

# The Importance of Automatic Syntactic Features in Vietnamese Named Entity Recognition

**Thai-Hoang Pham**

R&D Department

Alt Inc

Hanoi, Vietnam

phamthaihoang.hn@gmail.com

**Phuong Le-Hong**

College of Science

Vietname National University in Hanoi

Hanoi, Vietnam

phuonglh@vnu.edu.vn

## Abstract

This paper presents a state-of-the-art system for Vietnamese Named Entity Recognition (NER). By incorporating automatic syntactic features with word embeddings as input for bidirectional Long Short-Term Memory (Bi-LSTM), our system, although simpler than some deep learning architectures, achieves a much better result for Vietnamese NER. The proposed method achieves an overall  $F_1$  score of 92.05% on the test set of an evaluation campaign, organized in late 2016 by the Vietnamese Language and Speech Processing (VLSP) community. Our named entity recognition system outperforms the best previous systems for Vietnamese NER by a large margin.

## 1 Introduction

Named entity recognition (NER) is an essential task in natural language processing that falls under the domain of information extraction. The function of this task is to identify noun phrases and categorize them into a predefined class. NER is a crucial pre-processing step used in some NLP applications such as question answering, automatic translation, speech processing, and biomedical science. In two shared tasks, CoNLL 2002<sup>1</sup> and CoNLL 2003<sup>2</sup>, language independent NER systems were evaluated for English, German, Spanish, and Dutch. These systems focus on four named entity types namely person, organization, location, and remaining miscellaneous entities.

<sup>1</sup><http://www.cnts.ua.ac.be/conll2002/ner/>

<sup>2</sup><http://www.cnts.ua.ac.be/conll2003/ner/>

Lately, an evaluation campaign that systematically compared NER systems for the Vietnamese language has been launched by the Vietnamese Language and Speech Processing (VLSP)<sup>3</sup> community. They collect data from electronic newspapers on the web and annotate named entities in this corpus. Similar to the CoNLL 2003 share task, there are four named entity types in VLSP dataset: person (PER), organization (ORG), location (LOC), and miscellaneous entity (MISC).

In this paper, we present a state-of-the-art NER system for Vietnamese language that uses automatic syntactic features with word embedding in Bi-LSTM. Our system outperforms the leading system of the VLSP campaign utilizing a number of syntactic and hand-crafted features, and an end-to-end system described in (Pham and Le-Hong, 2017) that is a combination of Bi-LSTM, Convolutional Neural Network (CNN), and Conditional Random Field (CRF) about 3%. To sum up, the overall  $F_1$  score of our system is 92.05% as assessed by the standard test set of VLSP. The contributions of this work consist of:

- We demonstrate a deep learning model reaching the state-of-the-art performance for Vietnamese NER task. By incorporating automatic syntactic features, our system (Bi-LSTM), although simpler than (Bi-LSTM-CNN-CRF) model described in (Pham and Le-Hong, 2017), achieves a much better result on Vietnamese NER dataset. The simple architecture also contributes to the feasibility of our system in

<sup>3</sup><http://vlsp.org.vn/>

practice because it requires less time for inference stage. Our best system utilizes part-of-speech, chunk, and regular expression type features with word embeddings as an input for two-layer Bi-LSTM model, which achieves an  $F_1$  score of 92.05%.

- We demonstrate the greater importance of syntactic features in Vietnamese NER compared to their impact in other languages. Those features help improve the  $F_1$  score of about 18%.
- We also indicate some network parameters such as network size, dropout are likely to affect the performance of our system.
- We conduct a thorough empirical study on applying common deep learning architectures to Vietnamese NER, including Recurrent Neural Network (RNN), unidirectional and bidirectional LSTM. These models are also compared to conventional sequence labelling models such as Maximum Entropy Markov models (MEMM).
- We publicize our NER system for research purpose, which is believed to positively contributing to the long-term advancement of Vietnamese NER as well as Vietnamese language processing.

The remainder of this paper is structured as follows. Section 2 summarizes related work on NER. Section 3 describes features and model used in our system. Section 4 gives experimental results and discussions. Finally, Section 5 concludes the paper.

## 2 Related Works

We categorize two main approaches for NER in a large number of research published in the last two decades. The first approach is characterized by the use of traditional sequence labelling models such as CRF, hidden markov model, support vector machine, maximum entropy that are heavily dependent on hand-crafted features (Florian et al., 2003; Lin and Wu, 2009; Durrett and Klein, 2014; Luo and Xiaojiang Huang, 2015). These systems made an endeavor to exploit external information instead of the available training data such as gazetteers and unannotated data.

In the last few years, deep neural network approaches have gained in popularity dealing with NER task. With the advance of computational power, there has been more and more research that applied deep learning methods to improve performances of their NLP systems. LSTM and CNN are prevalent models used in these architectures. Firstly, (Collobert et al., 2011) used a CNN over a sequence of word embeddings with a CRF layer on the top. They nearly achieved state-of-the-art results on some sequence labelling tasks such as POS tagging, chunking, albeit did not work for NER. To improve the accuracy for recognizing named entities, (Huang et al., 2015) used Bi-LSTM with CRF layer for joint decoding. This model also used hand-crafted features to ameliorate its performance. Recently, (Chiu and Nichols, 2016) proposed a hybrid model that combined Bi-LSTM with CNN to learn both character-level and word-level representations. Instead of using CNN to learn character-level features like (Chiu and Nichols, 2016), (Lample et al., 2016) used BI-LSTM to capture both character and word-level features.

For Vietnamese, VLSP community has organized an evaluation campaign that follows the rules of CoNLL 2003 shared task to systematically compare NER systems. Participating systems have approached this task by both traditional and deep learning architectures. In particular, the first-rank system of the VLSP campaign which achieved an  $F_1$  score of 88.78% used MEMM with many hand-crafted features (Le-Hong, 2016). Meanwhile, (Nguyen et al., 2016) adopted deep neural networks for this task. They used the system provided by (Lample et al., 2016), which consists of two types of LSTM models: Bi-LSTM-CRF and Stack-LSTM. Their best system achieved an  $F_1$  score of 83.80%. More recently, (Pham and Le-Hong, 2017) used an end-to-end system that is a combination of Bi-LSTM-CNN-CRF for Vietnamese NER. The  $F_1$  score of this system is 88.59% that is competitive with the accuracy of (Le-Hong, 2016).

### 3 Methodology

#### 3.1 Feature Engineering

**Word Embeddings** We use a word embedding set trained from 7.3GB of 2 million articles collected through a Vietnamese news portal<sup>4</sup> by word2vec<sup>5</sup> toolkit. Details of this word embedding set are described in (Pham and Le-Hong, 2017).

**Automatic Syntactic Features** To ameliorate a performance of our system, we incorporate some syntactic features with word embeddings as input for Bi-LSTM model. These syntactic features are generated automatically by some public tools so the actual input of our system is only raw texts. These additional features consist of part-of-speech (POS) and chunk tags that are available in the dataset, and regular expression types that capture common organization and location names. These regular expressions over tokens described particularly in (Le-Hong, 2016) are shown to provide helpful features for classifying candidate named entities, as shown in the experiments.

#### 3.2 Long Short-Term Memory

Long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) is a special kind of Recurrent Neural Network (RNN) which is capable of dealing with possible gradient exploding and vanishing problems (Bengio et al., 1994; Pascanu et al., 2013) when handling long-range sequences. It is because LSTM uses memory cells instead of hidden layers in a standard RNN. In particular, there are three multiplicative gates in a memory cell unit that decides on the amount of information to pass on to the next step. Therefore, LSTM is likely to exploit long-range dependency data. Each multiplicative gate is computed as follows:

$$\begin{aligned} \mathbf{i}_t &= \sigma(\mathbf{W}_i \mathbf{h}_{t-1} + \mathbf{U}_i \mathbf{x}_t + \mathbf{b}_i) \\ \mathbf{f}_t &= \sigma(\mathbf{W}_f \mathbf{h}_{t-1} + \mathbf{U}_f \mathbf{x}_t + \mathbf{b}_f) \\ \mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tanh(\mathbf{W}_c \mathbf{h}_{t-1} + \mathbf{U}_c \mathbf{x}_t + \mathbf{b}_c) \\ \mathbf{o}_t &= \sigma(\mathbf{W}_o \mathbf{h}_{t-1} + \mathbf{U}_o \mathbf{x}_t + \mathbf{b}_o) \\ \mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \end{aligned}$$

<sup>4</sup><http://www.baomoi.com>

<sup>5</sup><https://code.google.com/archive/p/word2vec/>

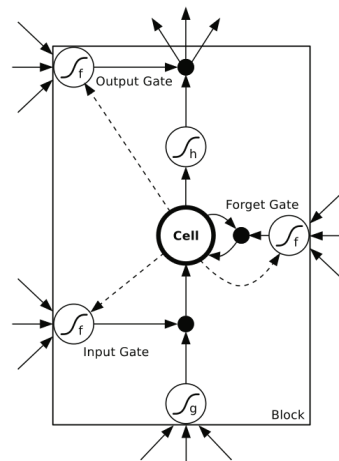


Figure 1: LSTM memory cell

where  $\sigma$  and  $\odot$  are element-wise sigmoid function and element-wise product,  $\mathbf{i}$ ,  $\mathbf{f}$ ,  $\mathbf{o}$  and  $\mathbf{c}$  are the input gate, forget gate, output gate and cell vector respectively.  $\mathbf{U}_i, \mathbf{U}_f, \mathbf{U}_c, \mathbf{U}_o$  are weight matrices that connect input  $\mathbf{x}$  and gates, and  $\mathbf{U}_i, \mathbf{U}_f, \mathbf{U}_c, \mathbf{U}_o$  are weight matrices that connect gates and hidden state  $\mathbf{h}$ , and finally,  $\mathbf{b}_i, \mathbf{b}_f, \mathbf{b}_c, \mathbf{b}_o$  are the bias vectors. Figure 1 illustrates a single LSTM memory cell.

#### 3.3 Bidirectional Long Short-Term Memory

The original LSTM uses only past features. For many sequence labelling tasks, it is beneficial when accessing both past and future contexts. For this reason, we utilize the bidirectional LSTM (Bi-LSTM) (Graves and Schmidhuber, 2005; Graves et al., 2013) for NER task. The basic idea is running both forward and backward passes to capture past and future information, respectively, and concatenate two hidden states to form a final representation. Figure 2 illustrates the backward and forward passes of Bi-LSTM.

#### 3.4 Our Deep Learning Model

For Vietnamese named entity recognition, we use a 2-layer Bi-LSTM with softmax layer on the top to detect named entities in sequence of sentences. The inputs are the combination of word and syntactic features, and the outputs are the probability distributions over named entity tags. Figure 3 describes the details of our deep learning model. In the next sections, we present our experimental results.

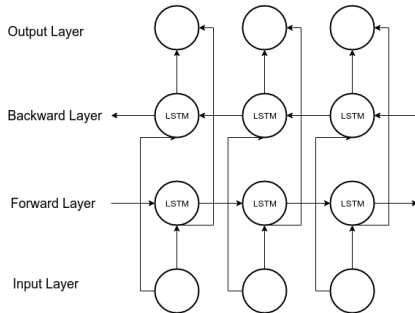


Figure 2: Bidirectional LSTM

## 4 Results And Discussions

### 4.1 VLSP Corpus

We conduct experiments on the VSLP NER shared task 2016 corpus. Four named entity types are evaluated in this corpus including person, location, organization, and other named entities. Definitions of these entity types match with their descriptions in the CoNLL shared task 2003.

There are five columns in this dataset including surface word, automatic POS and chunking tags, named entity and nested named entity labels, of which the first four columns conform to the format of the CoNLL 2003 shared task. We do not use the fifth column because our system focuses on only named entity without nesting. Named entities are labelled by the IOB notation as in the CoNLL 2003 shared tasks. In particular, there are 9 named entity labels in this corpus including B-PER and I-PER for persons, B-ORG and I-ORG for organizations, B-LOC and I-LOC for locations, B-MISC and I-MISC for other named entities, and O for other elements. Table 1 presents the number of annotated named entities in the training and testing set.

Entity Types	Training Set	Testing Set
Location	6,247	1,379
Organization	1,213	274
Person	7,480	1,294
Miscellaneous names	282	49
All	15,222	2,996

Table 1: Statistics of named entities in VLSP corpus

Because we use early stopping method described in (Graves et al., 2013) to avoid overfitting when training our neural network models, we hold one

part of training data for validation. The number of sentences of each part of VLSP corpus is described in Table 2.

Data sets	Number of sentences
Train	14,861
Dev	2,000
Test	2,831

Table 2: Size of each data set in VLSP corpus

### 4.2 Evaluation Method

We evaluate the performance of our system with  $F_1$  score:

$$F_1 = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

Precision and recall are the percentage of correct named entities identified by the system and the percentage of identified named entities present in the corpus respectively. To compare fairly with previous systems, we use an available evaluation script provided by the CoNLL 2003 shared task<sup>6</sup> to calculate  $F_1$  score of our NER system.

### 4.3 Results

In this section, we analyze the efficiency of word embeddings, bidirectional learning, model configuration, and especially automatic syntactic features.

**Embeddings** To evaluate the effectiveness of word embeddings, we compare the systems on three types of input: skip-gram, random vector, and one-hot vector.

The number of dimensions we choose for word embedding is 300. We create random vectors for words that do not appear in word embeddings set by uniformly sampling from the range  $[-\sqrt{\frac{3}{dim}}, +\sqrt{\frac{3}{dim}}]$  where  $dim$  is the dimension of embeddings. For random vector setting, we also sample vectors for all words from this distribution. The performances of the system with each input type are represented in Table 3.

We can conclude that word embedding is an important factor of our model. Skip-gram vector significantly improves our performance. The improve-

<sup>6</sup><http://www.cnts.ua.ac.be/conll2003/ner/>

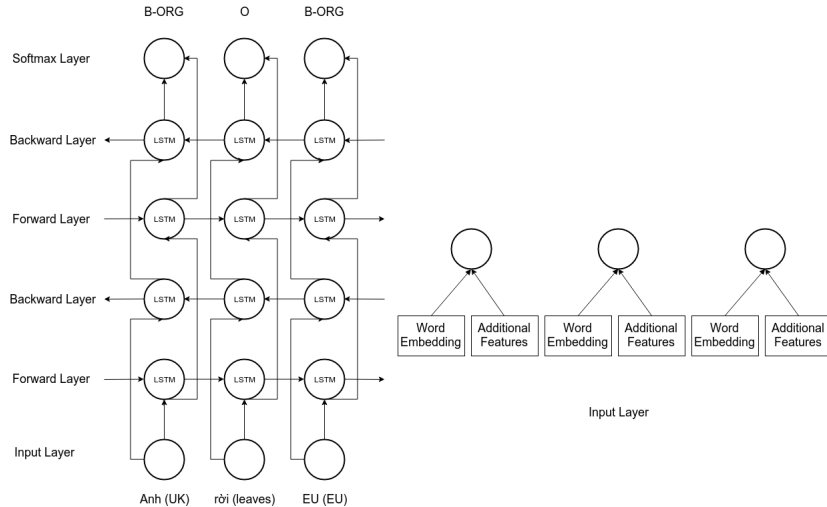


Figure 3: Our deep learning model

Entity	Skip-Gram			Random			One-hot		
	Pre.	Rec.	$F_1$	Pre.	Rec.	$F_1$	Pre.	Rec.	$F_1$
LOC	83.63	82.48	83.05	79.39	66.37	72.26	79.21	72.37	75.63
MISC	84.14	78.37	81.07	65.23	69.80	76.70	82.14	46.94	59.74
ORG	49.85	50.51	50.07	35.19	19.56	25.11	30.56	12.04	17.28
PER	72.77	65.73	69.06	70.76	50.35	58.83	69.13	52.09	59.41
ALL	75.88	72.26	<b>74.02</b>	72.99	55.23	62.87	57.68	72.88	64.39

Table 3: Performance of our model on three input types

ment is about 11% when using skip-gram vectors instead of random vectors. Thus, we use skip-gram vectors as inputs for our system.

**Effect of Bidirectional Learning** In the second experiment, we examine the benefit of accessing both past and future contexts by comparing the performances of RNN, LSTM and Bi-LSTM models. In this task, RNN model fails because it faces the gradient vanishing/exploding problem when training with long-range dependencies (132 time steps), leading to the unstable value of the cost functions. For this reason, only performances of LSTM and Bi-LSTM models are shown in Table 4.

Entity	Bi-LSTM			LSTM		
	Pre.	Rec.	$F_1$	Pre.	Rec.	$F_1$
LOC	83.63	82.48	83.05	74.60	77.38	75.96
MISC	84.14	78.37	81.07	2.15	2.04	2.09
ORG	49.85	50.51	50.07	32.22	34.60	33.60
PER	72.77	65.73	69.06	67.95	60.73	64.12
ALL	75.88	72.26	<b>74.02</b>	66.61	65.04	65.80

Table 4: Performance of our model when using one and two layers

We see that learning both past and future contexts is very useful for NER. Performances of all of the entity types are increased, especially for ORG and MISC. The total accuracy is improved greatly, from 65.80% to 74.02%.

**Number of Bi-LSTM Layers** In the third experiment, we investigate the improvement when adding more Bi-LSTM layers. Table 5 shows the accuracy when using one or two Bi-LSTM layers. We observe a significant improvement when using two layers of Bi-LSTM. The performance is increased from 71.74% to 74.02%.

Entity	Two layers			One layer		
	Pre.	Rec.	$F_1$	Pre.	Rec.	$F_1$
LOC	83.63	82.48	83.05	82.22	80.64	81.41
MISC	84.14	78.37	81.07	85.15	74.29	79.32
ORG	49.85	50.51	50.07	44.10	40.88	42.39
PER	72.77	65.73	69.06	72.70	62.15	66.91
ALL	75.88	72.26	<b>74.02</b>	74.83	68.91	71.74

Table 5: Performance of our model when using one and two layers

**Effect of Dropout** In the fourth experiment, we compare the results of our model with and without dropout layers. The optimal dropout ratio for our experiments is 0.5. The accuracy with dropout is 74.02%, compared to 68.27% without dropout. It proves the effectiveness of dropout for preventing overfitting.

Entity	Dropout = 0.5			Dropout = 0.0		
	Pre.	Rec.	$F_1$	Pre.	Rec.	$F_1$
LOC	83.63	82.48	83.05	80.98	76.79	78.79
MISC	84.14	78.37	81.07	84.09	64.49	72.73
ORG	49.85	50.51	50.07	41.09	32.92	36.43
PER	72.77	65.73	69.06	67.35	59.23	62.97
ALL	75.88	72.26	<b>74.02</b>	71.97	64.99	68.27

Table 6: Performance of our model with and without dropout

**Syntactic Features Integration** As shown in the previous experiments, using only word features in deep learning models is not enough to achieve the state-of-the-art result. In particular, the accuracy of this model is only 74.02%. This result is far lower in comparison to that of state-of-the-art systems for Vietnamese NER. In the following experiments, we add more useful features to enhance the performance of our deep learning model. Table 7 shows the improvement when adding part-of-speech, chunk, case-sensitive, and regular expression features.

Features	Pre.	Rec.	$F_1$
Word	75.88	72.26	74.02
Word+POS	84.23	87.64	85.90
Word+Chunk	90.73	83.18	86.79
Word+Case	83.68	84.45	84.06
Word+Regex	76.58	71.86	74.13
Word+POS+Chunk+Case+Regex	90.25	92.55	91.39
Word+POS+Chunk+Regex	91.09	93.03	<b>92.05</b>

Table 7: Performance of our model when adding more features

As seen in this table, adding each of these syntactic features helps improve the performance significantly. The best result we get is adding part-of-speech, chunk and regular expression features. The accuracy of this final system is 92.05% that is much higher than 74.02 of the system without using syntactic features. An explanation for this problem is possibly a characteristic of Vietnamese. In particular, Vietnamese named entities are often a noun phrase chunk.

**Comparison with Other Languages** In the sixth experiment, we want to compare the role of syntactic features for NER task in other languages. For this reason, we run our system on CoNLL 2003 data set for English. The word embedding set we use for En-

glish is pre-trained by Glove model and is provided by the authors<sup>7</sup>. Table 8 shows the performances of our system when adding part-of-speech and chunk features.

Features	Vietnamese			English		
	Pre.	Rec.	$F_1$	Pre.	Rec.	$F_1$
Word	75.88	72.26	74.02	87.39	89.66	88.51
Word + POS + Chunk	90.39	92.59	91.48	87.08	89.59	88.31

Table 8: The importance of syntactic features for Vietnamese compared to it for English

For English NER task, adding the syntactic features does not help to improve the performance of our system. Thus, we can conclude that syntactic features have the greater importance in Vietnamese NER compared to their impact in English.

**Comparison with Previous Systems** In VLSP 2016 workshop, there are several different systems proposed for Vietnamese NER. These systems focus on only three entities types *LOC*, *ORG*, and *PER*. For the purpose of fairness, we evaluate our performances based on these named entity types on the same corpus. The accuracy of our best model over three entity types is 92.02%, which is higher than the best participating system (Le-Hong, 2016) in that shared task about 3.2%.

Moreover, (Pham and Le-Hong, 2017) used a combination of Bi-LSTM, CNN, and CRF that achieved the same performance with (Le-Hong, 2016). This system is end-to-end architecture that required only word embeddings while (Le-Hong, 2016) used many syntactic and hand-crafted features with MEMM. Our system surpasses both of these systems by using Bi-LSTM with automatically syntactic features, which takes less time for training and inference than Bi-LSTM-CNN-CRF model and does not depend on many hand-crafted features as MEMM. Table 9 presents the accuracy of each system.

Models	Types	$F_1$
(Le-Hong, 2016)	ME	88.78
(Pham and Le-Hong, 2017)	Bi-LSTM-CNN-CRF	88.59
Our model	Bi-LSTM	92.02

Table 9: Performance of our model and previous systems

<sup>7</sup><https://nlp.stanford.edu/projects/glove/>

## 5 Conclusion

In this work, we have presented a state-of-the-art named entity recognition system for the Vietnamese language, which achieves an  $F_1$  score of 92.05% on the standard dataset published by the VLSP community. Our system outperforms the first-rank system of the related NER shared task with a large margin, 3.2% in particular. We have also shown the effectiveness of using automatic syntactic features for Bi-LSTM model that surpass the combination of Bi-LSTM-CNN-CRF models albeit requiring less time for computation.

## Acknowledgement

The second author is partly funded by the Vietnam National University, Hanoi (VNU) under project number QG.15.04. Any opinions, findings and conclusion expressed in this paper are those of the authors and do not necessarily reflect the view of VNU.

## References

- Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.
- Jason P.C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370.
- Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.
- Greg Durrett and Dan Klein. 2014. A joint model for entity analysis: Coreference, typing, and linking. *Transactions of the Association for Computational Linguistics*, 2:477–490.
- Radu Florian, Abe Ittycheriah, Hongyan Jing, and Tong Zhang. 2003. Named entity recognition through classifier combination. In Walter Daelemans and Miles Osborne, editors, *Proceedings of CoNLL-2003*, pages 168–171. Edmonton, Canada.
- Alex Graves and Jürgen Schmidhuber. 2005. Frame-wise phoneme classification with bidirectional lstm networks. In *Proceedings of 2005 IEEE International Joint Conference on Neural Networks*, volume 4, pages 2047–2052, Montreal, QC, Canada. IEEE.
- Alex Graves, Abdel rahmand Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Proceedings of 2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649, Vancouver, BC, Canada. IEEE.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.
- Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.
- Phuong Le-Hong. 2016. Vietnamese named entity recognition using token regular expressions and bidirectional inference. In *Proceedings of The Fourth International Workshop on Vietnamese Language and Speech Processing*, Hanoi, Vietnam.
- Dekang Lin and Xiaoyun Wu. 2009. Phrase clustering for discriminative learning. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, volume 2, pages 1030–1038. Association for Computational Linguistics.
- Gang Luo and Zaiqing Nie Xiaojiang Huang, Chin-Yew Lin. 2015. Joint entity recognition and disambiguation. In *Proceedings of the 2015 Conference on Empirical Methods on Natural Language Processing*, pages 879–888. Association for Computational Linguistics.
- Truong Son Nguyen, Le Minh Nguyen, and Xuan Chien Tran. 2016. Vietnamese named entity recognition at vlsp 2016 evaluation campaign. In *Proceedings of The Fourth International Workshop on Vietnamese Language and Speech Processing*, Hanoi, Vietnam.
- Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *The 30th International Conference on Machine Learning*, volume 28, pages 1310–1318, Atlanta, USA.
- Thai-Hoang Pham and Phuong Le-Hong. 2017. End-to-end recurrent neural network models for vietnamese named entity recognition: Word-level vs. character-level. In *Proceedings of The 15th International Conference of the Pacific Association for Computational Linguistics*, pages 251–264.