Bidirectional LSTM-CRFs Model for Vietnamese Named-Entity Recognition Task

Minh-Hong Truong

Hanoi University of Science and Technology

Hanoi, Vietnam

hong.tm194576@sis.hust.edu.vn

I. BiLSTM – CRF MODEL

In this section, we will briefly describe the way we built our model.

A. Word embedding

In our experiment, we used the available word embedding set provided by PhoW2V [1], which was pre-trained on a 20GB corpus of Vietnamese texts to initialize word lookup tables. We used 100-dimension-word-level vectors on this task. For unknown words that don't appear in the set, we randomly initialize a vector called <UNK>. The UNK embedding is created by random vectors uniformly sampled from the range $\left[-\sqrt{\frac{3}{dim}};+\sqrt{\frac{3}{dim}}\right]$ where *dim* is the dimension of the word embedding vector.

B. Character-level embedding

Firstly, each character will be encoded to a 10 – dimension vector, then will be given to a Bidirectional LSTM to extract features. Two last hidden states from forward and backward layers of Bidirectional LSTM will be concatenated to become character-level word embedding vectors as described in Fig. 1.

C. Part of Speech (POS)

For POS tagging task, we use the available Python Vietnamese Toolkit provided by [2]. There are 18 POS tags in total, For each tag, we encode a one-hot vector whose length is equal to the number of POS tags, in this scenario is 18. For example:

- L (Determiner): 000001000000000000
- A (Adjective): 010000000000000000

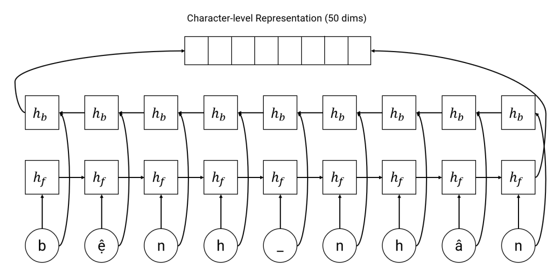

Fig. 1. The character embeddings of the word "bệnh_nhân" are given to a bidirectional LSTM to extract character-level word features.

D. Combined Bi-LSTM-CRFs Model

The concatenation of 100 – dim word embedding vector, 50 – dim character-level embedding vector, 18 – dim one-hot POS tagging vector will be concatenated and fed into Bidirectional LSTM. After that, the output vector of Bi-LSTM layer is passed through the CRF layer and decoded via the Viterbi algorithm (part of the CRF layer) instead of decoding each label independently, to select the most possible sequence of named-entity tags. The CRF layer can add constraints to the final predicted to ensure that they are vaid tags. For example, "B-NAME I-LOCATION" is impossible so it is invalid. The CRF loss function takes the transition score matrix between each tag as parameters. That is why we can prevent invalid sequence tags. We used free and available CRF layer's implementation in Pytorch in documentation for BiLSTM/CRF layer's code [3]. The architecture of the model is shown in Fig. 2 below.

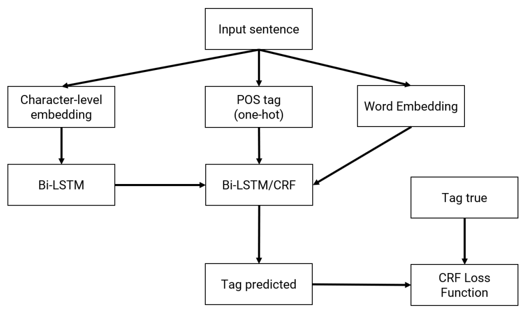

Fig. 2. The completed Bi-LSTM-CRF Model for the task of Vietnamese NER

II. EXPERIMENTS

A. Datasets

We will briefly describe the dataset that were used to evaluate the model performance on this task. We used COVID-19 Named Entity Recognition for Vietnamese dataset [4] which contains sentences with pre-tokenized vietnamese words and their respective NER tags. Training set, testing set and validation set has 5027, 3000, 2000 sentences respectively. There are 20 tags in total, the format of each tag in each category is shown in Table 1 below. The distribution of NER tags (excluding "O" tag) is shown in Fig. 3 below.

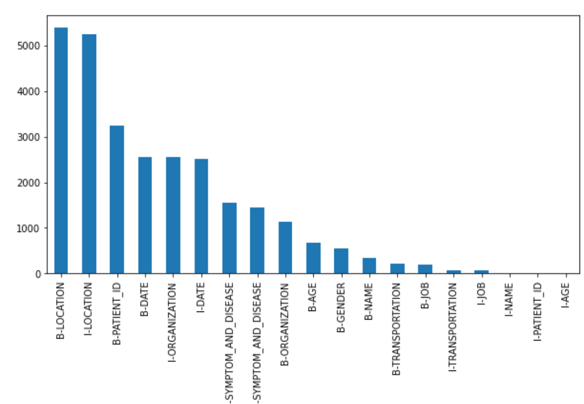

Fig. 3. The distribution of the amount of NER tags

Category	Format	Description	Example
Location	B-LOCATION I-LOCATION	Beginning of a location name Inside of a location name	ĐH Hà_Nội
Organization	B-ORGANIZATION I-ORGANIZATION	Beginning of an organization name Inside of an organization name	Sở Y_tế
Symptom	B-SYMPTOM_AND_DISEASE I-SYMPTOM_AND_DISEASE	Beginning of a symptom name Inside of a symptom name	viêm phổi
Name	B-NAME I-NAME	Beginning of any proper name Inside of any proper name	N.T.T ..
Transportation	B-TRANSPORTATION I-TRANSPORTATION	Beginning of a transportation name Inside of a transportation name	máy bay SQ
Age	B-AGE I-AGE	Beginning of age Inside of age	5 tháng
Patient ID	B-PATIENT_ID I-PATIENT_ID	Beginning of patient ID Inside of patient ID	BN 1069
Job	B-JOB I-JOB	Beginning of a job name Inside of a job name	nhân_viên ngân_hàng
Date	B-DATE I-DATE	Beginning of a date Inside of a date	24 –
Gender	B-GENDER	Beginning of Vietnamese gender words.	nam
Others	O	Words don't belong to any type of entity.	Trong

Table 1. Format of each tag in each category

B. Hyper-parameters

Table 2 shown below summarizes hyper-parameters that were chosen for all experiments. For your information, character extract features Bidirectional LSTM's hidden layer size is 25 so when we concatenate two layers (forward layer and backward layer), we will have a 50-dimension vector at the end of the model.

Hyper-parameter	Value
Word dimension	100
Hidden size word	200
Character dimension	10
Hidden size char	25
Update function	Adam
Learning rate	0.001
Batch size	32

Table 2. Hyper-parameters

C. Experimental Results

We used early-stopping based on performance on the development set with maximum 30 epochs on the training set. We measured the performance of the model by conlleval [5]. We experimented our model in three scenarios: only word embedding, without and with POS tag feature. Number of epochs in which our model had trained in three scenarios is 19, 27, 15 respectively. The tagging results on the testing dataset are shown in the Table 3 below.

Features	Accuracy	Precision	Recall	F1
Word	91.63%	86.14%	69.63%	77.19%
Word + Char	93,66%	84,68%	80,32%	82,44%
Word + Char + POS	**95,71%**	**91,31%**	**87,12%**	**89,17%**

Table 3. Experiments result

More detailed results can be found in source code.

III. COMPARISON

A. Word Embedding

We also tested our BiLSTM - CRF model with different pretrained Vietnamese pre-trained word embedding sets such as word-level embedding (300 dims), syllable-level embedding (100 dims and 300 dims) by [1]

Result of different pre-trained word embedding sets is shown in the Table 4 below.

Word Embedding	Accuracy	Precision	Recall	F1
[PhoW2V] Word-level 100d	95,71%	**91,31%**	87,12%	**89,17%**
[PhoW2V] Word-level 300d	95,75%	89,11%	86,85%	87,96%
[PhoW2V] Syllable-level 100d	95,25%	88,92%	86,67%	87,78%
[PhoW2V] Syllable-level 300d	**95,78%**	88,68%	**87,28%**	87,97%

Table 4. Result with different word embedding sets

More specific result of some important labels are shown in Table 5 and Table 6 below

Labels	Precision	Recall	F1
Location	89,32%	88,57%	88,94%
Organization	77,64%	70,43%	73,86%
Patient ID	97,61%	96,62%	97,11%

Table 5. Some labels's result with Word-level 100 - dimension word embedding set

Labels	Precision	Recall	F1
Location	85,9%	88,27%	87,07%
Organization	70,84%	73,9%	72,33%
Patient ID	95,12%	98,45%	96,76%

Table 6. Some labels's result with Syllable-level 300 - dimension word embedding set

B. Other models

Table 7 shows the F1 score of other models and our model on the dataset PhoNER_COVID19 [4]

Model	F1(%)
ViT5 (Long Phan, Hieu Tran et al., 2022)	94,5
PhoBERT (Thinh Hung Truong et al., 2021)	94,5
ViHealthBERT (Minh Phuc Nguyen, Vu Hoang Tran, Vu Hoang et al., 2022)	96,7
Our best model	89,17

Table 7. NER result for PhoNER_COVID dataset

IV. CONCLUSIONS

In this task, we used Bidirectional LSTM/CRF model that combines character-level and word-level embedding to approach named-entity recognition in Vietnamese. We tested our model in three scenarios mentioned above, as a result, our model provides the best performance with word embedding, character-level representation and POS tag feature. We also trained models with different pre-trained Vietnamese word embedding sets and compared the best result with other top performances in the same dataset.

Source code can be found ***here***.

IV. REFERENCES

[1] Anh Tuan Nguyen, Mai Hoang Dao, and Dat Quoc Nguyen. 2020. A Pilot Study of Text-to-SQL Semantic Parsing for Vietnamese. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 4079–4085, Online. Association for Computational Linguistics.

[2] https://github.com/trungtv/pyvi.git

[3] https://github.com/pytorch/tutorials/blob/5880829e63e7e6b4db974404c24a798d12bc19ec/beginner_source/nlp/advanced_tutorial.py

[4] Thinh Hung Truong, Mai Hoang Dao, and Dat Quoc Nguyen .2021. COVID-19 Named Entity Recognition for Vietnamese. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 2146-2153, Online. Association for Computational Linguistics.

[5] https://github.com/sighsmile/conlleval.git