VNU Journal of Science:
Computer Science and Communication Engineering
Journal homepage: http://www.jcsce.vnu.edu.vn/index.php/jcsce

Original Article

NER - VLSP 2021: Two Stage Model for Nested Named Entity Recognition

Quan Chu Quoc, Vi Ngo Van*

VCCorp Corporation, 1 Nguyen Huy Tuong Thanh Xuan, Hanoi, Vietnam

Received 27 December 2021
Revised 5 April 2022; Accepted 5 May 2022

Abstract: Named entity recognition (NER) is a widely studied task in natural language processing. Recently, a growing number of studies have focused on the nested NER. The span-based methods consider the named entity recognition as span classification task, can deal with nested entities naturally. But they suffer from class imbalance problem because the number of non-entity spans accounts for the majority of total spans. To address this issue, we propose a two stage model for nested NER. We utilize an entity proposal module to filter an easy non-entity spans for efficient training. In addition, we combine all variants of the model to improve overall accuracy of our system. Our method achieves 1st place on the Vietnamese NER shared task at the 8th International Workshop on Vietnamese Language and Speech Processing (VLSP) with F1-score of 62.71 on the private test dataset. For research purposes, our source code is available at https://github.com/quancq/VLSP2021_NER

Keywords: Nested Named Entity Recognition, Vietnamese.

1. Introduction

Named entity recognition (NER) is a fundamental task in natural language processing. Nested entities are named entities containing other named entities as [Tổng công ty điện lực Hà Nội] entity contains [Hà Nội] entity. Nested NER task studied on several benchmark datasets of high-resource languages as ACE 2004, ACE 2005, GENIA. However, it has not been widely analyzed in low-resource languages like Vietnamese. Data resources for the Vietnamese nested NER task are limited, including the public dataset from VLSP 2018 NER shared task [1]. This dataset contains four generic entity categories as PERSON, ORGANIZATION, LOCATION and MISCELLANEOUS. At VLSP 2021 NER shared task, organizer released a diverse and challenging nested NER dataset [2]. Some characteristics of this dataset are depicted in section 4.1.

In this paper, we introduce a method to deal with nested NER task by adopting ideas from biaffine NER model [3]. Biaffine model is

* Corresponding author.
E-mail address: vingovan@admicro.vn

https://doi.org/10.25073/2588-1086/vnucsce.368

widely used and obtained effective results in several works as dependency parsing model [4], named entity recognition model [3], relation extraction model [5], multi-task model [6]. Yu [3] formulates NER as the task of identifying start and end indices, as well as assigning a category to the span defined by these pairs. Their system uses a biaffine classifier to assign scores to all possible spans in a sentence. In our implementation, we found that classifying all possible spans caused a class imbalance problem. Because the number of non-entity spans accounts for the majority of total spans, loss function is dominated by too many easy non-entity spans. This lead to inefficient training. In object detection task of the computer vision area, class imbalance problem between foreground and background class is quite often. Inspired by Faster R-CNN model [7], we propose a two stage model for nested NER. We utilize an entity proposal module to filter an easy non-entity spans for efficient training. In addition, we combine all variants of the model to improve overall accuracy of our system. The experiments show that our system achieves promising results on VLSP 2021 NER shared task. Our main contributions are as follows:

- We propose a two stage model for nested NER. We utilize an entity proposal module to handle class imbalance problem.
- We present an ensemble way to improve accuracy of system.
- We design several pre-processing steps to decrease noise of the dataset.
- Experimental results show that our system achieves 1st place on Vietnamese NER shared task at VLSP 2021.

In the rest of the paper, the related work is briefly summarized in section 2, section 3 and 4 present our methods and experiments respectively. Finally, the conclusion is drawn in section 5.

2. Related Work

Traditional works [8-10] model named entity recognition as sequence labeling task, thus enabling to leverage methods to recognize flat entities. However, since tokens in nested entities may belong to multiple labels, the traditional sequence labeling approaches can not meet the demand.

Various methods have been proposed to deal with the nested NER task, including the sequence-to-sequence methods [11, 12] and the span-based methods [3]. The sequence-to-sequence method treats nested NER as a sequence generation task in which labels decoded one by one in order. However, in the NER task, the output labels are essentially an unordered set. In this manner, even if the model predicts all correct labels, it may cause an unreasonable training loss as a result of inconsistent order. Moreover, due to long inference time, sequence-to-sequence method is not suitable for real-world applications. The span-based method classifies the candidate spans which are extracted from a text sequence. Yu [3] uses biaffine model [4] as the entity classifier. Due to is widely used in many works for high-resource languages, we adopt the biaffine model in our system to deal with Vietnamese nested NER task. Different from Yu [3], we utilize an entity proposal module to handle class imbalance problem better.

3. Methods

3.1. Overview

Problem formalization: Given a document D, the goal of the system is providing an entity set. Each entity is identified by start and end indices of tokens in D, as well as a category in the predefined set.

We formulate nested NER as the span classification task. With n is the number of input tokens, we have total possible spans is n(n2−1) . Because number of the non-entity spans much more than number of the entity spans, loss function is dominated by many easy non-entity spans. This class imbalance problem lead to inefficient training. To address this issue, we propose a two stage model which inspired by models in the computer vision area. In first stage, we utilize an entity proposal module to generate candidate spans. In second stage, we employ a biaffine model to classify candidate spans. Compared to Yu [3], instead of classifying all possible candidate spans, our method only classifies selectively candidate spans. Overview, our model contains three modules: text representation module (TRM), entity proposal module (EPM) and entity classification module (ECM). Figure 1 shows an overview of the proposed architecture. Firstly, TRM module encodes information of input tokens and generate contextual representations. These representations are fed into EPM module to generate candidate spans. Finally, ECM module classifies all candidate spans generated by EPM module. Details of TRM, EPM and ECM module are presented in section 3.2, 3.3 and 3.4 respectively. Section 3.5 presents some details how the system work.

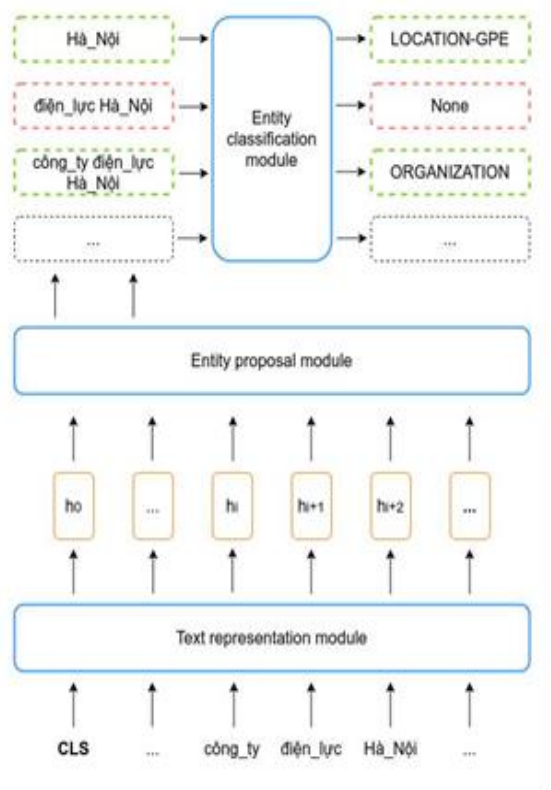


Figure 1. Our proposed model.

3.2. Text Representation Module

The purpose of TRM module is generating representations which encode information of the input tokens. Due to remarkable representation capacity, many Pretrained Language Models (PLMs) [10, 13] are used to provide strong performance for various tasks. PhoBERT [14] is one of the most powerful PLMs for Vietnamese and has been widely used in many works. Therefore, we employ PhoBERT as backbone of TRM module. Input of TRM module is a list of the tokens $X = \{x0, x1, ..., xn-1\}$ with n is the number tokens. Output of TRM module is contextual representation vectors $H = \{h_0, h_1, ..., h_{n-1}\}$, where $H \in R^{n \times d1}$ and d_1 is the dimension of the PhoBERT features.

3.3. Entity Proposal Module

EPM module works with two steps: entity-related ability classification step and candidate generation step.

In first step, with each input token, we predict possibility it related to the entity. This is binary classification task. Table 1 illustrates how we generate ground truth labels for the binary classifier. We propose three variants of EPM module to concrete for mentioned entity-related ability. The purpose of the variants is as follows:

• EPM-1: predict whether token belong to the entity.

• EPM-2: predict whether token is start or end token of the entity.

• EPM-3: predict whether token belong to start or end word of the entity.

To clearly explain for EPM-3, because PhoBERT uses fastBPE [15], each word may be segmented into several tokens. We employ one (for EPM-1) or two (for EPM-2 and EPM-3) fully connected layers to predict entity-related probability. This probability of the tokens is calculated as follow:

$$\begin{aligned} P &= \sigma(WH + b) \\ P^s &= \sigma(W^s H + b^s) \qquad (1) \\ P^e &= \sigma(W^e H + b^e) \end{aligned}$$

where W, W^s, W^e, b, b^s and b^e are learnable parameters.

We use binary cross entropy (BCE) as the training objective of the EPM module.

Loss function of EPM module is calculated as follows:

$L_{EPM-1} = BCE(Y, P)$

$L_{EPM-2} = L_{EPM-3} = BCE(Y^s, P^s) + BCE(Y^e, P^e)$ (2)

where P, P^s and P^e are predictions of our model. Y, Y^s and Y^e are ground truth labels as illustrated in table 1.

In second step, we generate all possible valid candidate spans from the entity-related tokens.

3.4. Entity Classification Module

From the representation vectors H obtained by TRM module, we use two fully connected layers to calculate new representations which are obtained by:

$H^s = HW_1 + b_1$

$H^E = HW_2 + b_2,$ (3)

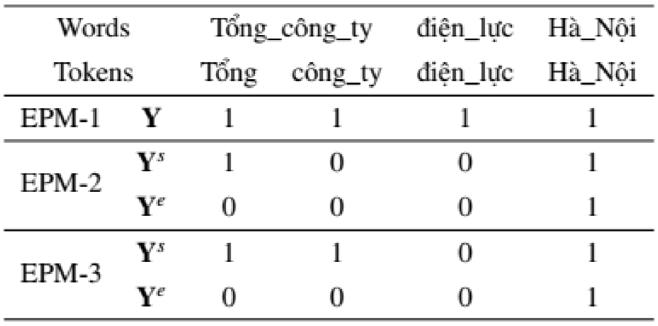
where W_1, W_2, b_1 and b_2 are learnable parameters.

H^S and $H^E \in R^{n\times d2}$ and d_2 is the dimension of the hidden layer. We found that $H^S{}_i$ and $H^E{}_j$ representations are necessary features to classify candidate span $s_{i.j}$. Therefore, we concat representations in H^S and H^E to obtain the tensor $T \in R^{n\times n\times d2}$. Finally, we use the fully connected layer to predict categories of the candidate spans.

$M = SoftMax(TW_3 + b_3)$ (4)

where W_3 and b_3 are learnable parameters. Loss LECM is calculated by cross entropy function. In LECM, we only consider $s_{i.j}$ candidates in which ith and jth tokens are entity-related tokens.

Table 1. Ground truth labels of EPM module.
Tổng công ty điện lực Hà Nội and Hà Nội are gold entities:



Words		Tổng_công_ty		điện_lực	Hà_Nội
Tokens		Tổng	công_ty	điện_lực	Hà_Nội
EPM-1	**Y**	1	1	1	1
EPM-2	**Y**^{s}	1	0	0	1
	Y^e	0	0	0	1
EPM-3	**Y**^{s}	1	1	0	1
	Y^e	0	0	0	1

3.5. Training and Inference Stage

Training stage: We use jointly LEPM and LEC M to train our system. Final loss is calculated as follows:

$L = \alpha L_{EPM} + \beta L_{ECM}$ (5)

where α and β are fine-tuning hyper-parameters.

In addition, to prevent error propagation by false negative of the EPM module, we also feed ground truth spans that did not generate by EPM into final loss.

Inference stage: Only candidate spans generated by EPM are fed into ECM and obtain final prediction. In EPM-1 and EPM-3, final word-level span predictions are calculated by majority combination from all token-level span predictions.

4. Experiments

4.1. Dataset

The VLSP 2021 organizer released a diverse and challenging dataset for Vietnamese nested NER. The dataset contains 1500 raw documents, 14 entity categories and 26 entity sub-categories. The number of entity categories of VLSP 2021 dataset is more than previously published VLSP NER dataset [1]. The challenge of the dataset is revealed in table 2.

4.2. Experiment Setup

After analyzing the dataset carefully, we design some pre-processing steps to clean the dataset as follows:

- We use VnCoreNLP library for sentence and word segmentation.
- We truncate tokens which exceed max input length of PhoBERT.
- We remove some punctuations end of the entity tags.
- We analyze the inconsistently annotated cases and design rules to annotate them consistently. We revised nearly 3000 cases.
- The dataset has too many missed annotated cases. This issue is quite often for several sub-categories of DATETIME and QUANTITY. Therefore, we use regular expression to annotate them automatically. We revised about 5506 cases.

Table 2. Examples illustrate the challenge of the dataset:

Challenge	**Example**
Understand input context	Ông ấy sử dụng FacebookPRODUCT để giải trí.
	Ông ấy ứng tuyển vào FacebookORGANIZAT ION .
Memory specific entity	Ông ấy đang công tác tại Bộ Y tếORGANIZAT ION−MED.
	Ông ấy đang công tác tại Bộ Giáo dục và đào tạoORGANIZAT ION.
Diverse pattern	PERSONTYPE: chủ tịch, giám đốc, giáo viên, bác sĩ, etc.
Long-span entity	EVENT: hội thi kỷ niệm ngày Nhà giáo Việt Nam.
	ADDRESS: 56/4 Hồng Lĩnh, tổ dân phố 9, phường Tứ Hạ - thị xã Hương Trà - TT Huế.
	SKILL: giàu kinh nghiệm làm việc trong lĩnh vực truyền thông, khởi nghiệp.

To prevent overfitting, we split original dataset (1500 documents) into a new training set (1200 documents) and a development set (300 documents). This splitting steps made by two different random seeds. We use the development set to choose the best model. Table 3 shows the hyper-parameters of the our best model.

4.3. Results

F1-micro is used to evaluate nested NER model in the competition. Details are as follows:

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

With

$$\text{Precision} = \frac{NEtrue}{NEsys}$$

$$\text{Recall} = \frac{NEtrue}{NEref}$$

where NEtrue is the number of the entities which is correctly recognized by the system. NEsys and NEre f are the number of the entities in recognizing system and gold data respectively.

Table 3. Hyper-parameter setting:

Hyper-parameter	Value
d1	768
d2	384
α	0.5
β	0.5
Max input length	256
Learning rate	2e-5
Batch size	4
Optimizer	AdamW
Number epochs	30

Table 4 shows our experimental results on the development set. As we can see, no variant better each other consistently. In our experiments, we use F1-macro as main metric to evaluate models. Because model 01 obtains highest F1-macro on two development sets, we submit this model as our best single model. To improve accuracy of our system, we make combination by majority voting from single models. Our final solutions are as follows:

- VCTus-01: model 01.
- VCTus-02: ensemble model from three models: model 01, 02 and 03.
- VCTus-03: ensemble model from six models: model 01, 02, 03, 04, 05 and 06.

Table 4. Results on the development set:

Model	Variant	F1-macro	F1 - micro
Random seed 1			
01	EPM-1	65.41	75.44
02	EPM-2	65.24	77.13
03	EPM-3	64.91	76.68
Random seed 2			
04	EPM-1	61.81	77.74
05	EPM-2	60.32	77.28
06	EPM-3	60.57	77.48

Results on the private test set are shown in table 5. We also consider the results of top 3 participants on the private test set. The results are provided by the organizer. Our system achieved highest F1-micro in the competition.

Table 5. Results on the private test set. The results provided by the organizer are marked with †:

Team	Model	F1-micro
NER1	01	55.38†
	01	58.50†
NER2	02	58.41†
	03	59.71†
	01	61.04
NER3 (Our)	02	62.26
	03	62.71

5. Conclusion

In this paper, we have presented several methods to adopt biaffine model and deal with Vietnamese nested NER task. We proposed a two stage model to handle class imbalance problem. We also analyzed dataset carefully and designed several pre-processing steps to clean the dataset. Experimental results showed that our methods are effective and achieve promising performance on the dataset of the competition.

In the future, we plan to analyze error of our model and adopt more recent approaches to our method. In addition, we also want to investigate the data augmentation and semi-supervised methods to improve accuracy for low-resource NER task.

References

[1] H. Nguyen, Q. Ngo, L. Vu, V. Tran, H. Nguyen, VLSP Shared Task: Named Entity Recognition, Journal of Computer Science and Cybernetics, Vol. 34, 2019, pp. 283–294. doi:10.15625/1813-9663/34/4/13161.

[2] H. M. Linh, D. D. Dao, N. T. M. Huyen, T. Quyen, D. X. Dung, NER Challenge: Named Entity Recognition for Vietnamese, VLSP 2021.

[3] J. Yu, B. Bohnet, M. Poesio, Named Entity Recognition as Dependency Parsing, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020.

[4] T. Dozat, C. D. Manning, Deep Biaffine Attention for Neural Dependency Parsing, in: International Conference on Learning Representations (ICLR).

[5] D. Q. Nguyen, K. Verspoor, End-to-End Neural Relation Extraction Using Deep Biaffine Attention, in: L. Azzopardi, B. Stein, Fuhr, P. Mayr, C. Hauff, D. Hiemstra (Eds.), Advances in Information Retrieval - 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, April 14-18, 2019, Proceedings, Part I, Vol. 11437 of Lecture Notes in Computer Science, Springer, 2019, pp. 729–738.

[6] L. T. Nguyen, D. Q. Nguyen, 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 of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations, Association for Computational Linguistics, Online, 2021.

[7] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, in: C. Cortes, Lawrence, D. Lee, M. Sugiyama, R. Garnett

(Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc.

[8] G. Lample, M. Ballesteros, S. Subramanian, Kawakami, C. Dyer, Neural Architectures for Named Entity Recognition, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, San Diego, California, 2016.

[9] M. E. Peters, M. Neumann, M. Iyyer, Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep Contextualized Word Representations, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018.

[10] J. Devlin, M.-W. Chang, K. Lee, 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: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019.

[11] neural architectures for nested NER through linearization.

[12] J. Wang, L. Shou, K. Chen, G. Chen, Pyramid: A Layered Model for Nested Named Entity Recognition, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020.

[13] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A Robustly Optimized BERT Pretraining Approach, ArXiv abs/1907.11692.

[14] D. Q. Nguyen, A. Tuan Nguyen, PhoBERT: Pre-trained language models for Vietnamese, in: Findings of the Association for Computational Linguistics: EMNLP 2020, Association for Computational Linguistics, Online, 2020.

[15] R. Sennrich, B. Haddow, A. Birch, Neural Machine Translation of Rare Words with Subword Units, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Berlin, Germany, 2016.