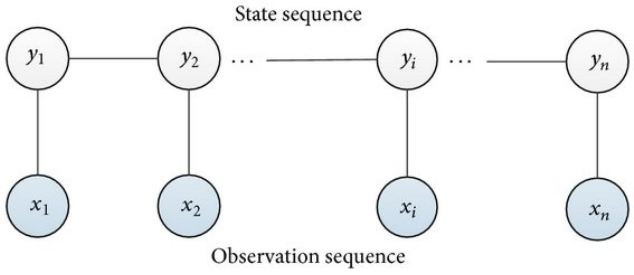

Figure 5. Linear-chain CRF model

There are two main types of features in CRF:

- Emission scores: These capture the probability of assigning the label y_i at position i based on the input x_i.
- Transition scores: These capture the probability of transitioning from label y_i to label y_{i+1}.

The conditional probability of a label sequence y given the input sequence x is defined as:

$$P(y \mid x) = \frac{\exp(\text{score}(x, y))}{\sum_{y' \in \mathcal{Y}} \exp(\text{score}(x, y'))}$$

where the score function is defined as:

$$\text{score}(x, y) = \sum_i (\text{emission}(x_i, y_i) + \text{transition}(y_i, y_{i+1}))$$

After understanding how to compute the score function $\text{score}(x, y)$, we can proceed to calculate the conditional log-likelihood of a sequence of tags given the emission scores using the following formula:

$$\text{Loss} = -\log P(y \mid x) = -\text{score}(x, y) + \log Z(x)$$

In this equation:

- $\text{score}(x, y)$: The total score of the ground-truth label sequence.
- $\log Z(x)$: The log-partition function, which computes the logarithm of the sum of scores of all possible label sequences.

In practice, the CRF implementation in the `torch-crf` library adopts this negative log-likelihood as the loss function.

During inference, the Viterbi algorithm is employed — a dynamic programming method used to find the most probable sequence of labels for a given observation sequence. The algorithm operates as follows:

- Probability Computation: It calculates the highest scoring path through a trellis (a grid structure representing all possible label sequences).
- Optimization: Intermediate results are stored and reused to avoid redundant computations on overlapping subproblems.
- Final Prediction: The optimal label sequence is retrieved by backtracking from the final position to the beginning, yielding the globally most probable path.

IV. Experiment

A. Data Preparation

During EDA phase, we observed that certain entity types, specifically `B-NAT` and `I-NAT` (used to label nationalities), appeared with very low frequency in the VLSP 2016 dataset. To reduce label imbalance and simplify the learning task,

we decided to replace these rare tags with the `O` label, which denotes tokens that do not belong to any named entity. This transformation excludes nationality-related entities from detection, allowing it to focus more effectively on entity types with higher representation in the dataset. While this decision may reduce the granularity of the model's output, it is justified in scenarios where recognizing nationalities is not a primary objective, and where improved performance on dominant classes is desired.

B. Training Configuration

a) Hardware and Software: All experiments were conducted on Google Colaboratory using a single NVIDIA Tesla T4 GPU with 16GB of VRAM. The model was implemented in PyTorch (v1.x) and leveraged the `torch-crf` library for CRF layers. We tracked training with Weights & Biases (`wandb`) for loss curves and metric logging.

b) Data Preparation: The dataset was randomly split into training and test sets using an 70/15/15 ratio. Embeddings were obtained from the pre-trained PhoBERT model [3], yielding n token-level vectors (each of dimension 768) per sentence. Labels were encoded as integer indices and padded to the maximum sequence length in each mini-batch, with a masking value of `-1` for padding positions.

c) Model Architecture: Our CRF tagger consists of a linear layer mapping the 768-dimensional PhoBERT embeddings to K tag scores, followed by a Conditional Random Field layer for sequence decoding and loss computation.

d) Hyperparameters:

- Optimizer: Adam [18] with learning rate $\alpha = 10^{-3}$.
- Batch size: 16 sentences per mini-batch.
- Epochs: 20 full passes over the training data.
- Loss: Negative log-likelihood computed by the CRF implementation (reduction=`mean`).
- Train/Val/Test split: 70% train,15% val and 15% test (random seed = 42).

e) Training Procedure: At each epoch, model parameters were updated to minimize the average CRF loss over the training set. Sequence lengths varied per batch, and a boolean mask was applied to ignore padded positions during both loss computation and decoding. After each epoch, we evaluated on both the training and test splits, computing precision, recall, F1-score (macro average), and accuracy. The best model (by test F1 or accuracy) was checkpointed.

C. Evaluation Metrics

Due to the inherently imbalanced nature of the Named Entity Recognition (NER) task—particularly between the O class, which denotes non-entity tokens, and the classes representing actual named entities—this project adopts two primary evaluation metrics: negative log-likelihood (NLL) and F1 score. NLL serves as the main objective function during training, while the F1 score is used to assess the model's effectiveness in identifying entities. In addition, precision and recall are also reported to provide complementary insights into the model's performance across different aspects of entity detection.

1) Primary Metrics:

a) Negative Log-Likelihood Loss: is the standard loss function used for training Conditional Random Fields (CRF). It penalizes the model when the correct label sequence does not receive a high score and normalizes the total score across all possible label sequences to compute a probability.

$$\begin{aligned}\mathcal{L}(x, y) &= -\log P(y \mid x) \\ &= -\text{score}(x, y) + \log \sum_{\tilde{y} \in \mathcal{Y}(x)} \exp(\text{score}(x, \tilde{y}))\end{aligned} \tag{1}$$

where:

- x - Input sequence.
- y - Ground truth label sequence.
- $\tilde{y}$ - A possible label sequence from the set of all valid sequences $\mathcal{Y}(x)$.
- $\text{score}(x, y)$ - The unnormalized score assigned by the model to the sequence (x, y).
- $\sum_{\tilde{y} \in \mathcal{Y}(x)} \exp(\text{score}(x, \tilde{y}))$ - The partition function, summing over scores of all possible label sequences.

b) F1 Score: is the harmonic mean of precision and recall, providing a balanced evaluation of the model's ability to correctly identify named entities. It is especially suitable for imbalanced datasets, such as in Named Entity Recognition (NER), where the majority of tokens belong to the non-entity class `O`.

$$F1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2}$$

In this project, we adopt Macro-F1 as the overall evaluation metric. This is computed by averaging the F1 scores of individual entity labels, giving equal weight to each label regardless of their frequency in the dataset.

$$\text{Macro-F1} = \frac{1}{L} \sum_{i=1}^{L} F1_i \tag{3}$$

where:

- $F1_i$ – the F1 score of the i-th named entity label.
- L – total number of entity labels (excluding the `O` class).

2) Secondary Metrics:

a) Precision and Recall: are standard evaluation metrics used in sequence labeling tasks such as Named Entity Recognition (NER). Precision measures the proportion of predicted entities that are correct, while recall measures the proportion of true entities that are successfully identified by the model.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{4}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{5}$$

where:

- True Positives (TP) – the number of correctly predicted entities.

- False Positives (FP) – the number of predicted entities that do not match any ground truth.
- False Negatives (FN) – the number of ground truth entities that the model failed to predict.

b) Entity-level Accuracy: measures the proportion of named entities that are predicted exactly correctly by the model—both in terms of the entity boundaries and the entity type. Unlike token-level accuracy, which can be misleading in highly imbalanced datasets, entity-level accuracy focuses on the correctness of entire entities, making it a more informative metric for Named Entity Recognition tasks.

$$\text{Entity-level Accuracy} = \frac{\text{Number of correctly predicted entities}}{\text{Total number of ground truth entities}} \quad (6)$$

V. Result

A. CRF Performance

Table I
CLASSIFICATION REPORT OF THE PHOBERT + CRF MODEL ON THE VIETNAMESE NER TEST SET

Label	Precision	Recall	F1-score	Support
O	1.00	1.00	1.00	51 036
B-PER	0.99	0.98	0.98	1 112
I-PER	0.97	0.99	0.98	506
B-ORG	0.83	0.84	0.84	180
I-ORG	0.88	0.84	0.86	291
B-LOC	0.93	0.95	0.94	939
I-LOC	0.93	0.91	0.92	428
Macro avg	0.93	0.93	0.93	54 492
Weighted avg	0.99	0.99	0.99	54 492

During the training phase in Fig 6, both training and validation loss decreased and stabilized, indicating good convergence. However, the training and validation F1 scores increased and then plateaued, with a slight gap suggesting mild overfitting.

On the test set, the combination of PhoBERT and CRF achieved an overall F1-score of 0.9044, with F1-scores of 0.9828 for `B-PER` and 0.9860 for `I-PER`, indicating highly accurate identification of person entities. For location entities, it attains F1-scores of 0.9070 on `B-LOC` and 0.8620 on `I-LOC`, with some interior tokens of multi-word locations occasionally missed. In contrast, performance on organizations is lower, with F1-scores of 0.7901 for `B-ORG` and 0.8054 for `I-ORG`, indicating confusion or missed detection of some organization names.

B. Compared to Softmax and Random Forest

1) Performance of PhoBERT with Softmax Classifier: Table II presents the classification metrics of the PhoBERT model combined with a Softmax classifier evaluated on the VLSP2016 test set. The model demonstrates robust performance, especially on frequently occurring entity types.

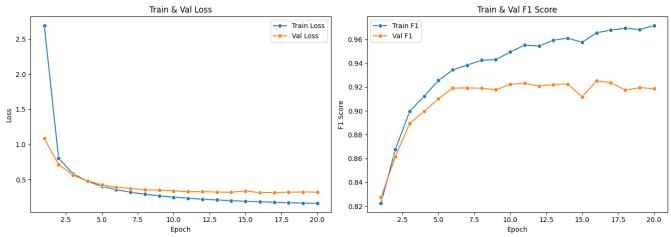

Figure 6. Loss and F1-Score of PhoBert and CRF

Table II
PERFORMANCE METRICS OF PHOBERT + SOFTMAX ON VLSP2016

Label	Precision	Recall	F1-score	Support
O	1.00	1.00	1.00	68 476
B-PER	0.99	0.98	0.98	1 464
I-PER	0.98	0.98	0.98	686
B-ORG	0.77	0.82	0.80	257
I-ORG	0.80	0.77	0.78	430
B-LOC	0.88	0.90	0.89	1 241
I-LOC	0.83	0.82	0.82	554
Macro avg	0.89	0.89	0.89	73 108
Weighted avg	0.99	0.99	0.99	73 108

The model achieves particularly high performance on non-entity tokens (`O`) with an F1-score of 0.9975, which reflects strong ability in identifying irrelevant or background text. For named entities, the model is most accurate on person and location entities. Specifically, `B-PER` and `I-PER` achieve F1-scores of 0.9864 and 0.9575, respectively. These results highlight the model's effectiveness in capturing consistent and frequent entity patterns.

However, performance drops significantly for organizational and location categories. For instance, `I-ORG` and `B-ORG` receive F1-scores of 0.8 and 0.7904, indicating difficulty in consistently detecting organization names, which often appear in more complex or varied contexts.

In summary, the PhoBERT-Softmax architecture performs strongly on dominant classes, particularly person and location entities.

2) Performance of PhoBERT with Random Forest Classifier: Table III presents the classification metrics of the PhoBERT model combined with a Random Forest classifier evaluated on the VLSP2016 test set.

The model is excellent at recognizing background tokens (`O`), achieving an F1-score of 0.9728, which shows it rarely miss label non-entity words. For person entities, performance is acceptable: `B-PER` scores an F1 of 0.6911 and `I-PER` reaches 0.6977. Organizations remain the hardest class, with `B-ORG` at 0.3901 and `I-ORG` at just 0.2994. Location aren't doing so well, with F1-scores of 0.5177 for `B-LOC` and 0.3907 for `I-LOC`. Together, the macro-averaged F1 across all entity labels is 0.5657, highlighting strong background detection but struggle with less frequent entity types.

Table III
PERFORMANCE METRICS OF PHOBERT + RANDOM FOREST ON VLSP2016

Label	Precision	Recall	F1-score	Support
O	0.99	0.99	0.99	69 221
B-PER	0.79	0.94	0.86	1 496
I-PER	0.87	0.95	0.91	704
B-ORG	0.78	0.58	0.66	242
I-ORG	0.73	0.46	0.57	411
B-LOC	0.85	0.77	0.81	1 249
I-LOC	0.73	0.65	0.69	557
Macro avg.	0.82	0.76	0.78	73 880
Weighted avg.	0.98	0.98	0.98	73 880

C. Error Analysis

1) Uncertainty in Detecting Entities Due to Imbalanced Data: During training, the model struggled to detect entities from less frequent classes because of the imbalance in the dataset. This led to a bias toward more common entity types. We considered using oversampling to address this, but it was not suitable for NER tasks, where preserving the natural structure of language is important. We also tried adding more data by crawling news articles, but this didn't lead to noticeable improvements. As a result, we decided to continue using the original VLSP 2016 dataset.

VI. CONCLUSION

In this project, we have integrated the monolingual PhoBERT model with both Softmax and CRF decoders. Through experiments on the VLSP 2016 dataset, our PhoBERT+CRF architecture achieved an overall F1-score of 0.93, outperforming simpler classification heads, including Softmax (0.89) and Random Forest (0.78).

Our error analysis revealed that class imbalance and scarce entity types remain challenging for sequence labeling models. To address these issues, future work will explore data augmentation techniques, domain-specific corpora, and multi-task learning frameworks that jointly model related tasks such as part-of-speech tagging or dependency parsing.

Overall, this project underscores the value of combining language-specific pretraining with Conditional Random Fields for high-performance NER in Vietnamese, and provides a solid foundation for our future research and practical deployments in information extraction systems.

REFERENCES

[1] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," *arXiv preprint arXiv:1911.02116*, 2019.

[2] T. Pires, E. Schlinger, and D. Garrette, "How multilingual is multilingual BERT?" *arXiv preprint arXiv:1906.01502*, 2019.

[3] D. Q. Nguyen and A. T. Nguyen, "PhoBERT: Pre-trained language models for Vietnamese," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020.

[4] J. Lafferty, A. McCallum, and F. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," in *Proceedings of the 18th International Conference on Machine Learning (ICML)*, 2001, pp. 282–289.

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

[6] H. Isozaki and H. Kazawa, "Efficient support vector classifiers for named entity recognition," in *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002.

[7] A. Ekbal and S. Bandyopadhyay, "Named entity recognition using Support Vector Machine: A language independent approach," *International Journal of Electrical, Computer, and Systems Engineering*, vol. 4, no. 2, pp. 155–170, 2010.

[8] B. Sivaji and E. Asif, "Named Entity Recognition in Bengali and Hindi using Support Vector Machine," *Lingvisticæ Investigationes*, vol. 34, no. 1, pp. 35–67, 2011.

[9] P. Q. Nhat Minh, "A Feature-Rich Vietnamese Named Entity Recognition Model," *Computación y Sistemas*, vol. 26, no. 3, pp. 1323–1331, 2022.

[10] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF models for Sequence Tagging," in *Proceedings of the 29th Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2015.

[11] J. P. Chiu and E. Nichols, "Named entity recognition with Bidirectional LSTM-CNNs," in *Transactions of the Association for Computational Linguistics*, vol. 4, 2016, pp. 357–370.

[12] X.-S. Vu, M. L. Nguyen, and X.-K. Nguyen, "VnCoreNLP: A vietnamese natural language processing toolkit," in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018.

[13] N. M. Do and H. T. Nguyen, "COVID-19 Named Entity Recognition for Vietnamese: Datasets and Baselines," *arXiv preprint arXiv:2105.02724*, 2021.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, 2019.

[15] A. T. Nguyen and N. Dat Quoc, "PhoNLP: A joint multi-task learning model for vietnamese part-of-speech tagging, named entity recognition and dependency parsing," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021.

[16] W. Zhang, C. Li, and M. Tan, "Evaluating Large Language Models on Named Entity Recognition: Capabilities and Limitations," *To Appear, ACL 2025*, 2025.

[17] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," in *arXiv preprint arXiv:1907.11692*, 2019.

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.