Error Analysis for Vietnamese Named Entity Recognition on Deep Neural Network Models

Binh An Nguyen, Kiet Van Nguyen, and Ngan Luu-Thuy Nguyen

University of Information Technology,
Vietnam National University - Ho Chi Minh City, Vietnam
nguyenanbinh96@gmail.com, kietnv@uit.edu.vn, ngannlt@uit.edu.vn

Abstract In recent years, Vietnamese Named Entity Recognition (NER) systems have had a great breakthrough when using Deep Neural Network methods. This paper describes the primary errors of the state-of-the-art NER systems on Vietnamese language. After conducting experiments on BLSTM-CNN-CRF and BLSTM-CRF models with different word embeddings on the Vietnamese NER dataset. This dataset is provided by VLSP in 2016 and used to evaluate most of the current Vietnamese NER systems. We noticed that BLSTM-CNN-CRF gives better results, therefore, we analyze the errors on this model in detail. Our error-analysis results provide us thorough insights in order to increase the performance of NER for the Vietnamese language and improve the quality of the corpus in the future works.

1 Introduction

Named Entity Recognition (NER) is one of information extraction subtasks that is responsible for detecting entity elements from raw text and can determine the category in which the element belongs, these categories include the names of persons, organizations, locations, expressions of times, quantities, monetary values and percentages.

The problem of NER is described as follow:

Input: A sentence S consists a sequence of n words: $S = w_1, w_2, w_3, \ldots, w_n$ (w_i: the ith word)

Output: The sequence of n labels $y_1, y_2, y_3, \ldots, y_n$. Each y_i label represents the category which w_i belongs to.

For example, given a sentence:

Input: Giám đốc điều hành Tim Cook của Apple vừa giới thiệu 2 điện thoại iPhone, đồng hồ thông minh mới, lớn hơn ở sự kiện Flint Center, Cupertino.

(Apple CEO Tim Cook introduces 2 new, larger iPhones, Smart Watch at Cupertino Flint Center event)[1]

The algorithm will output:

---

[1] http://sanfrancisco.cbslocal.com/2014/09/09/apple-ceo-tim-cook-introduces-2-new-iphones-at-cupertino-flint-center-event/

Output: ⟨O⟩Giám đốc điều hành⟨O⟩ ⟨PER⟩Tim Cook⟨PER⟩ ⟨O⟩của⟨O⟩ ⟨ORG⟩Apple⟨ORG⟩ ⟨O⟩vừa giới thiệu 2 điện thoại iPhone, đồng hồ thông minh mới, lớn hơn ở sự kiện⟨O⟩ ⟨ORG⟩Flint Center⟨ORG⟩, ⟨LOC⟩Cupertino⟨LOC⟩.

With LOC, PER, ORG is Name of location, person, organization respectively. Note that O means Other (Not a Name entity). We will not denote the O label in the following examples in this article because we only care about name of entities.

In this paper, we analyze common errors of the previous state-of-the-art techniques using Deep Neural Network (DNN) on VLSP Corpus. This may contribute to the later researchers the common errors from the results of these state-of-the-art models, then they can rely on to improve the model.

Section 2 discusses the related works to this paper. We will present a method for evaluating and analyzing the types of errors in Section 3. The data used for testing and analysis of errors will be introduced in Section 4, we also talk about deep neural network methods and pre-trained word embeddings for experimentation in this section. Section 5 will detail the errors and evaluations. In the end is our contribution to improve the above errors.

2   Related work

Previously publicly available NER systems do not use DNN, for example, the MITRE Identification Scrubber Toolkit (MIST) [11], Stanford NER [12], BAN-NER [13] and NERsuite [14]. NER systems for Vietnamese language processing used traditional machine learning methods such as Maximum Entropy Markov Model (MEMM), Support Vector Machine (SVM) and Conditional Random Field (CRF). In particular, most of the toolkits for NER task attempted to use MEMM [6], and CRF [5] to solve this problem.

Nowadays, because of the increase in data, DNN methods are used a lot. They have archived great results when it comes to NER tasks, for example, Guillaume Lample et al with BLSTM-CRF in [4] report 90.94 F1 score, Chiu et al with BLSTM-CNN in [1] got 91.62 F1 score, Xeuzhe Ma and Eduard Hovy with BLSTM-CNN-CRF in [8] achieved F1 score of 91.21, Thai-Hoang Pham and Phuong Le-Hong with BLSTM-CNN-CRF in [16] got 88.59% F1 score. These DNN models are also the state-of-the-art models.

3   Error-analysis method

The results of our analysis experiments are reported in precision and recall over all labels (name of person, location, organization and miscellaneous). The process of analyzing errors has 2 steps:

– **Step 1**: We use two state-of-the-art models including BLSTM-CNN-CRF and BLSTM-CRF to train and test on VLSP's NER corpus. In our experiments, we implement word embeddings as features to the two systems.

– **Step 2**: Based on the best results (BLSTM-CNN-CRF), error analysis is performed based on five types of errors (No extraction, No annotation, Wrong range, Wrong tag, Wrong range and tag), in a way similar to [15], but we analyze on both gold labels and predicted labels (more detail in figure 1 and 2).

A token (an entity name maybe contain more than one word) will be extracted as a correct entity by the model if both of the followings are correct:

1. The length of it (range) is correct: The word beginning and the end is the same as gold data (annotator).
2. The label (tag) of it is correct: The label is the same as in gold data.

If it is not meet two above requirements, it will be the wrong entity (an error). Therefore, we divide the errors into five different types which are described in detail as follows:

1. **No extraction**: The error where the model did not extract tokens as a name entity (NE) though the tokens were annotated as a NE.

    LSTM-CNN-CRF: Việt_Nam
    Annotator: ⟨LOC⟩ Việt_Nam ⟨LOC⟩

2. **No annotation**: The error where the model extracted tokens as an NE though the tokens were not annotated as a NE.

    LSTM-CNN-CRF: ⟨PER⟩ Châu Âu ⟨PER⟩

    Annotator: Châu Âu

3. **Wrong range**: The error where the model extracted tokens as an NE and only the range was wrong. (The extracted tokens were partially annotated or they were the part of the annotated tokens).

    LSTM-CNN-CRF: ⟨PER⟩ Ca_sĩ Nguyễn Văn A ⟨PER⟩
    Annotator:
    Ca_sĩ ⟨PER⟩ Nguyễn Văn A ⟨PER⟩

4. **Wrong tag**: The error where the model extracted tokens as an NE and only the tag type was wrong.

    LSTM-CNN-CRF: Khám phá ⟨PER⟩ Yangsuri ⟨PER⟩
    Annotator:
    Khám phá ⟨LOC⟩ Yangsuri ⟨LOC⟩

5. **Wrong range and tag**: The error where the model extracted tokens as an NE and both the range and the tag type were wrong.

LSTM-CNN-CRF: ⟨LOC⟩ gian_hàng Apple ⟨LOC⟩
Annotator:

<div align="center">gian_hàng ⟨ORG⟩ Apple ⟨ORG⟩</div>

We compare the predicted NEs to the gold NEs (*Fig*.1), if they have the same range, the predicted NE is a correct or **Wrong tag**. If it has different range with the gold NE, we will see what type of wrong it is. If it does not have any overlap, it is a **No extraction**. If it has an overlap and the tag is the same at gold NE, it is a **Wrong range**. Finally, it is a **Wrong range and tag** if it has an overlap but the tag is different. The steps in Fig. 2 is the same at Fig. 1 and the different only is we compare the gold NE to the predicted NE, and **No extraction** type will be **No annotation**.
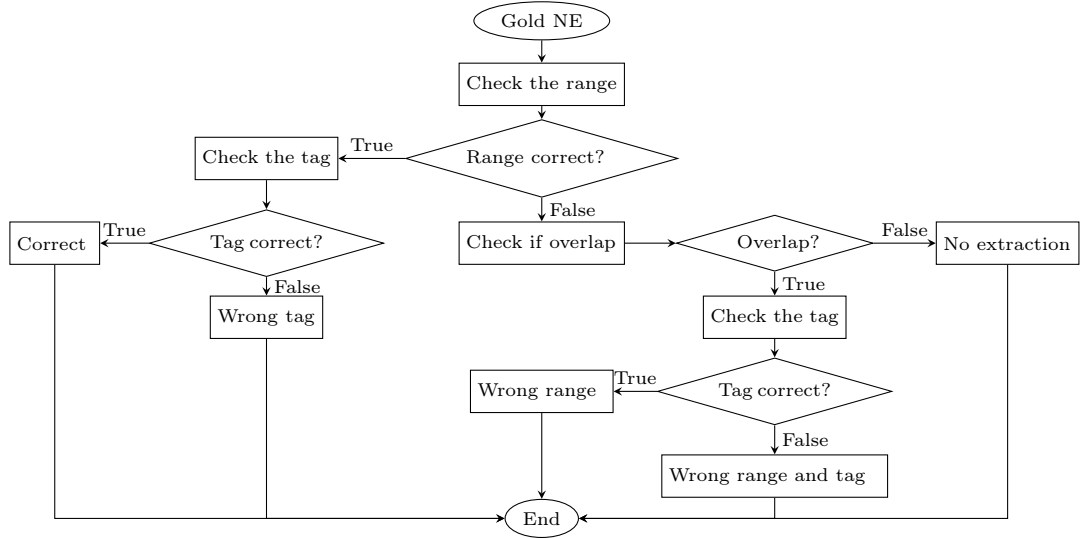

Figure 1. Chart flow to analyze errors based on gold labels

4 Data and model

4.1 Data sets

To conduct error analysis of the model, we used the corpus which are provided by VLSP 2016 - Named Entity Recognition[2]. The dataset contains four different types of label: Location (LOC), Person (PER), Organization (ORG) and Miscellaneous - Name of an entity that do not belong to 3 types above (Table 1).

---

[2] More detail in http://vlsp.org.vn/vlsp2016/eval/ner

Predict NE → Check the range → Range correct?

True → Check the tag → Tag correct?
True → Correct
False → Wrong tag

False → Check if overlap → Overlap?
False → No Annotation
True → Check the tag → Tag correct?
True → Wrong range
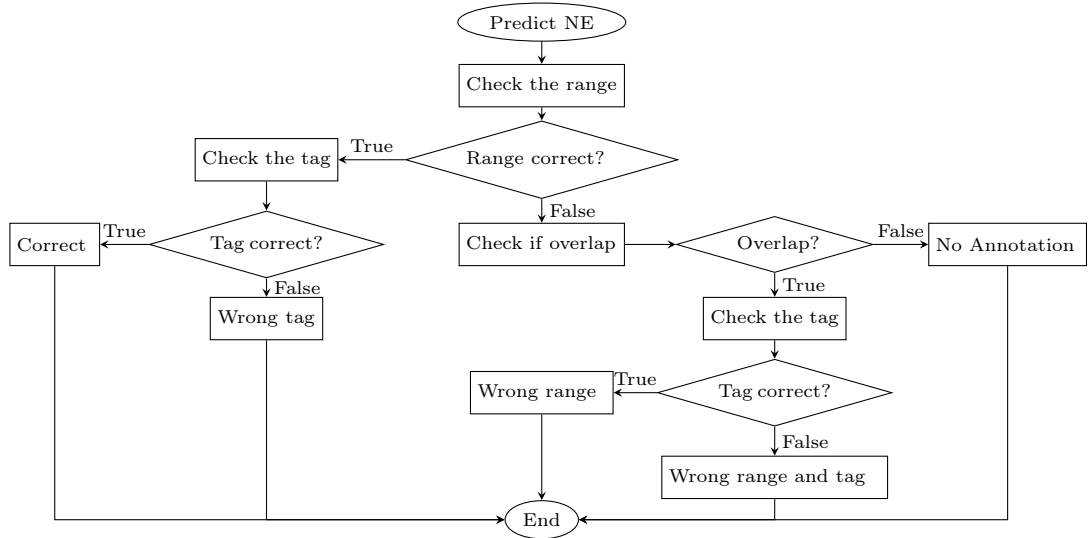False → Wrong range and tag

Figure 2. Chart flow to analyze errors based on predicted labels

Although the corpus has more information about the POS and chunks, but we do not use them as features in our model.

Table 1. Number type of each tags in the corpus

Tags	Number of tag	%
Person	1294	43.22
Location	1379	46.06
Organization	274	9.15
MISC	49	1.64
All Tags	2994	100

There are two folders with 267 text files of training data and 45 text files of test data. They all have their own format. We take 21 first text files and 22 last text files and 22 sentences of the 22th text file and 55 sentences of the 245th text file to be a development data. The remaining files are going to be the training data. The test file is the same at the file VSLP gave. Finally, we have 3 text files only based on the CoNLL 2003 format: train, dev and test.

4.2 Pre-trained word Embeddings

We use the word embeddings for Vietnamese that created by Kyubyong Park[3] and Edouard Grave at al[4]:

Kyubyong Park: In his project, he uses two methods including fastText[5] and word2vec[6] to generate word embeddings from wikipedia database backup dumps[7]. His word embedding is the vector of 100 dimension and it has about 10k words.

Edouard Grave et al [17]: They use fastText tool to generate word embeddings from Wikipedia[8]. The format is the same at Kyubyong's, but their embedding is the vector of 300 dimension, and they have about 200k words

4.3 Model

Based on state-of-the-art methods for NER, BLSTM-CNN-CRF is the end-to-end deep neural network model that achieves the best result on F-score [16]. Therefore, we decide to conduct the experiment on this model and analyze the errors.

We run experiment with the Ma and Hovy (2016) model [8], source code provided by (Motoki Sato)[9] and analysis the errors from this result. Before we decide to analysis on this result, we have run some other methods, but this one with Vietnamese pre-trained word embeddings provided by Kyubyong Park obtains the best result. Other results are shown in the Table 2.

5   Experiment and Results

Table 2 shows our experiments on two models with and without different pre-trained word embedding – KP means the Kyubyong Park's pre-trained word embeddings and EG means Edouard Grave's pre-trained word embeddings.

We compare the outputs of BLSTM-CNN-CRF model (predicted) to the annotated data (gold) and analyzed the errors. Table 3 shows perfomance of the BLSTM-CNN-CRF model. In our experiments, we use three evaluation parameters (precision, recall, and F1 score) to access our experimental result. They will be described as follow in Table 3. The "correctNE", the number of correct label

---

[3] The pre-trained word vector of 30+ languages are available at https://github.com/Kyubyong/wordvectors

[4] The pre-trained word vector of 294 languages are available at https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.mdh

[5] https://research.fb.com/fasttext/

[6] https://code.google.com/archive/p/word2vec/

[7] wikipedia database backup dumps: https://dumps.wikimedia.org/backup-index.html

[8] https://www.wikipedia.org/

[9] The code of the BLSTM-CNN-CRF for NER systems are available at https://github.com/aonotas/deep-crf

Table 2. F1 score of two models with different pre-trained word embeddings

Model	F1 (%)
Bi-LSTM-CRF (no word embedings)	84.87
Bi-LSTM-CRF (KP word embedings)	86.69
Bi-LSTM-CRF (EG word embedings)	85.80
Bi-LSTM-CNN-CRF (no word embedings)	84.31
Bi-LSTM-CNN-CRF (KP word embedings)	86.87

for entity that the model can found. The "goldNE", number of the real label annotated by annotator in the gold data. The "foundNE", number of the label the model find out (no matter if they are correct or not).

$$Recall = \frac{correctNE \times 100}{goldNE} \tag{1}$$

$$Precision = \frac{correctNE \times 100}{foundNE} \tag{2}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{3}$$

Table 3. Performances of LSTM-CNN-CRF on the Vietnamese NER corpus

Tag name	Precision (%)	Recall (%)	F1(%)
All Result	87.70	85.71	86.70
LOC	87.63	86.87	87.25
MISC	97.44	77.55	86.36
PER	90.15	91.27	90.71
ORG	71.23	55.11	62.14

In Table 3 above, we can see that recall score on ORG label is lowest. The reason is almost all the ORG label on test file is name of some brands that do not appear on training data and pre-trained word embedding. On the other side, the characters inside these brand names also inside the other names of person in the training data. The context from both side of the sentence (future- and past-feature) also make the model "think" the name entity not as it should be.

Table 4 shows that the biggest number of errors is **No extraction**. The errors were counted by using logical sum (OR) of the gold labels and predicted labels (predicted by the model). The second most frequent error was **Wrong tag** means the model extract it's a NE but wrong tag.

5.1 Error analysis on gold data

First of all, we will compare the predicted NEs to the gold NEs (Fig. 1). Table 4 shows the summary of errors by types based on the gold labels, the "correct" is the number of gold tag that the model predicted correctly, "error" is the number of gold tag that the model predicted incorrectly, and "total" is sum of them. Four columns next show the number of type errors on each label.

Table 5 shows that Person, Location and Organization is the main reason why **No extraction** and **Wrong tag** are high.

After analyzing based on the gold NEs, we figure out the reason is:

 – Almost all the NEs is wrong, they do not appear on training data and pre-trained embedding. These NEs vector will be initial randomly, therefore, these vectors are poor which means have no semantic aspect.

 – The "weird" ORG NE in the sentence appear together with other words have context of PER, so this "weird" ORG NE is going to be label at PER.
    For example:
    gold data: VDV được xem là đầu_tiên ký hợp_đồng quảng_cáo là võ_sĩ ⟨PER⟩ Trần Quang Hạ ⟨PER⟩ sau khi đoạt HCV taekwondo Asiad ⟨LOC⟩ Hiroshima ⟨LOC⟩.
    (The athlete is considered the first to sign a contract of boxing Tran Quang Ha after winning the gold medal Asiad Hiroshima)
    predicted data: . . . là võ_sĩ ⟨PER⟩Trần Quang Hạ⟨PER⟩ sau khi đoạt HCV taekwondo Asiad ⟨PER⟩Hiroshima⟨PER⟩.

 – Some mistakes of the model are from training set, for example, anonymous person named "P." appears many times in the training set, so when model meets "P." in context of "P. 3 Quận 9" (Ward 3, District 9) – "P." stands for "Phường" (Ward) model will predict "P." as a PER.
    Training data: nếu ⟨PER⟩P.⟨PER⟩ có ở đây – (If P. were here) Predicted data: ⟨PER⟩P. 3⟨PER⟩, Gò_vấp – (Ward 3, Go_vap District)

Table 4. Summary of error results on gold data

Error type	Number (NE)	Rate (%)
No extraction	142	33.18
Wrong tag	112	26.17
Wrong range	100	23.36
Wrong range and tag	74	17.29
All errors	428	100

Table 5. Summary of detailed error results on gold data

Tags	Correct	Error	Total	No Extraction	Wrong Tag	Wrong Range	Wrong Range & Tag
Person	1181	113	1294	51	32	24	6
Location	1198	181	1377	54	39	59	29
Org	151	123	274	31	41	17	34
MISC	38	11	49	6	0	0	5
All Tags	2566	428	2994	142	112	100	74

5.2 Analysis on predicted data

Table 6 shows the summary of errors by types based on the predicted data. After analyzing the errors on predicted and gold data, we noticed that the difference of these errors are mainly in the **No anotation** and **No extraction**. Therefore, we only mention the main reasons for the **No anotation**:

Most of the wrong labels that model assigns are brand names (Ex: Charriol, Dream, Jupiter, ...), words are abbreviated (XKLD – xuất khẩu lao động (labour export)), movie names, ... All of these words do not appear in training data and word embedding. Perhaps these reasons are the followings:

- The vectors of these words are random so the semantic aspect is poor.

- The hidden states of these words also rely on past feature (forward pass) and future feature (backward pass) of the sentence. Therefore, they are assigned wrongly because of their context.

- These words are primarily capitalized or all capital letters, so they are assigned as a name entity. This error is caused by the CNN layer extract characters information of the word.

Table 6. Summary of error results on predicted data

Error type	Number (NE)	Rate (%)
Wrong tag	113	31.48
Wrong range	88	24.51
Wrong range and tag	69	19.22
No annotation	89	24.79
All errors	359	100

Table 7 shows the detail of errors on predicted data where we will see number kind of errors on each label.

Table 7. Summary of detailed error results on predicted data

Tags	Correct	Error	Total	No Annotation	Wrong Tag	Wrong Range	Wrong Range & Tag
Person	1181	129	1310	40	52	20	17
Location	1198	169	1367	26	54	53	36
Org	151	60	212	22	7	15	16
MISC	38	1	39	1	0	0	0
All Tags	2566	359	2928	89	113	88	69

5.3 Errors of annotators

After considering the training and test data, we realized that this data has many problems need to be fixed in the next run experiments. The annotators are not consistent between the training data and the test data, more details are shown as follow:

– The organizations are labeled in the train data but not labeled in the test data:
  Training data: ⟨ORG⟩ Sở Y_tế ⟨ORG⟩ (Department of Health)
  Test data: Sở Y_tế (Department of Health)
  Explanation: "Sở Y_tế" in train and test are the same name of organization entity. However the one in test data is not labeled.

– The entity has the same meaning but is assigned differently between the train data and the test:
  Training data: ⟨MISC⟩ người Việt ⟨MISC⟩ (Vietnamese people)
  Test data: dân ⟨LOC⟩ Việt ⟨LOC⟩ (Vietnamese people)
  Explanation: Both "người Việt" in train data and "dân Việt" in test data are the same meaning, but they are assigned differently.

– The range of entities are differently between the train data and the test data:
  Training data: ⟨LOC⟩ làng Atâu ⟨LOC⟩ (Atâu village)
  Test data: làng ⟨LOC⟩ Hàn_Quốc ⟨LOC⟩ (Korea village)
  Explanation: The two villages differ only in name, but they are labeled differently in range

– Capitalization rules are not unified with a token is considered an entity:
  Training data: ⟨ORG⟩ Công_ty Inmasco ⟨ORG⟩ (Inmasco Company)
  Training data: công_ty con (Subsidiaries)
  Test data: công_ty ⟨ORG⟩ Yeon Young Entertainment ⟨ORG⟩ (Yeon Young Entertainment company)
  Explanation: If it comes to a company with a specific name, it should be

labeled ⟨ORG⟩ Công_ty Yeon Young Entertainment ⟨ORG⟩ with "C" in capital letters.

6   Conclusion

In this paper, we have presented a thorough study of distinctive error distributions produced by Bi-LSTM-CNN-CRF for the Vietnamese language. This would be helpful for researchers to create better NER models.

Based on the analysis results, we suggest some possible directions for improvement of model and for the improvement of data-driven NER for the Vietnamese language in future:

1. The word at the begin of the sentence is capitalized, so, if the name of person is at this position, model will ignore them (no extraction). To improve this issue, we can use the POS feature together with BIO format (Inside, Outside, Beginning) [4] at the top layer (CRF).
2. If we can unify the labeling of the annotators between the train, dev and test sets. We will improve data quality and classifier.
3. It is better if there is a pre-trained word embeddings that overlays the data, and segmentation algorithm need to be more accurately.

References

1. Chiu, J.P., Nichols, E.: Named entity recognition with bidirectional lstm-cnns. Transactions of the Association for Computational Linguistics 4, 357–370 (2016)
2. Huang, Z., Xu, W., Yu, K.: Bidirectional lstm-crf models for sequence tagging. arXiv preprint arXiv:1508.01991 (2015)
3. Kuru, O., Can, O.A., Yuret, D.: Charner: Character-level named entity recognition. In: Proceedings of The 26th International Conference on Computational Linguistics. pp. 911–921 (2016)
4. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. arXiv preprint arXiv:1603.01360 (2016)
5. Le, T.H., Nguyen, T.T.T., Do, T.H., Nguyen, X.T.: Named entity recognition in vietnamese text. In: Proceedings of The Fourth International Workshop on Vietnamese Language and Speech Processing. Hanoi, Vietnam (2016)
6. Le-Hong, P.: 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 (2016)
7. Le-Hong, P., Nguyen, T.M.H., Roussanaly, A., Ho, T.V.: A hybrid approach to word segmentation of Vietnamese texts. In: Language and Automata Theory and Applications, Lecture Notes in Computer Science, vol. 5196, pp. 240–249. Springer Berlin Heidelberg (2008)
8. Ma, X., Hovy, E.: End-to-end sequence labeling via bi-directional lstm-cnns-crf. arXiv preprint arXiv:1603.01354 (2016)
9. Nguyen, T.C.V., Pham, T.S., Vuong, T.H., Nguyen, N.V., Tran, M.V.: Dsktlabner: Nested named entity recognition in vietnamese text. In: Proceedings of The Fourth International Workshop on Vietnamese Language and Speech Processing. Hanoi, Vietnam (2016)

10. Nguyen, T.S., Nguyen, L.M., Tran, X.C.: Vietnamese named entity recognition at vlsp 2016 evaluation campaign. In: Proceedings of The Fourth International Workshop on Vietnamese Language and Speech Processing. Hanoi, Vietnam (2016)

11. John Aberdeen, Samuel Bayer, Reyyan Yeniterzi, Ben Wellner, Cheryl Clark, David Hanauer, Bradley Malin, and Lynette Hirschman. 2010. The mitre identification scrubber toolkit: design, training, and assessment. International journal of medical informatics 79(12):849–859.

12. Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In Proceedings of the 43rd annual meeting on association for computational linguistics. Association for Computational Linguistics, pages 363– 370.

13. Robert Leaman, Graciela Gonzalez, et al. 2008. Banner: an executable survey of advances in biomedical named entity recognition. In Pacific symposium on biocomputing. volume 13, pages 652–663.

14. HC Cho, N Okazaki, M Miwa, and J Tsujii. 2010. Nersuite: a named entity recognition toolkit. Tsujii Laboratory, Department of Information Science, University of Tokyo, Tokyo, Japan.

15. Masaaki Ichihara, Kanako Komiya, Tomoya Iwakura, and Maiko Yamazaki. 2015. Ichihara2015ErrorAO: Error Analysis of Named Entity Recognition in BCCWJ

16. Thai-Hoang Pham, 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.

17. Bojanowski, Piotr and Grave, Edouard and Joulin, Armand and Mikolov, Tomas. 2016. Enriching Word Vectors with Subword Information.

18. Franck Dernoncourt, Ji Young Lee and Peter Szolovits. 2017. NeuroNER: an easy-to-use program for named-entity recognition based on neural networks.