VNU Journal of Science: Comp. Science & Com. Eng, Vol. 38, No. 1 (2022) 98-104

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: A Span-Based Model for Named Entity Recognition Task with Co-teaching+ Training Strategy

Pham Hoai Phu Thinh*, Vu Tran Duy, Do Tran Anh Duc

University of Science, Vietnam National University, Ho Chi Minh City, Vietnam

Received 28 December 2021
Revised 27 April 2022; Accepted 5 May 2021

Abstract: Named entities containing other named entities inside are referred to as nested entities, which commonly exist in news articles and other documents. However, most studies in the field of Vietnamese named entity recognition entirely ignore nested entities. In this report, we describe our system at VLSP 2021 evaluation campaign, adopting the technique from dependency parsing to tackle the problem of nested entities. We also apply Coteaching+ technique to enhance the overall performance and propose an ensemble algorithm to combine predictions. Experimental results show that the ensemble method achieves the best F1 score on the test set at VLSP 2021.

Keywords: Named entity recognition, Vietnamese.

1. Introduction

Named entity recognition (NER) is the process of automatically identifying entities in text with their pre-defined categories, commonly used in information extraction. The term "Named Entity" first appeared at the sixth Message Understanding Conference (MUC-6) [1], and accordingly, there have been scientific events giving much effort to this field, such as CoNLL 2003 [2].

In Vietnamese, the first evaluation campaign to promote the development of high quality NER systems is VLSP 2016 NER evaluation [3], considering four entity types: person (PER), organization (ORG), location (LOC) and miscellaneous (MISC). The dataset in VLSP 2016 includes morpho-syntactic and NE annotations, namely gold word segmentation, POS and chunking tags, using CoNLL format [2]. In contrast, the corpus at VLSP 2018 is in XML format, with only raw texts and named-entity tags, which is more complicated since no linguistic information is provided [4]. VLSP 2021 NER evaluation [5] is one of the next developments of VLSP 2018, considering more types of categories (14 main categories, 26 subcategories and 1 generic). The third competition on evaluating NER systems is more challenging since the data also contain only raw texts enriched with much more NE tags, enabling to fully capture meaningful information. Besides, the models need to

* Corresponding author.
E-mail address: phpthinh18@apcs.fitus.edu.vn

https://doi.org/10.25073/2588-1086/vnucsce.328

distinguish main categories and subcategories effectively.

In this report, we describe our new approach to the NER problem. In general, in both the two previous campaigns, most of the systems consider NER to be a sequence labeling problem, leading to them ignoring or tackling not fully nested entities. By adopting ideas from biaffine dependency parsing model, we develop a span-based system to be able to effectively identify nested entities, following the study presented in [6]. We consider NER to be the task of recognizing the start and end indices, similar to heads and dependents in dependency parsing, and assigning entity type to the span. From the score matrix from the model, we rank the candidates based on their scores and select top-ranked spans satisfying the constraints for nested entities.

Intending to improve the performance, we attempt to apply regular expressions to catch some entities with specific patterns, such as QUANTITY-PER, DATETIME-DATE, EMAIL and IP. In addition, to effectively distinguish different categories, especially main categories and subcategories, we adopt Co-teaching+ technique [7], assuming the dataset to be slightly noisy. This allows the system to learn from the disagreement between the predictions from two networks, then optimize the parameters from high-confidence data, ignoring a small chunk of noisy data. Furthermore, we propose an algorithm to ensemble two models for better results.

In summary, our main contributions are:

- We introduce a new approach to NER task for Vietnamese, with the ability to tackle the problem of nested entities.
- We apply a training technique to deal with noisy labels from the dataset to our systems.
- We propose an algorithm to combine different predictions that obtains better results.

The rest of the paper is as follows. Section 2 covers other studies in the field of Vietnamese Named Entity Recognition. In section 3, we describe the NER system that we develop to participate in the workshop. The evaluation results of our system are presented in section 4. Finally, we conclude our work at the workshop in section 5.

2. Related Work

At VLSP 2018, the author of [8] proposed a sequence labeling model, Conditional Random Fields (CRF) to tackle the nested NER problem, which just considers level-1 and level-2 entities. This model combines word, word-shape, Brown cluster-based and word embedding-based features. By combining entity tags at all levels to generate joint-tags, they show that the model improved the accuracy of nested named-entity recognition, achieving 73.48% F1 score for all levels.

ZA-NER [9], which is the best system participating in VLSP 2018, is the combination of BiLSTM and CRF, with the help of word embeddings from the character level. This system achieves the highest results on the standard test set, 74% F1 score for level 1 and 68% F1 score for nested evaluation.

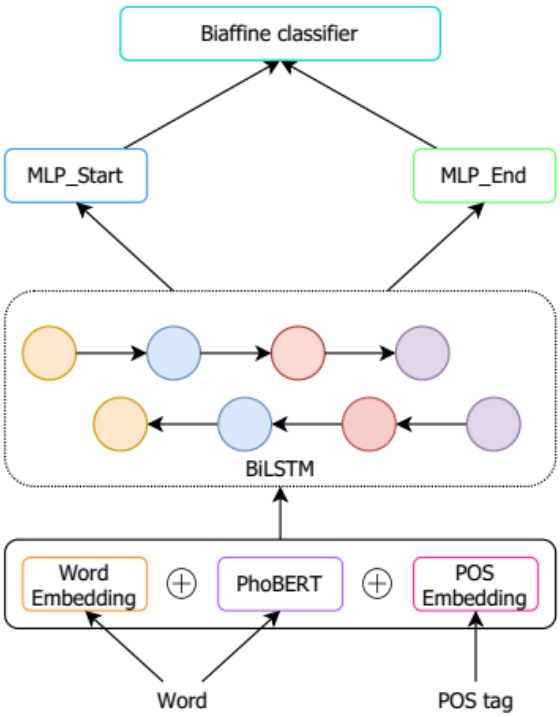

Figure 1. Biaffine architecture.

VNER [10] takes advantage of character-level language models and word embeddings to encode words. The model contains attention layers to compute probability distribution for each word, enabling such layers to focus on parts of the sentence. The best F1 score from this model shows potential, obtaining 77.52% on VLSP 2018 corpus.

3. Methods

3.1. Biaffine model

Our model is largely based on the dependency parser of [11] and [6]. Figure 1 illustrates the architecture of our model.

We use both word embeddings and POS embeddings, with the addition of PhoBERT [12] to get the contextual representations. The concatenation of 3 features is forwarded into BiLSTM layers to obtain the word representations (equations 1, 2). Given an n-length sequence:

$$x_i = E(w_i) \oplus \text{PhoBERT}(w_i) \oplus E(pos_i) \quad (1)$$

$$B = \text{BiLSTM}(X) \quad (2)$$

where E refers to the embedding layers, and wi, posi are the ith word, POS tag respectively of the input sequence.

After that, two separate MLP layers are used to compute different representations for the start and end of the spans (equations 3, 4) because of dissimilar contexts of the start and end of entities. This allows the models to (1) extract relevant information from recurrent output states; (2) reduce the dimensions to avoid the risk of overfitting and low computation speed; (3) distinguish the start and end indices from the single recurrent output. Finally, a scoring tensor T with the size of n ×n ×c is computed through a biaffine classifier, where n is the length of the sentence and c is the number of NE categories +1 (for non-entity) (equation 5).

$$M(\text{start}) = \text{MLP}(\text{start})(B) \quad (3)$$

$$M(\text{end}) = \text{MLP}(\text{end})(B) \quad (4)$$

$$T = M(\text{end})D(M(\text{start}))T \quad (5)$$

where $D \in R^{d \times c \times d}$ is learned parameters and d is the output size of the two MLP layers.

With the constraint that $s_i \leq e_i$ (the start of entity is before its end), the final tensor provides scores for all spans that could form a named entity. Each span with start/end index s/e is then assigned a category c with the highest score:

$$y(s, e) = \arg\max_c T(s, e, c) \quad (6)$$

Since regular expression (regex) is an effective tool to catch tokens with specific patterns, we use it to assist the model in matching some NER categories. To the best of our knowledge and from our observations, some categories have well-defined structures, such as EMAIL and URL. Using regexes to recognize such types of entities is a common approach, which is powerful and cost-effective. Table 1 summarizes some regexes used in our system. All entities predicted by the model will be considered and selected in the post-processing step.

Post-processing: All the spans having the category other than non-entity are ranked according to their scores in descending order. An entity i will be selected if there is no entity j in the set of higher-ranked entities such that $s_i < s_j \leq e_i < e_j$ or $s_j < s_i \leq e_j < e_i$. Since the predictions from regexes have no score, they are considered to be in lower priority. The reason is that regexes could be effective to search tokens in a string, but they ignore the surrounding context.

3.2. Co-teaching+

Co-teaching+ [7] is a learning paradigm that simultaneously trains two separate networks and updates their parameters by prediction disagreement. First, two networks predict all data, but only consider the predictions that differ from the two networks. Then, from the disagreement data, each network selects its small-loss data to optimize and update its peer network. Figure 2 illustrates the main flow of this technique.

Based on the observation that the gold labels are slightly noisy, where some entities

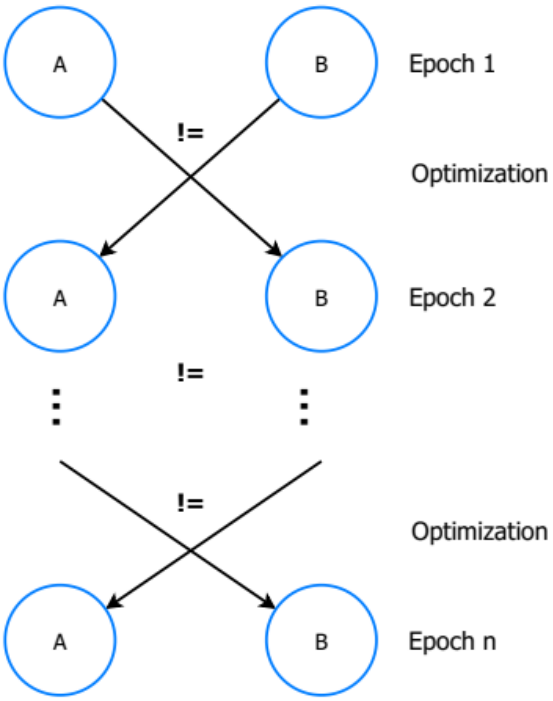

Figure 2. Coteaching+ flow.

are wrongly assigned, such as the disagreement in the rule of labeling LOCATION and LOCATION-GPE, we decide to adopt this training strategy. We train two networks with different learning rates, called η1, η2, and define a forget rate α that decides the rate to ignore high-loss data from two predictions. This allows two networks to predict different results, and accordingly optimize their parameters effectively.

3.3. Ensemble algorithm

In our work, we try to combine two predictions to make an ensemble model by proposing an algorithm to merge results from two models. Algorithm 1 shows steps to ensemble two predictions to generate the final result. It should be noted that we have to do post-processing steps described in subsection 3.1 to avoid violating the constraints for nested entities.

Table 1. Regular expressions used to assist the model to capture some categories:

Categories	Regular expressions
DATETIME-TIME	\d{1,2}(h|[h:]\d{1,2})
EMAIL	[\w]+@[a-zA-Z][\w\.]+\.[a-zA-Z]{2,}
IP	\d{3}(\.\d{1,3}){3}
QUANTITY-PER	(-)?\d+([,\.]\d+)?%
URL	(https?\://)?[a-z]+(\.\w+)+(/[\w\.-]+)*(/)?
QUANTITY-AGE	\d+\stuổi
DATETIME-DATE	(ngày\s|hôm\s)?\d{1,2}[\.-/]\d{1,2}[\.-/]\d{4}

At first, with two sets of predictions t1, t2 from two networks, we compute common categories between t1 and t2. In addition, we also get common spans from the two sets with different categories. For each span (s, e) and category c in the union, but not in the intersection of t1 and t2, if the span is in the common set, the category c′ which has the higher probability is chosen and added to the final result; otherwise, only the category with a probability higher than a threshold τ will be accepted.

The value τ plays an important role in the performance of our model. Setting too high τ results in low recall while too low τ damages the precision. By default, 0.5 is a reasonable value since it is not too high nor too low.

4. Experiments and Results

4.1. Data Preprocessing and Analysis

Since the data are provided with only raw texts, we use VnCoreNLP [13] to do sentence segmentation, word segmentation and POS tagging.

In development, we train models on training data and evaluate the validation set. The training corpus contains 16,052 sentences and the validation consists of 8,736 sentences. In both

sets, the length of the sentences ranges from 1 to 169. In addition, from our analysis, the label distribution in the data set is pretty unbalanced, where 3 in 41 types of categories, namely PERSON, ORGANIZATION, and LOCATION-GPE, occupy nearly 50%. The details are given in Table 3.

Algorithm 1 Ensemble algorithm

Input: Two sets of predictions t_1, t_2
$common \leftarrow t_1 \cap t_2$
$span \leftarrow \{(s,e) | x \in t_1 \wedge y \in t_2 \wedge s = x_s = y_s \wedge e = x_e = y_e\}$
$add \leftarrow \emptyset$
for $(s,e,c) \in (t_1 \cup t_2) \setminus common$ **do**
  if $(s,e) \in span$ **then**
    Choose category c' at span (s,e) in t_1, t_2 with higher probability
    $add \leftarrow add \cup \{(s,e,c')\}$
  else
    Let p be the probability that c is the category at span (s,e) ▷ Accept the prediction if p is larger than a threshold
    if $p > \tau$ **then**
      $add \leftarrow add \cup \{(s,e,c)\}$
    end if
  end if
end for
return $common \cup add$

We have tried to apply Focal Loss [14] to deal with this issue. However, it does not obtain better results on the validation set after the balance step. Table 2 compares the performances at the nested level on the validation data of our proposed systems from different approaches and the one with focal loss applied. The result of the ensemble method is computed by a combination of two predictions from the Biaffine model and Co-teaching+ training strategy. Following the evaluation results on the validation set, we decide not to apply the focal loss to our systems since its performance in F1 score is not as high as others.

Table 2. Evaluation results on validation set. The main measurement to evaluate NER system is F1 score computed by the formula F1 = 2 × P × R/(P + R). Precision (P) is the proportion of NEs correctly recognized by the system. Recall (R) is the percentage of NEs correctly retrieved in the gold data:

Method	P	R	F1
Biaffine	70.5	61.1	65.4
Co-teaching+	67.2	63.6	65.4
Ensemble	66.1	65.7	65.9
Biaffine w/ Focal Loss	71.1	59.0	64.5

Table 3. Label distribution in both training and validation sets:

Category	Train	Validation
PERSON	5,936	3,485
ORGANIZATION	3,632	2,072
LOCATION-GPE	5,028	1,776
Others	15,622	8,541
Total	29,862	15,864

4.2. Experimental Setup

The experiments are carried out on Google Colab Pro, with the GPU NVIDIA Tesla P100. Table 4 shows the hyper-parameters used for training models. We follow the original configuration proposed by the author of [11], except embedding size. By increasing the size of word embeddings, we can capture more useful information from the words. However, too high embedding size will not enhance the performance much, but take long training and inference time because it requires heavy computation. In our experiments, we change the size to 200 since it could balance these two factors, consuming acceptable training time and giving good enough results. The version of PhoBERT is PhoBERT-base [12] since it is lighter, accordingly memory-efficient, reducing the training time. Due to time limitations, we intuitively choose the forget rate α without testing on different values and analyzing to choose the best value.

4.3. Results

At the VLSP 2021 Evaluation Campaign, we submit 3 results from 3 approaches to the organizers for evaluating the performance of our models as follows.

- Submission 1: the Biaffine model.

- Submission 2: the model with higher evaluation result on the validation set from the proposed training strategy.
- Submission 3: ensemble of two predictions from submission 1 and submission 2.

Table 4. Hyper-parameters used in training models:

Hyper-Parameters	Value
Max sequence length	170
LSTM hidden states	400
LSTM layers	3
Embedding size	200
MLP size	500
Learning rate η_1	0.002
Learning rate η_2	0.001
Forget rate α	0.2
Threshold τ	0.5
Dropout rate	0.33

Table 5 shows the evaluation results obtained on the private test set at VLSP 2021. It is clear that the proposed ensemble method achieves the best results (F1) among 3 submissions at all evaluation levels.

To be more specific, although the F1 scores from the Biaffine model and from training by Co-teaching+ strategy are nearly the same, each system has its strength. The Biaffine model gives the highest precision at all evaluation levels (66.43%, 64.48% and 65.42% for top-level, nested and overall respectively), while the recall is still low, 56.32% for the best. In contrast, when training with the proposed strategy, we achieve better recall for all evaluation levels, increasing by 1.5-1.7%. However, submission 3 produced by the ensemble method outperforms both submissions 1 and 2, improving the F1 score up to 62.55%.

From the evaluation results, we show the effectiveness of the ensemble method in the task, achieving the highest F1 score. The reason might be that it combines the predictions, inheriting the strengths from two models. While the baseline Biaffine model is better at precision, the model trained with Co-teaching+ strategy achieves higher recall. By combining two systems together, the ensemble model could balance the precision and recall, leading to better results.

Table 5. Official results from VLSP organizers:

Model		P	R	F1
Biaffine (1)	Top-level	66.43	56.32	60.96
	Nested	64.48	50.43	56.60
	Overall	65.42	53.15	58.65
Co-teaching+ (2)	Top-level	64.69	58.02	61.17
	Nested	61.31	51.97	56.26
	Overall	62.92	54.77	58.56
Ensemble (1 + 2)	Top-level	65.98	59.46	**62.55**
	Nested	62.17	53.44	**57.48**
	Overall	63.98	56.23	**59.85**

5. Conclusion

We have presented our work for the task of Vietnamese named entity recognition at VLSP 2021 Evaluation Campaign. We introduce a span-based model which has not been used for the Vietnamese NER task, and apply a training strategy to deal with the problem of noisy labels. We also propose an ensemble algorithm to combine two predictions. The results show that the ensemble model gives the best performance, achieving 62.55%, 57.48% and 59.85% for top-level, nested and overall evaluation respectively. In future work, we plan to study other methods to deal with noisy labels for the task as well as to design more effective architecture. Furthermore, we intend to do an ablation study and error analysis to explore how our proposed systems do better or worse than the base model to improve the quality of NER for the Vietnamese language.

References

[1] R. Grishman, B. M. Sundheim, Message Understanding Conference - 6: A Brief History , in: COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics, 1996.

[2] E. F. Sang, F. De Meulder, Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition, arXiv preprint cs/0306050.

[3] N. T. M. Huyen, V. X. Luong, VLSP 2016 Shared Task: Named Entity Recognition, Proceedings of Vietnamese Speech and Language Processing ,VLSP, 2016.
[4] H. T. Nguyen, Q. T. Ngo, L. X. Vu, V. M. Tran, H. T. Nguyen, VLSP Shared Task: Named Entity Recognition, Journal of Computer Science and Cybernetics, Vol. 34, No. 4, 2018, pp. 283–294.
[5] H. M. Linh, D. D. Dao, N. T. M. Huyen, N. T. Quyen, D. X. Dung, NER Challenge: Named Entity Recognition for Vietnamese, VLSP, 2021.
[6] J. Yu, B. Bohnet, M. Poesio, Named Entity Recognition as Dependency Parsing, arXiv preprint arXiv:2005.07150.
[7] X. Yu, B. Han, J. Yao, G. Niu, I. Tsang, Sugiyama, How does Disagreement Help Generalization against Label Corruption?, in: International Conference on Machine Learning, PMLR, 2019, pp. 7164–7173.
[8] P. Q. N. Minh, A Feature-Based Model for Nested Named-Entity Recognition at VLSP-2018 NER Evaluation Campaign, arXiv preprint arXiv:1803.08463.
[9] V. Luong, L. Pham, ZA-NER: Vietnamese Named Entity Recognition at VLSP 2018 Evaluation Campaign, in: The Fifth International Workshop on Vietnamese Language and Speech Processing, VLSP 2018, 2018.
[10] K. A. Nguyen, N. Dong, C.-T. Nguyen, Attentive Neural Network for Named Entity Recognition in Vietnamese, in: 2019 IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF), IEEE, 2019, pp. 1–6.
[11] T. Dozat, C. D. Manning, Deep Biaffine Attention for Neural Dependency Parsing, arXiv preprint arXiv:1611.01734.
[12] D. Q. Nguyen, A. T. Nguyen, PhoBERT: Pre-trained language models for Vietnamese, arXiv preprint arXiv:2003.00744.
[13] T. Vu, D. Q. Nguyen, D. Q. Nguyen, M. Dras, Johnson, VnCoreNLP: A Vietnamese Natural Language Processing Toolkit, arXiv preprint arXiv:1801.01331.
[14] T.Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal Loss for Dense Object Detection, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988.