Singapore Management University

Institutional Knowledge at Singapore Management University

Research Collection School Of Computing and Information Systems | School of Computing and Information Systems

1-2020

Vietnamese punctuation prediction using deep neural networks

Thuy PHAM

Nhu NGUYEN

Hong Quang PHAM
Singapore Management University, hqpham.2017@phdis.smu.edu.sg

Han CAO

Binh NGUYEN

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research

 Part of the Numerical Analysis and Scientific Computing Commons, and the South and Southeast Asian Languages and Societies Commons

Citation

PHAM, Thuy; NGUYEN, Nhu; PHAM, Hong Quang; CAO, Han; and NGUYEN, Binh. Vietnamese punctuation prediction using deep neural networks. (2020). *SOFSEM 2020: Theory and Practice of Computer Science: Limassol: January 20-24: Proceedings*. 12011, 388-400.
Available at: https://ink.library.smu.edu.sg/sis_research/7817

This Conference Proceeding Article is brought to you for free and open access by the School of Computing and Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Computing and Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email cherylds@smu.edu.sg.

Vietnamese Punctuation Prediction Using Deep Neural Networks

Thuy Pham[1], Nhu Nguyen[1], Quang Pham[2], Han Cao[3], and Binh Nguyen[1,3,4(✉)]

[1] University of Science, Vietnam National University in Ho Chi Minh City, Ho Chi Minh City, Vietnam
ngtbinh@hcmus.edu.vn
[2] Singapore Management University, Singapore, Singapore
[3] Inspectorio Research Lab, Ho Chi Minh City, Vietnam
[4] AISIA Research Lab, Ho Chi Minh City, Vietnam

Abstract. Adding appropriate punctuation marks into text is an essential step in speech-to-text where such information is usually not available. While this has been extensively studied for English, there is no large-scale dataset and comprehensive study in the punctuation prediction problem for the Vietnamese language. In this paper, we collect two massive datasets and conduct a benchmark with both traditional methods and deep neural networks. We aim to publish both our data and all implementation codes to facilitate further research, not only in Vietnamese punctuation prediction but also in other related fields. Our project, including datasets and implementation details, is publicly available at https://github.com/BinhMisfit/vietnamese-punctuation-prediction.

Keywords: Punctuation prediction · BiLSTM · Conditional random field · Attention model

1 Introduction

Punctuation is a system of symbols indicating the structure of a sentence where one needs to slow down, notice, or express emotion. Punctuation marks are vital to understand and disambiguate the meaning of sentences. Most automatic speech recognition systems usually do not provide punctuation in their outputs. Therefore, it is essential to assign appropriate punctuation marks to transcribed text so that it can be understood correctly.

In literature, punctuation prediction has been extensively studied during the last two decades, especially in the English language. Beerferman et al. [3] propose a lightweight approach for constructing a punctuation annotation system by relying on a trigram language model and Viterbi algorithm. Huang and Zweig [6] model the punctuation annotation problem as a sequence tagging problem where each word is tagged with appropriate punctuation. Lu et al. [12] present a new punctuation prediction approach for transcribed conversational speech texts using the dynamic conditional random field model on both Chinese and English.

Cuong et al. [14] propose efficient inference algorithms to capture long-range dependencies among punctuations using high-order semi-Markov conditional random fields. Peitz [16] formulate the punctuation prediction as machine translation instead of using a language model based punctuation prediction method. Zhang et al. [21] study a new technique in punctuation prediction for the stream of words in transcribed speech texts with excellent accuracy in both test datasets of IWSLT [15] and TDT4 [19]. Regarding neural network methods, Tilk et al. [20] introduce a two-stage recurrent neural network using LSTM units to predict suitable punctuation for automatic speech recognition systems. Ballesteros and Wanner [2] investigate a novel LSTM-based model for predicting punctuation marks into raw text material. Recently, Li et al. [9] introduce an efficient generative model for punctuation prediction without observing the underlying punctuation marks and reconstructing the tree's underlying punctuation. Regarding the Vietnamese language, there have been various works in different fields such as word segmentation [4,13] and Part-of-Speech (POS) tagging [17].

In this work, we aim at building a large-scale dataset and providing an extensive benchmark for predicting punctuation in the Vietnamese language. Notably, we collect over 40,000 articles from the Vietnamese news and novels to build two datasets with a total of over 900,000 sentences. Different from previous works of [14,18], which assume inputs are already segmented into sentences; we make a general assumption that inputs can contain several sentences without punctuation information. Therefore, we train our model on paragraphs, which is more realistic and challenging. To provide a comprehensive benchmark for this task, we consider both traditional methods using CRF [18] and deep neural networks. Generally, the punctuation distribution in the text is highly imbalanced: most words are followed by a space that makes training punctuation prediction systems even more difficult. To address this challenge, we propose to train deep neural networks with the *focal loss* [10], which can give more weights to rare classes. While the focal loss shows promising results with our experiments on the Vietnamese Novels dataset, the class imbalance nature of this task is still a challenging problem and becomes an important research direction. We strongly believe that different languages have divergent challenges to build an efficient punctuation prediction system. As a consequence, our work can be considered as an additional contribution to the problem for the Vietnamese language, where there is little publication using a deep learning approach.

Since training with paragraph requires a strong text representation and the model's ability to remember long-range dependencies, we argue that the traditional CRF based methods are not suitable for this setting. Mainly, each CRF model treats each word as a one-hot vector, thus does not exploit its rich semantic meaning. Moreover, CRFs, primarily linear CRFs, only consider the relationship among words in a small window, thus ignoring information from distant words, which can be potentially informative. To address the above limitations, we propose a deep LSTM with an attention model to predict punctuations from the text. Our model learns to represent words by embedding vectors to exploit their semantic relationship. LSTM [5] can model long term relationships in sequences,

which is used as the base of our model to accumulate knowledge in the paragraph. However, LSTM may remember information from a too far distant, which may be noisy and hinder the overall performance. Therefore, we equip LSTM with an attention layer so that it can selectively choose which information in the past is useful for the current prediction.

2 Punctuation Prediction as Sequence Tagging

2.1 Problem Formulation

Similar to the previous works for English and Chinese [11,22], we model the punctuation prediction task as a sequence labeling problem. Remarkably, we label each word by its immediately following punctuation, where label O denotes a space. In this study, we aim at considering seven main types of punctuation marks in the Vietnamese language including the period (.), the comma (,), the colon (:), the semicolon (;), the question mark (?), the exclamation mark (!), and the space. By modeling punctuation prediction as a sequence tagging problem, conventional methods such as conditional random fields (CRF) and neural networks can be applied directly without any significant modification. In the simple case, we use the label O to indicate that a word is not followed by any punctuation. For example, one can consider the following sentence in the Vietnamese language[1].

> Biển tạo ra 1/2 lượng oxy con người hít thở, giúp lưu chuyển nhiệt quanh Trái Đất và hấp thụ một lượng lớn CO2.
>
> (The ocean produces a half of the amount of oxygen that humans can breathe, and help to circulate heat around the Earth and absorb large amounts of CO2.)

This paragraph can be labeled as follows.

biển/O tạo/O ra/O 1/2/O lượng/O oxy/O con/O người/O hít/O thở/Comma giúp/O lưu/O chuyển/O nhiệt/O quanh/O trái đất/O và/O hấp/O thụ/O một/O lượng/O lớn/O co2/Period

It is worth noting that all the words are in lower case since the word case information is usually not available for the punctuation prediction task. For instance, when the texts are transcribed from speeches, we do not have the case information for the words.

2.2 Punctuation Prediction with Conditional Random Field

By formulating the punctuation prediction as a sequence labeling problem, a simplified approach is employing Conditional Random Field (CRF) [8], which has been applied successfully in the literature [14,18]. As our work is closely related to [18], we consider CRF as a baseline and implement CRF with three feature templates, as suggested in [18].

[1] https://vnexpress.net/khoa-hoc/dai-duong-can-thiet-voi-su-song-tren-trai-dat-the-nao-3976195.html.

3 Neural Networks for Punctuation Prediction

3.1 Network Architectures

Semantic Representation of Syllables. In this section, we describe our proposed approach to obtain the semantic vector of each syllable in a sequence. First, we initialize two embedding matrices for syllable and character as $E_s \in \mathbb{R}^{d \times S}$ and $E_c \in \mathbb{R}^{d \times C}$, where S and C are respectively the numbers of syllables and characters in the vocabulary, and d is the embedding dimension. For simplicity, here we use the same embedding dimension d for both syllables and characters. Given a sequence of L syllables $x = \{x_1, \ldots, x_L\}$, each of which is represented as a one-hot vector, we calculate the sequence of syllable embedding as:

$$\begin{aligned} e^s_x &= \{e^s_{x_1}, \ldots, e^s_{x_L}\} \text{ satisfying that} \\ e^s_{x_i} &= E_s \cdot x_i, \end{aligned} \tag{1}$$

where $(\cdot)$ is the matrix-vector dot product and $e^s_{x_i} \in \mathbb{R}^d$. Each element of $e^s_{x_i}$ is a semantic representation of the syllable x_i. However, a common practice is that we usually map rare words into the same vector corresponding to an "out of vocabulary (OOV)" token, which may lose useful information and hinder the performance. Therefore, we propose to enhance the semantic vectors $e^s_{x_i}$ with the semantic information from the character constructing x_i. Without loss of generality, we assume that each syllable x_i is itself a sequence of N characters $x_i = \{c_1, \ldots, c_N\}$. Similarly, we can obtain the sequential character representation of x_i as $se^c_{x_i} \in \mathbb{R}^{d \times N}$

$$se^c_{x_i} = \{e^c_{c_1}, \ldots, e^c_{c_N}\} \text{ satisfying that} \tag{2}$$

$$e^c_{c_i} = E_c \cdot c_i \tag{3}$$

Since characters in a syllable have short-range dependencies, we can learn such dependencies in $e^c_{x_i}$ by applying a convolution layer defined as

$$c_j = f(W \otimes e^c_{c_j : c_{j+h-1}}), \tag{4}$$

where $W \in \mathbb{R}^{d \times h}$ is the convolution parameter with length h and $\otimes$ denotes the convolution operation. By applying the operations defined in Eq. (3) on $e^c_{x_i}$, we get the character dependences in $e^c_{x_i}$ as $c = [c_1, \ldots, c_{N-h+1}]$, where each c_i represents the relationship among h consecutive characters in x_i. To obtain a fixed representation of the semantic vector built from characters, we apply the max pooling over c to compute $e^c_{x_i} \in \mathbb{R}^d$, and then, we combine it with $e^s_{x_i}$ for achieving a syllable representation as follows:

$$e_{x_i} = e^s_{x_i} \oplus e^c_{x_i}, \tag{5}$$

where $\oplus$ is the vector element-wise summation.

Predicting Punctuation with Deep Neural Networks. To this end, we have the semantic representation $e = \{e_{x_1}, \ldots, e_{x_L}\}$ of the original sequence x. In the next step, we use a Bidirectional LSTM to read the sequence e from both ends and obtain a sequence of hidden states, each of which is a concatenation of each individual LSTM's hidden state: $h_i = [\overrightarrow{h_i}, \overleftarrow{h_i}], i = 1, \ldots, L$ and $h_i \in \mathbb{R}^{2h}$, where h is the hidden size of one component LSTM. Subsequently, the model predicts the distribution of punctuations over the syllable x_i as

$$\hat{y}_i = \text{softmax}(R \cdot h_i), \tag{6}$$

where $R \in \mathbb{R}^{|Y| \times 2h}$ is the parameter of the softmax layer and $|Y|$ denotes the total number of punctuations in the vocabulary. Given the true punctuation prediction y, we can compute a loss (e.g. cross-entropy) between y and $\hat{y}$ and backprop to update all the parameters: $E_s, E_c, W, LSTM$, and R end-to-end.

Also, we consider two improved models that can potentially capture more complex structures of the data. First, we enhance the fully connected layer with the attention mechanism [1]. It means the model can focus on particular syllables in the past while predicting the current punctuation mark, and we refer to this as the **BiLSTM + Attention** model.

Finally, we replace the softmax classification layer by a CRF layer, which is a traditional method for this task [8]; this model is denoted as **BiLSTM + CRF**. It is important to remark that BiLSTM + Attention can be regarded as an additional improvement over BiLSTM due to the attention mechanism. Similarly, we also consider BiLSTM + CRF as an improvement over CRF for the reason that CRF models can use learned features from BiLSTM instead of manually designed features, as mentioned in [18].

3.2 Training with Focal Loss

A standard training procedure is to randomly sample a mini-batch from the training data, train the model, and then repeat until the convergence happens. As long as we model the punctuation prediction as a tagging problem, a nature choice of the loss function is the cross entropy loss between the predicted punctuation and the true punctuation. However, one main drawback of the classical cross entropy loss is that it has the same penalty for both easy and difficult classes, which is problematic as a result of the distribution of punctuation marks in natural languages is highly imbalanced. To address this problem, we propose to use the *focal loss* [10] that can give more weights to rare classes in the data:

$$FL\left(p_t\right) = -\alpha_t (1 - p_t)^{\gamma} \log\left(p_t\right). \tag{7}$$

Equation (7) shows the formula of the focal loss, where α_t is the balance factor of class t and γ is the *focusing factor*. Focal loss has been successfully used in the object detection problem where the training dataset is highly imbalanced with the background class. Nonetheless, focal loss has not been applied in natural language processing to the best of our knowledge (Fig. 1).

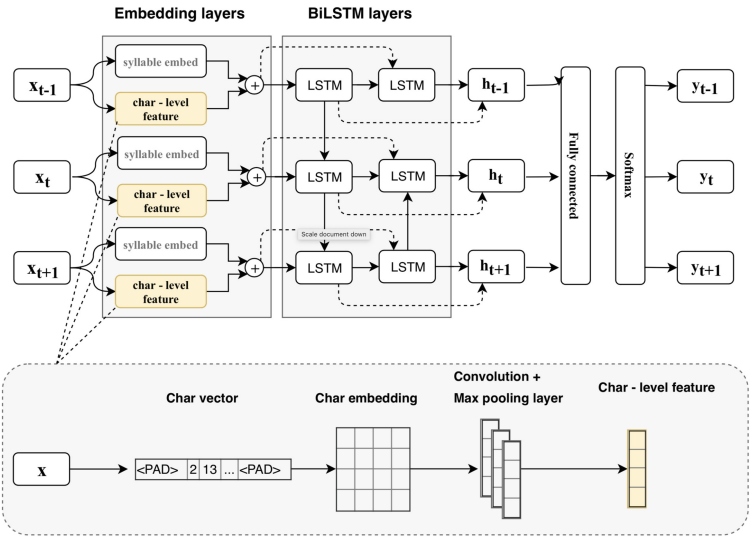


Fig. 1. A neural network architecture for the punctuation prediction problem.

4 Datasets for Vietnamese Punctuation Prediction

To investigate punctuation prediction for the Vietnamese language, we build two large-scale datasets from Vietnamese novels[2] and newspapers[3] with a total of over 900,000 sentences. Table 1 shows the different distribution of punctuation marks in these two datasets.

There are 734244 sentences in the Vietnamese Newspapers dataset, while the Vietnamese Novels dataset only has 183734. Although the top two punctuation marks having the most significant percentage of occurrence are comma and period in both datasets, the distribution of remaining ones is quite different. For instance, the appearance rate of the colon mark is 0.26% in Vietnamese newspapers, nearly three times bigger than the corresponding rate (0.092%) in Vietnamese novels. From Table 1, there exist much more (about 32 times) exclamative sentences in novels (1.894%) than newspapers (0.059%).

Similarly, we observe that authors prefer using interrogative sentences in Vietnamese novels (0.994%) rather than Vietnamese newspapers (0.113%). However, the occurrence rates of both colon and semicolon marks in newspapers are much larger than novels. These rates for both colon and semicolon marks in newspapers are 0.260% and 0.047%, respectively. Meanwhile, the corresponding values are 0.092% and 0.004% in novels. It turns out that Vietnamese novelists rarely use semicolon mark in their work. As a result, we decide not to merge two datasets owing to their inherently different sources, thus having different punctuation distributions. Therefore, it is worth seeing how proposed models perform on entirely different datasets.

[2] https://gacsach.com/tac-gia/nguyen-nhat-anh.html.
[3] https://baomoi.com.

Table 1. The distribution of punctuation marks in the training, testing sets from Vietnamese Novels and News dataset.

	Novel dataset				News dataset			
Punctuation	Training set		Test set		Training set		Test set	
	Number	%	Number	%	Number	%	Number	%
Comma (,)	50909	3.77	21231	4.045	482435	4.041	160472	4.054
Period (·)	66519	4.926	29643	5.648	419580	3.514	138967	3.51
Colon (:)	742	0.055	1153	0.221	32177	0.269	10728	0.271
Qmark (?)	14899	1.103	5271	1.004	13902	0.116	4468	0.113
Exclam (!)	30183	2.235	9167	1.747	7384	0.062	2333	0.059
Semicolon (;)	48	0.004	43	0.008	5675	0.048	2045	0.052
Sentences	**111601**		**44081**		**440866**		**145768**	

To pre-process the data, we first remove special characters, convert all words into lower cases, and standardize URLs, emails, and hashtags. Then, we remove sentences that do not contain any punctuation mark, do not end with a punctuation mark, or the ending punctuation is not a period, a question mark, or an exclamation mark. Different from previous works [14,18] assuming data are already segmented into sentences, here we do not make such assumptions and allow each model to work on arbitrary paragraphs of the text. Therefore, as most of the lengths of sentences on our datasets are smaller than or equal to 100, we decide to split the data into segments of length 100 and label them using the format as described in Sect. 2.1 and [14,18].

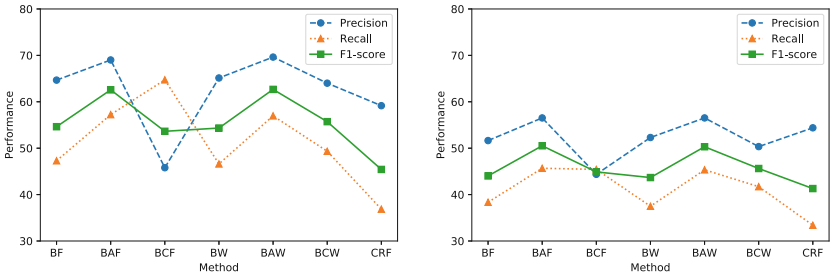
Finally, we divide the data into training, validation and testing sets with the ratio 60%–20%–20%. The distribution of punctuation marks among the training and testing sets for two datasets (Vietnamese Novels and Vietnamese Newspapers) can be found in Table 1.

5 Experiments

In this section, we present our experiments on two datasets described in Sect. 4. We consider both traditional CRF models as described in Sect. 2, and deep learning models (**BiLSTM, BiLSTM+Attention**, and **BiLSTM+CRF**) trained with both focal loss and normal cross-entropy loss. For deep learning models, we initialize the character embedding randomly and use Fasttext[4] as an initialization for the syllable embedding and syllables that are not in Fasttext have their embeddings initialized randomly. Both character and syllable embedding matrices are updated during the training process. All hyper-parameters such as the learning rate and focal loss hyperparameters are cross-validated from the validation set. We use the CRF++ toolkit[5] and implement other models with

[4] https://fasttext.cc/.
[5] https://taku910.github.io/crfpp/.

Tensorflow[6]. For deep learning models, we set the LSTM's hidden dimension to be 300 and train using Adam optimizer [7] for 30 epochs.



(a) Vietnamese Newspapers (b) Vietnamese Novels

Fig. 2. The performance comparison by micro precision, recall and F_1 score on the testing set (B: BiLSTM, A: Attention, C: CRF, F: trained with focal loss, W: trained without focal loss).

Table 2. Experimental results on the Vietnamese Newspapers dataset with focal loss (B: BiLSTM, A: Attention, C: CRF, F: trained with focal loss, W: trained without focal loss).

Punctuation	BF			BAF			BCF		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Comma (,)	62.99	41.33	49.91	66.96	53.46	59.45	42.57	62.65	50.69
Period (·)	66.90	60.32	63.44	72.51	67.20	69.76	50.12	74.54	59.94
Colon (:)	59.49	21.71	31.81	58.59	32.00	41.39	54.25	24.29	33.56
Qmark (?)	58.86	33.68	42.85	61.12	49.40	54.64	47.75	42.82	45.15
Exclam (!)	34.51	4.20	7.49	43.03	5.96	10.47	34.88	4.20	7.50
Semicolon (;)	25.58	2.52	4.58	32.48	4.35	7.67	24.85	1.60	3.01
MICRO AVERAGE	64.67	47.29	54.63	**69.01**	**57.23**	**62.57**	45.79	64.73	53.64

Tables 2, 3, 6, and 7 show the performance of different deep learning based methods in terms of Precision (P), Recall (R), and F1-score (F) using cross-entropy loss or focal loss in different methods for datasets. Here, **B** stands for BiLSTM, **W** stands for the case not using focal loss, **A** stands for the Attention model, and **C** stands for the CRF model. Finally, the performance of CRF models are reported in Tables 4 to 5. Due to space constraints, we refer to [18] for details for the three templates.

As the data are highly imbalanced among punctuation marks, we opt to use micro averaged precision (P), recall (R), and F_1 score [18] to evaluate these

[6] https://www.tensorflow.org/.

Table 3. Experimental results on the Vietnamese Newspapers dataset without using focal loss (B: BiLSTM, A: Attention, C: CRF, F: trained with focal loss, W: trained without focal loss).

Punctuation	BW			BAW			BCW		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Comma (,)	62.30	41.03	49.47	68.30	52.42	59.32	62.90	42.10	50.44
Period (·)	68.80	58.84	63.43	72.09	68.13	70.06	65.69	63.89	64.77
Colon (:)	58.10	23.12	33.07	61.54	29.87	40.22	56.62	26.35	35.96
Qmark (?)	63.10	33.48	43.75	61.01	51.30	55.73	57.66	39.32	46.76
Exclam (!)	37.27	5.96	10.27	35.71	7.50	12.40	44.71	5.62	9.98
Semicolon (;)	26.51	3.01	5.41	29.25	4.92	8.43	32.07	2.90	5.32
MICRO AVERAGE	65.13	46.61	54.34	**69.63**	**56.97**	**62.67**	64.01	49.34	55.72

Table 4. Experimental results on the Vietnamese Newspapers dataset using CRF models.

Punctuation	Template 1			Template 2			Template 3		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Comma (,)	50.22	14.03	21.93	58.07	34.77	43.50	58.50	33.13	42.31
Period (·)	60.46	24.86	35.23	60.95	43.54	50.80	62.22	42.40	50.43
Colon (:)	47.02	8.68	14.65	53.01	17.00	25.75	52.86	16.31	24.93
Qmark (?)	46.91	11.37	18.30	55.43	19.32	28.65	54.92	19.47	28.75
Exclam (!)	29.84	3.90	6.90	32.58	4.93	8.56	38.49	5.23	9.21
Semicolon (;)	20.00	0.76	1.47	26.80	1.56	2.96	27.97	1.26	2.41
MICRO AVERAGE	55.05	17.80	26.90	**59.16**	**36.86**	**45.42**	59.96	35.49	44.59

models. Models' hyper-parameters are cross-validated on the validation set and we report the best setting on the test set. Figure 2 shows the results of various models we considered. First, we observe that deep learning methods outperform the traditional CRF model significantly on both datasets. Moreover, BiLSTM+Attention achieves the highest performance overall. Second, on the Vietnamese Novels dataset, we observe that, except BiLSTM+CRF, models trained with focal loss have a modest improvement over the traditional cross-entropy loss. However, on the Vietnamese Newspapers dataset, training with focal loss results in nearly identical performance. One possible reason is that it is much more difficult to perform hyper-parameter selection on the Vietnamese Newspapers dataset, which results in the non-optimal setting for focal loss. Overall, experimental results show that class imbalance is a challenging problem in punctuation prediction, and focal loss can become a promising strategy to alleviate this difficulty.

Table 5. Experimental results on the Vietnamese Novels dataset using CRFs.

Punctuation	Template 1			Template 2			Template 3		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Comma (,)	42.47	15.92	23.16	51.69	25.92	34.53	52.66	35.26	42.23
Period (·)	44.94	21.25	28.86	52.77	34.72	41.88	51.78	26.22	34.81
Colon (:)	21.43	0.24	0.47	30.77	0.32	0.62	27.27	0.24	0.47
Qmark (?)	58.34	34.19	43.11	71.20	49.48	58.38	72.00	49.55	58.70
Exclam (!)	44.90	27.23	33.90	54.90	41.99	47.58	54.79	41.79	47.42
Semicolon (;)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
MICRO AVERAGE	45.52	20.77	28.52	54.40	33.09	41.15	**54.39**	**33.40**	**41.38**

Table 6. Experimental results on the Vietnamese Novels dataset with focal loss (B: BiLSTM, A: Attention, C: CRF, F: trained with focal loss, W: trained without focal loss).

Punctuation	BF			BAF			BCF		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Comma (,)	49.00	29.74	37.01	56.10	38.45	45.63	36.71	47.26	41.32
Period (·)	50.20	41.74	45.58	55.86	47.33	51.24	46.56	45.73	46.14
Colon (:)	50.00	0.24	0.47	21.43	0.95	1.81	0.00	0.00	0.00
Qmark (?)	69.56	55.45	61.71	70.34	65.60	67.89	60.90	67.41	63.99
Exclam (!)	51.48	45.79	48.47	52.09	54.30	53.18	59.47	33.52	42.88
Semicolon (;)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
MICRO AVERAGE	51.64	38.35	44.02	**56.52**	**45.67**	**50.52**	44.37	45.43	44.89

Table 7. Experimental results on the Vietnamese Novels dataset without using focal loss (B: BiLSTM, A: Attention, C: CRF, F: trained with focal loss, W: trained without focal loss).

Punctuation	BW			BAW			BCW		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Comma (,)	52.04	27.05	35.60	56.13	38.35	45.57	48.53	32.46	38.90
Period (·)	49.69	41.72	45.36	55.51	47.01	50.91	49.22	44.45	46.71
Colon (:)	26.67	0.32	0.62	66.67	0.63	1.25	14.71	0.39	0.77
Qmark (?)	69.05	56.69	62.26	71.67	64.69	68.00	68.39	61.09	64.54
Exclam (!)	51.81	45.63	48.53	52.20	53.66	52.92	47.86	52.03	49.86
Semicolon (;)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
MICRO AVERAGE	52.30	37.49	43.67	**56.52**	**45.34**	**50.31**	50.34	41.71	45.62

Detailedly, for the best model using BiLSTM + Attention, from the Vietnamese Newspapers dataset, using the focal loss could achieve a slightly lower (about 0.1%) F1-score than without using it (62.57% vs. 62.67%). Meanwhile, for the Vietnamese Novels dataset, using the focal loss could obtain a slightly higher F1-score than without using the focal loss (50.52% vs. 50.31%). The experimental results in both datasets are a little bit different due to the difference between the distribution of punctuation marks in these two datasets, and especially, some punctuation marks rarely occur in Vietnamese novels rather than Vietnamese newspapers. In addition, we perform grid search on the pairs (α, γ) with α in $\{0.1, 0.25, 0.5, 0.75, 0.99\}$ and γ in $\{0.1, 0.5, 1.0, 2.0, 5.0\}$ for focal loss hyper-parameter selection. In future work, we plan to increase the grid size and tune these parameters carefully to achieve better performance. For BiLSTM-CRF, the focal loss is originally developed for *softmax* classifiers on the top of deep networks. It may be one reason explaining the performance drop observed in BiLSTM-CRF models.

Finally, regardless of the training loss, our results (Fig. 2) show that BiLSTM with Attention is the best among all the models considered. Furthermore, training with the focal loss can provide modest improvement to BiLSTM and BiLSTM with Attention.

6 Conclusion and Future Work

We have studied the punctuation prediction problem for the Vietnamese language. We collect two large-scale datasets and conduct extensive experiments with both traditional method (using CRF models) and a deep learning approach. We address the class imbalance problem in this task and show promising results using the focal loss on the Vietnamese Newspapers data.

In future work, we plan to use word embeddings and other techniques (ELMO, BERT, or word segmentation) for data pre-processing. Also, we do different experiments with more challenging datasets using Vietnamese speech/spoken- conversation transcripts. For instance, datasets from the IWSLT evaluation campaigns can be used to construct an efficient method for Vietnamese punctuation prediction. Another research direction is combining the punctuation prediction problem with other classical NLP tasks such as word segmentation or named entity recognition. For example, if one could correctly tokenize a paragraph into words and label each token with a named entity, the disambiguation level of this paragraph would reduce. It turns out that the punctuation prediction system would be easier to train. However, existing tokenizer and NER systems trained with punctuation information available is not the case in our problem. Therefore, directly applying a tokenizer might be a suboptimal solution. We strongly believe that learning these two tasks together will offer a better solution. Eventually, both the data and the implementation are publicly available at https://github.com/BinhMisfit/vietnamese-punctuation-prediction for further research.

Acknowledgement. We would like to thank The National Foundation for Science and Technology Development (NAFOSTED), University of Science, Inspectorio Research Lab, and AISIA Research Lab for supporting us throughout this paper.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. CoRR abs/1409.0473 (2015)
2. Ballesteros, M., Wanner, L.: A neural network architecture for multilingual punctuation generation. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 1048–1053. Association for Computational Linguistics, Austin, November 2016. https://doi.org/10.18653/v1/D16-1111. https://www.aclweb.org/anthology/D16-1111
3. Beeferman, D., Berger, A., Lafferty, J.: Cyberpunc: a lightweight punctuation annotation system for speech. In: Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 2, pp. 689–692, May 1998. https://doi.org/10.1109/ICASSP.1998.675358
4. Dien, D., Hoang, K., Toan, N.V.: Vietnamese word segmentation. In: NLPRS (2001)
5. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
6. Huang, J., Zweig, G.: Maximum entropy model for punctuation annotation from speech, January 2002
7. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
8. Lafferty, J., McCallum, A., Pereira, F.C.: Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of the 18th International Conference on Machine Learning 2001 (ICML 2001), pp. 282–289 (2001)
9. Li, X.L., Wang, D., Eisner, J.: A generative model for punctuation in dependency trees, pp. 357–373, July 2019
10. Lin, T.Y., Goyal, P., Girshick, R.B., He, K., Dollár, P.: Focal loss for dense object detection. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2999–3007 (2017)
11. Lu, W., Ng, H.T.: Better punctuation prediction with dynamic conditional random fields. In: Conference on Empirical Methods in Natural Language Processing (2010)
12. Lu, W., Tou Ng, H.: Better punctuation prediction with dynamic conditional random fields, pp. 177–186, January 2010
13. Nguyen, C.T., Nguyen, T.K., Phan, X.H., Nguyen, L.M., Ha, Q.T.: Vietnamese word segmentation with CRFs and SVMs: an investigation. In: PACLIC (2006)
14. Nguyen, V.C., Ye, N., Lee, W.S., Chieu, H.L.: Conditional random field with high-order dependencies for sequence labeling and segmentation. J. Mach. Learn. Res. **15**, 981–1009 (2014)
15. Paul, M.: Overview of the IWSLT 2009 evaluation campaign. In: International Workshop on Spoken Language Translation (IWSLT) 2009, pp. 1–18 (2009)
16. Peitz, S., Freitag, M., Mauser, A., Ney, H.: Modeling punctuation prediction as machine translation. In: IWSLT (2011)
17. Pham, D.D., Tran, G.B., Pham, S.B.: A hybrid approach to Vietnamese word segmentation using part of speech tags. In: 2009 International Conference on Knowledge and Systems Engineering, pp. 154–161 (2009)

18. Pham, Q.H., Nguyen, B.T., Cuong, N.V.: Punctuation prediction for Vietnamese texts using conditional random fields. In: ACML Workshop: Machine Learning and Its Applications in Vietnam, pp. 1–9 (2014)
19. Stephanie, S., Kong, J., Graff, D.: TDT4 multilingual text and annotations LDC2005T16 (2005)
20. Tilk, O., Alumae, T.: LSTM for punctuation restoration in speech transcripts. In: INTERSPEECH 2015, pp. 683–687 (2015)
21. Zhang, D., Wu, S., Yang, N., Li, M.: Punctuation prediction with transition-based parsing. In: ACL (2013)
22. Zhao, Y., Wang, C., Fu, G.: A CRF sequence labeling approach to Chinese punctuation prediction. In: Pacific Asia Conference on Language, Information and Computation (2012)