Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 3913–3921
Marseille, 11–16 May 2020
© European Language Resources Association (ELRA), licensed under CC-BY-NC

Introducing a Large-Scale Dataset for Vietnamese POS Tagging on Conversational Texts

Oanh Thi Tran[*,†], **Tu Minh Pham**[*], **Vu Hoang Dang**[*], **Bang Ba Xuan Nguyen**[*]
[*]FPT Technology Research Institute - FPT University
82 Duy Tan, Cau Giay, Hanoi, Vietnam
(oanhtt12, tupm2, vudh5, bangnbx)@fpt.com.vn
[†]International School, Vietnam National University, Hanoi
144 Xuan Thuy, Cau Giay, Hanoi, Vietnam
oanhtt@isvnu.vn

Abstract

This paper introduces a large-scale human-labeled dataset for the Vietnamese POS tagging task on conversational texts. To this end, we propose a new tagging scheme (with 36 POS tags) consisting of exclusive tags for special phenomena of conversational words, develop the annotation guideline and manually annotate 16.310K sentences using this guideline. Based on this corpus, a series of state-of-the-art tagging methods has been conducted to estimate their performances. Experimental results showed that the Conditional Random Fields model using both automatically learnt features from deep neural networks and handcrafted features yielded the best performance. This model achieved 93.36% in the accuracy score which is 1.6% and 2.7% higher than the model using either handcrafted features or automatically-learnt features, respectively. This result is also a little bit higher than the model of fine-tuning BERT by 0.94% in the accuracy score. The performance measured on each POS tag is also very high with >90% in the F1 score for 20 POS tags and >80% in the F1 score for 11 POS tags. This work provides the public dataset and preliminary results for follow-up research on this interesting direction.

Keywords: Vietnamese POS tagging, conversational texts, CRF, neural networks

1. Introduction

POS tagging is one of the most critical tasks in Natural Language Processing (NLP) and has drawn much attention among its research community. Its performance usually greatly affect many NLP downstream tasks such as question answering, machine translation, named entity recognition, sentiment analysis, etc. So far, most POS tagging work has been dedicated to formal (or standardized) texts such as e-newspapers, official documents, and stories.

Along with the development of social media, the amount and types of data available from Twitter, Facebook, blogs, and chat-chit platforms are increasing exponentially. Unfortunately, most current POS taggers typically perform poorly on these kinds of social data[1]. The main reason is that these POS taggers were trained on standardized texts which are from a quite different domain (Gimpel et al., 2011). Compared with these formal texts, informal texts usually contain many informal inputs, such as acronyms (e.g also known as –> aka), abbreviation (e.g technology –> tech), out-of-vocabulary words (alooooo –> alo), no accent markers, etc. Recognizing its POS tags, hence is a challenging task esp. for poor-resource languages like Vietnamese.

In fact, most work about POS tagging on social media texts was dedicated to popular languages such as English (Gimpel et al., 2011), Chinese (Wang et al., 2019), Spanish (Meftah and Semmar, 2018), etc. Meanwhile, very little work was performed for POS tagging on social texts of poor-resource languages. For Vietnamese, we have noticed that there is only one group working on this problem. Bach et al. (Bach et al., 2018; Bach et al., 2019) built an annotated corpus for social POS tagging, which consists of more than four thousand sentences collected from Facebook. Based on this corpus, the authors performed extensive experiments to measure the performance of several proposed models including traditional CRFs and some deep neural network models.

Similar to the previous work, this study also focuses on POS tagging on social texts but with some main differences. Firstly, we choose to annotate another genre of social texts which is the conversational texts from the chatlog of customers in a famous e-commerce site. This kind of texts is slightly different with Facebook's texts of the previous work (Bach et al., 2018; Bach et al., 2019). For example, they are usually shorter, don't contain many emoticons, etc. Therefore, we have to re-design a new tagging scheme including exclusive tags for expressing special phenomena of conversational words. Secondly, we aim at constructing a much larger dataset which includes 16.310 social sentences (about four times as large as the previous one's). The model built on this corpus is expected to be particularly useful in developing applications to help computers directly interacting with users of systems such as sentiment analysis (Bach et al., 2015), chatbots (Tran and Luong, 2020), virtual assistant systems, dialog agents, etc. This dataset also facilitates the exploitation of strong deep learning methods which are data-hungry today. Thirdly, based on this dataset, we propose several strong methods exploiting state-of-the-art (SOTA) machine learning methods and conduct extensive experiments to provide the preliminary results for follow-up research on this direction. In conclusion, this paper makes the following contributions:

- Publish a new large-scale dataset[2] on the Vietnamese POS tagging task for online conversational texts.
- Based on that dataset, we perform extensive experiments using strong SOTA machine learning techniques

[1] It can be named as informal data, or unstandardized data.

[2] Contact the first author for getting the full corpus.

such as CRFs, deep neural networks and fine-tuning the pre-trained BERT model.

The remaining of this paper is organized as follows. Section 2 discusses the related work. Section 3 introduces a new large-scale dataset including the annotation process and some statistics about it. Section 4 describes our proposed methods exploited on this dataset. Experimental setups, experimental results, and error analysis are reported in Section 5. Finally, we conclude the paper and point out some future lines of work in Section 6.

2. Related Work

Nowadays, work about POS tagging on social texts is growing considerably. Many researches have been performed to contribute the corpora and/or develop different state-of-the-art POS taggers on different kinds of social texts for popular languages such as English, Chinese, German, etc.

For **English**, (Gimpel et al., 2011) presented a study on POS tagging for micro-blog Twitter in English. They proposed a POS tagset, annotated a dataset including 1,827 tweets, employed CRFs as the learning method and got nearly 90% accuracy on this dataset. (Owoputi et al., 2013) introduced a dataset for online conversational texts of English language and then proposed to utilize word cluster features to improve POS tagging for English tweets. (Derczynski et al., 2013) combined available POS taggers using different tagsets associated with assigning prior probabilities to some tokens and handling of unknown words and slang. For **German**, (Neunerdt et al., 2013) introduced a new social corpus and measured the performance of different SOTA POS taggers on this corpus. They showed that re-training the POS taggers on in-domain training data increases the tagging accuracies by more than five percentage points. (Proisl, 2018) described a POS tagger - SoMeWeTa – which is capable of domain adaptation and that can use various external resources. For **Italian**, (T. and Zesch, 2015) compared some domain adaptation approaches for PoS tagging of social media data. They concluded that the most effective approach is based on clustering of unlabeled data. In 2016, (Horsmann and Zesch, 2016) trained another model based on FlexTag using only the provided training data and external resources like word clusters and a POS dictionary which are build from public Italian corpora. This work was submitted to the PoSTWITA shared-task[3] for POS tagging of Italian social media text. For **Chinese**, (Wang et al., 2019) manually built a dataset of Chinese-English mixed social media texts and proposed a language-agnostic POS tagger for social media texts, which is able to learn from heterogeneous data with different genre and language type. For **multi-lingual**, (Meftah and Semmar, 2018) proposed a neural network model for POS tagging of social texts which uses both character and word level representations, combined with transfer learning approach. They demonstrated the validity and genericity of the model on a POS tagging task by conducting experiments on five social media languages (including English, German, French, Italian and Spanish).

While most researches done for popular languages, little work for poor-resource languages like Vietnamese has been performed so far. To our knowledge, there is only one research group of Bach et al. (Bach et al., 2018; Bach et al., 2019) dedicated to this issue. They have built a Vietnamese POS corpus including 4150 sentences with 24 POS labels on Facebook texts. Based on this dataset, several methods have been exploited such as CRFs with rich feature sets, deep neural networks (CNNs and biLSTMs).

In this paper, we focus on another genre of social texts - conversational texts from e-commerce sites. We aim at contributing a large-scale corpus of 16.310k sentences on this field and then evaluate SOTA POS tagging methods on this corpus including the previous methods done by (Bach et al., 2018; Bach et al., 2019) and the most recent and robust method, the BERT-based model. We also re-design the POS label set with exclusive tags for this kinds of texts at a more detailed level.

3. Building the Corpus

The goal of the project is the creation of a large-scale corpus of online conversational Vietnamese text with word segment and POS tagging information. This corpus will be released via this paper and then is available to the public research community. Figure 1 shows the annotation process which includes 4 main steps as follows:

Figure 1: The annotation process of manually building the Vietnamese POS tagging corpus on conversational texts.

- **Collecting raw texts**: The dataset is mainly collected from a social media resource which is a biggest chat-log history of a famous e-commerce website in Vietnam.
- **Pre-processing**: The data collected is pre-processed to extract only content texts. Because, the social texts are quite free-style, so we try cleaning to make them better for labeling. We also standardized character encoding of input texts to UTF-8 and then removed duplicate sentences. Finally, we obtained 16.310 plain sentences available for tagging.
- **Designing POS Labels**: We hired two linguistic experts, one expert comes from Vietlex - Vietnam Lexicography Centre[4] and another expert comes from Institute of Linguistics, Vietnam [5] together designed the POS tagsets and then developed guidelines for every single POS label. This documentation [6] has been revised several times during the project. Finally, we got the POS tagset including 36 tags as shown in Table 1. We explain each POS tag in more detail as follows:
  - Popular tags for adjective, adverb, pronoun, preposition, coordinator, noun, and verb words

[3] https://universaldependencies.org/it/overview/introduction.html

[4] www.vietlex.com

[5] http://vienngonnguhoc.gov.vn/

[6] It is available in Vietnamese. Contact the corresponding authors for more details about this.

No.	Samples	Phenomena
1	*pas*/Vb ban/N số/N khi/N nào/Pro có/V *b*/Ny *lh*/Vy nhe/Aux *đưa bạn số khi nào có bạn liên hệ lại nhé* give me the number so I can call you when ready	*abbreviation* *borrowed words* *numbers attached with uoms*
2	*e*/Ny thay/V *ship*/Vb ca/Q nuoc/N la/C *17k*/Numx *em thấy ship cả nước là 17k* I saw that the ship fee is 17k in the whole country	
3	Cho/V xin/V *contact*/Nb đi/Adv *Shop*/Nb *Cho xin địa chỉ liên hệ đi Shop* Please tell me your address, Shop	

Figure 2: Some examples of popular phenomena of social texts in the corpus.

such as *Aux, Adj, Adv, Pro, Pre, C, N, V* - are quite similar to POS tags on formal texts.

- *Num* for numbers in digits or in words. If the number is attached with other nouns (e.g unit-of-measurement (uom) like *kg, m, etc.*), assign its POS label as *Numx*.
- *NNP* for proper nouns such as person names, organization names, location names, product names, etc.
- *Nu* for uoms. *Nux* for extended uoms (e.g C_degree, ml/24h, cycles/minute, etc.)
- Attach *y* after the main POS tags for abbreviated words in sentences.
- *G* for word collocations which are commonly used and can be guessed its meaning.
- *X* for words that we can not determine their POS labels (eg. mathematical formula, Chinese words, or mis-spelling words which can not be translated, etc.)
- Attach *b* after the main POS tags for borrowed words which are considered as the new Vietnamese language. For other not-so-popular words, we label them as *FW*.
- For the remaining words that do not belong to the above tags, we assign them the POS tag *Others*.

In comparison to the POS tag set proposed in (Bach et al., 2018), our POS tag set contains exclusive tags to describe typical phenomena of social texts such as *Vy, Ny, Proy, Auxy, Vb, Nb, NNPy, Adjb, Numx, etc.* The reason is that the amount of words belonging to these types is quite high and accounts for about 15% of all words in the corpus. Figure 2 shows some sentences which contain these phenomena. Finally, the POS tag set consists of 36 detailed POS tags as expressed in Table 1) (extend addtional 12 POS tags in comparison to the POS tag set of the previous work).

- **Annotating the Dataset**: This stage includes at least two passes, that is, the data are annotated by one annotator, then the resulting files are checked by another annotator. When it is not clear whether a word in a sentence should be tagged as label *X* or *Y*, two annotators sat down to discuss and finalized a solution to follow.

After this process, we achieved the corpus with some statistics as shown in Table 1.

POS Tags	Quantities	POS Tags	Quantities
V	29.859	Nby	620
N	28.708	Q	531
Pro	12.546	Auxy	392
Adv	9.495	Nu	291
Aux	8.734	Adjb	261
Nb	8.380	G	198
Adj	7.885	Ib	196
Advy	6488	NNPy	168
C	5.870	Gy	129
SYM	4.442	Adjy	123
Ny	3.803	Cy	101
Pre	3.786	Vby	57
Num	3.652	FW	52
Nc	2.134	Numy	23
NNP	1.822	X	21
Numx	1.552	Others	85
I	1.444		
Proy	1.334		
Vy	1.276		
Vb	838		

Table 1: Some statistics about the corpus.

To measure the quality of the corpus, we use the Kappa coefficient (Cohen, 1960) agreement. The Cohen's kappa coefficient of our corpus was 0.94, which usually is interpreted as almost perfect agreement.

4. POS Tagging Models

This section presents our proposed POS tagging approaches performed on this dataset, which include CRFs with manually-built features and/or automatically-learnt features via different deep neural architectures, and the model of fine-tuning BERT.

4.1. Conditional Random Fields with handcrafted features

POS tagging can be solved by making the optimal label for a given word dependent on the choices of surrounding words. To this end, we use CRFs (Lafferty et al., 2001) which are widely applied, and yield state-of-the-art results in many sequence labelling problems. Specifically, the conditional probability of a state sequence $S =< s_1, s_2, ..., s_T >$ given an observation sequence $O =< o_1, o_2, ..., o_T >$ is calculated as:

$$P(s|o) = \frac{1}{Z} exp(\sum_{t=1}^{T} \sum_{k} \lambda_k \times f_k(s_{t-1}, s_t, o, t)) \quad (1)$$

where $f_k(s_{t-1}, s_t, o, t)$ is a feature function (manually designed or automatically learnt from deep neural models) whose weight λ_k is to be learned via training. To make all conditional probabilities sum up to 1, we must calculate the normalization factor Z over all state sequences:

$$Z = \sum_{s} exp(\sum_{t=1}^{T} \sum_{k} \lambda_k \times f_k(s_{t-1}, s_t, o, t)) \quad (2)$$

To build the strong model, CRFs need a good feature set. These features are manually extracted as proposed in (Bach et al., 2018):

- *Basic features*: Basic features consist of the set of all position-marked n-grams ($n = 1, 2, 3$) of words extracted in the window of size 5 centered around the current word.
- *Enhanced features*: a feature that checks whether the word contains special characters; whether the word contains digits; whether the word follows capitalization patterns, etc.
- *GENTAG features*: We also use the output (the predicted POS tags) of VnCoreNLP[7] a strong Vietnamese POS tagger trained on general text, as extra features.
- *Word-cluster features*: We use extra features derived from two word clustering models on the tagging performance, the Brown clustering method.

In this paper, we consider integrating an additional kind of features based on word vectors as follows (these features have not been used in the previous work):

- *Word vector features*: the pre-trained vector representation of words trained on a plain text corpus using the Glove method.

4.2. CRFs with automatically-learnt features via deep neural network architectures

CRFs need rich features to build the robust model. Instead of manually designing the feature sets, it is possible to automatically extract these features via neural network models. Figure 3 shows the architecture of applying deep neural networks to encode these features. This approach exploits non-linear neural networks which are LSTMs (Hochreiter and Schmidhuber, 1997) and CNNs (LeCun and Bengio, 1998) to encode character-level information of a t^{th} word into its character-level representation l_t. l_t was initialized randomly and trained with the whole network of CNNs or LSTMs. We then combine l_t with word-level representations w_t. w_t was also randomly initialized and fed into LSTMs or CNNs to capture the left and right context information of each word. The character and word representations are concatenated and then fed $x_t = concat(l_t, w_t)$ into bi-LSTM networks (Lample et al., 2016) to model context information of each word. Formally, the formulas to update an LSTM unit at time t are:

$$i_t = \sigma(W_i h_{t-1} + U_i X_t + b_i) \quad (3)$$

$$f_t = \sigma(W_f h_{t-1} + U_f X_t + b_f) \quad (4)$$

$$\tilde{c}_t = tanh(W_c h_{t-1} + U_c X_t + b_c) \quad (5)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (6)$$

$$o_t = \sigma(W_o h_{t-1} + U_o X_t + b_o) \quad (7)$$

$$h_t = o_t \odot tanh(c_t) \quad (8)$$

where σ is the element-wise sigmoid function and $\odot$ is the element-wise product. x_t is the input vector of the t_{th} word (concatenation of character and word embeddings) at time t. h_t is the hidden state vector storing all the useful information at (and before) the time t. U_i, U_f, U_c, U_o denote the weight matrices of different gates for the input x_t, and W_i, W_f, W_c, W_o are the weight matrices for hidden state h_t. b_i, b_f, b_c, b_o denote the bias vectors.

Here, we use the biLSTMs architecture to capture bi-directions to capture the representation of words in both directions. An forward LSTM calculates a representation of the left context of the sentence and a second backward LSTM that reads the same sequence in reverse. These two representations are concatenated and linearly projected onto a layer whose size is equal to the number of distinct contexts. We then use a CRF (Lafferty et al., 2001) as described in the previous section to take into account neighboring tags, yielding the final POS tag predictions for every word in the sentence. In the decoding phase, the Viterbi algorithm is chosen to find the best label sequence yielding the largest probability.

4.3. Integrating handcrafted features and automatically-learnt features into CRFs

Handcrafted features have been proven important in many sequence labeling tasks. Hence, from our own designed features m_t of a t^{th} word extracted in Section 4.1, we combine it with l_t and w_t to create a concatenated vector $x_t = concat(l_t, w_t, m_t)$ to feed into CRFs as shown in Figure 3. In experiments, we extract the features m_t for the t^{th} word as follows:

- A feature checks whether the word is a special character (hyphen, punctuation, dash, etc.)
- A feature detects whether the word contains digits
- Features looks for capitalization patterns (the first letter and all the letters) in the word

[7] https://github.com/vncorenlp/VnCoreNLP

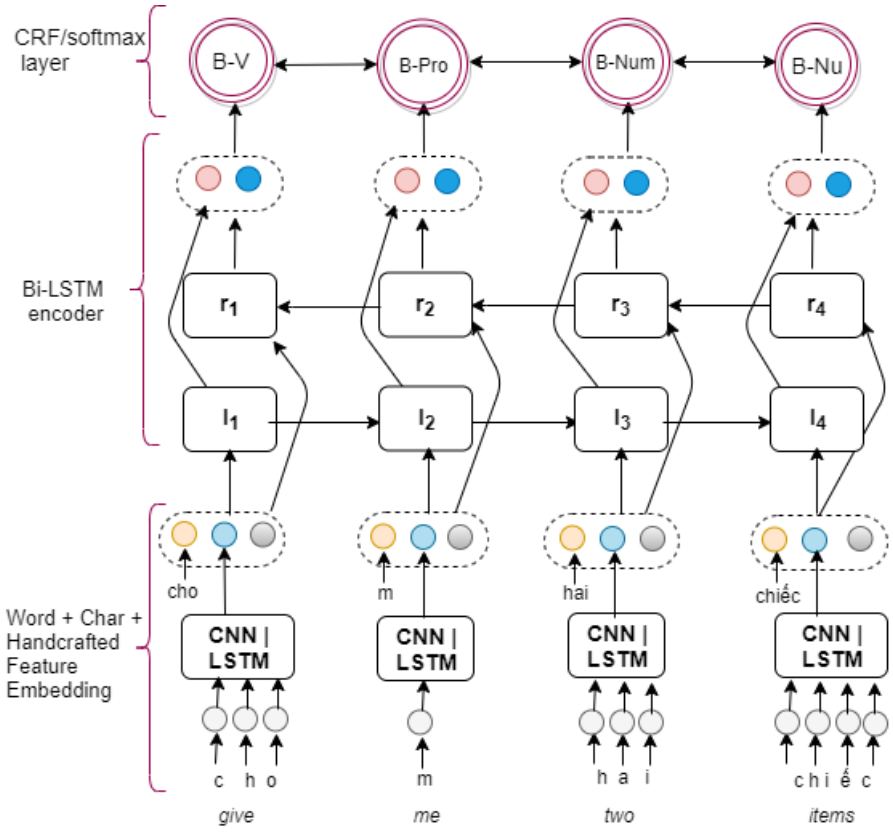

Figure 3: The deep neural network architecture using biLSTM.

- The predicted POS tag of the word - the predicted output of the vnCoreNLP tool, a widely-used Vietnamese POS tagger trained on general text

- Features of word-cluster information extracted using the Brown clustering algorithm

4.4. Fine-tuning the BERT-based model

The pre-trained language model, BERT (Devlin et al., 2019), has shown their effectiveness to alleviate the effort of feature engineering and has achieved excellent results in many NLP tasks. This motivates us to explore the effectiveness of BERT-based methods in predicting POS tags in Vietnamese social texts. BERT is deeply bidirectional, unsupervised language representation, pre-trained using only a plain text corpus. This contextual model generates a representation of each word based on the other words in the sentence. In other words, the word vector BERT outputs for a word is dependent on the surrounding context in which it occurs. This approach was shown to be a better or at least competitive alternative for many NLP tasks.

In this study, we exploit BERT to extract high-quality features for each word in sentences, and fine-tune BERT on the POS tagging task with the manually-built dataset to produce final POS tag predictions. Specifically, one fully-connected layer is added on top of BERT and trained for a few epochs.

5. Experiments

5.1. Experimental Setups

5.1.1. Pre-trained word embeddings and Brown word clustering

To create word embeddings, we collected the raw data from Vietnamese newpapers ($\approx$ 7GB texts) to train the word vector model using Glove[8]. The number of word embedding dimensions was fixed at 50.

Moreover, these raw texts were also used to induce clustering over words using Brown clustering algorithm (Brown et al., 1992). The number of clusters was set at 200. Features were extracted using 4-bit and 6-bit depth.

5.1.2. Evaluation metrics

The system performance is evaluated using precision, recall, and the F_1 score for each POS label as in many sequence labeling problems as follows:

$$F1 = \frac{2*precision*recall}{precision+recall}$$
$$precision = \frac{TP}{TP+FP}$$
$$recall = \frac{TP}{TP+FN}$$

In this context, TP (True Positive) is the number of POS tags that are correctly identified. FP (False Positive) is the number of POS tags that are not identified.

We also report the accuracy of different models which is computed over all kinds of POS tags as follows:

$$Accuracy = \frac{\#words\_correctly\_tagged}{\#of\_words}$$

[8] https://github.com/standfordnlp/GloVe

5.1.3. Model training

In performing experiments, we implemented the framework using CNNs and bi-LSTMs for detecting intents using pytorch library. For extracting contexts, we exploited three available tools with some modifications to fit the POS tagging task:

- CRFs: use the library of pyCRFsuite at https://python-crfsuite.readthedocs.io/en/latest/
- Deep Neural Network: use the NCRF++ toolkit published by (Yang and Zhang, 2018) at https://github.com/jiesutd/NCRFpp
- BERT: use the pre-trained model released by Google at https://github.com/google-research/bert/blob/master/multilingual.md

To conduct experiments, we randomly select 10% of the training data as the development set. The remaining 90% of the dataset is used for training and testing purposes. For each experiment type, we conducted 5-fold cross-validation tests. The hyper-parameters of models were chosen via a search on the development set.

In the deep neural settings, we varied different parameters of each model to find the optimized sets for filter windows sizes, dropout rates, optimization methods, learning rates, batch sizes, number of epochs, etc.

5.2. Experimental Results

This section presents four types of experimental results: the first one is to evaluate the performance of CRFs with rich features; the second one is to measure the effectiveness of the deep neural architectures in extracting features automatically; the third one is to see how handcrafted features and automatically-learnt features can boost the performance of the CRFs models; and the last one is to evaluate the strength of the latest BERT-based models on this dataset.

5.2.1. Experimental results of CRFs with handcrafted features

Table 2 shows the experimental results with different sets of features of the CRF model. We gradually added more features into the CRFs model to see the effects of each feature set on the POS tagging performance. As can be seen that adding more features slightly enhances the performance of the model. Experimental results once again confirm that the more features the model get, the higher the accuracy of the model (as stated by (Bach et al., 2018)). By integrating all kinds of features, the model *CRF_4* yielded the highest accuracy score of 91.78%.

Models	Features	Acc
CRF_1	Basic and advanced features	91.06
CRF_2	*CRF_1* + word cluster	91.18
CRF_3	*CRF_2* + GENTAG	91.71
CRF_4	*CRF_3* + word vectors	**91.78**

Table 2: Experimental results of CRFs with different sets of features (in %).

5.2.2. Experimental results of CRFs with automatically-learnt features via different deep neural architectures

In this experimental setting, we implemented several different neural architectures to extract character-level features using CNNs and LSTMs as illustrated in Section 4.2. For the word-level features, we use LSTMs to extract the features[9]. Table 3 indicates these experimental results. As shown in this table, the model using char(CNN) slightly outperformed the model using char(LSTM) on this dataset, however, the difference is not significant.

In comparison to CRFs with rich features, the best deep neural networks got a little bit lower performance. It degraded the accuracy by about 1% in the accuracy score. This result suggested that the effective traditional CRFs with rich features could still yield very good performance in comparison to advanced deep neural networks on the same dataset.

Methods	Features	Acc
DNN_1	char(LSTM) + word(LSTM)	90.59
DNN_2	char(CNN) + word(LSTM)	90.63
BERT	Fine-tuned BERT	92.42

Table 3: Experimental results of different deep neural architectures in learning features (in %).

Methods	Features	Acc
DNN_1++	*DNN_1* + handcrafted features	92.78
DNN_2++	*DNN_2* + handcrafted features	**93.36**

Table 4: Experimental results of combining both the handcrafted features and features learnt from deep neural architectures (in %).

5.2.3. Experimental results of fine-tuning BERT for POS tagging

We fine-tuned BERT multi-lingual[10] for this POS tagging task. Experimental results shown in Table 3 show that the model yielded 92.42% in the accuracy score. It is much higher than the performances of two previous approaches (about 0.64% higher than *CRF_4* and 1.79% higher than *DNN_2* in the accuracy score). This result proved that BERT-based model is very effective for this task.

5.2.4. Experimental results of combining both handcrafted features and automatically-learnt features into CRFs

Table 4 shows the experimental results of integrating some manually-built features and automatically-learnt features via deep neural architectures into CRFs, namely *DNN_1* and *DNN_2*.

The experimental results indicated that both two models significantly improves the accuracy of the POS tagger. Using *DNN_1++* and *DNN_2++*, we achieved 92.78% and

[9] The performance of models using pre-trained word embeddings did not yield good results on this dataset.

[10] https://github.com/google-research/bert/blob/master/multilingual.md

POS Tags	CRFs_4			BERT			DNN_2++			DNN_2		
	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
V	88.09	91	89.52	90.38	92.65	91.5	91.68	92.98	**92.33**	88.35	90.58	89.45
N	86.34	88.8	87.55	89.38	90.01	89.7	89.1	90.46	**89.77**	86.03	87.75	86.88
Pro	96.24	96.06	96.15	96.02	96.88	96.45	97.23	97.51	**97.37**	95.31	95.95	95.63
Adv	89.94	90.01	89.96	93.39	92.32	92.85	93.94	94.07	**94**	91.36	90.59	90.97
Aux	95.42	94.76	95.09	96.08	96.15	96.11	97	96.36	**96.68**	94.71	94.3	94.5
Nb	96.5	94.96	95.72	96.09	96.36	96.22	96.97	95.9	**96.43**	96.71	94.72	95.7
Adj	86.4	83.91	85.13	86.13	85	85.53	88.18	88.07	**88.13**	83.53	83.46	83.49
Advy	96.17	96.41	96.29	96.96	97.32	97.13	96.95	97.72	**97.33**	96.4	97.03	96.71
C	91.5	91.24	91.36	93.45	93.72	93.58	93.13	95.07	**94.09**	90.6	92.85	91.71
SYM	98.97	99.39	99.18	99.16	99.53	**99.34**	98.75	99.41	99.08	98.99	99	98.99
Ny	94.21	92.94	93.57	95.54	95.68	95.61	96.32	95.51	**95.91**	95.05	94	94.52
Pre	87.23	87.65	87.43	89.03	91.8	90.38	91.46	92.86	**92.13**	87.63	87.88	87.75
Num	94.63	97.92	96.24	96.53	97.68	97.09	95.88	98.58	**97.21**	94.36	97.59	95.94
Nc	91.88	92.79	92.33	91.28	94.05	92.61	91.08	95.08	**93.03**	90.67	92.28	91.47
NNP	77.85	66.62	71.78	83.75	74.27	**78.63**	83.55	73.77	78.22	73.72	60.68	66.32
Numx	94.98	96.29	95.62	95.86	97.87	**96.85**	95.82	96.96	96.39	95.66	96.12	95.89
I	95.02	94.12	94.56	96.31	95.05	**95.67**	96.41	94.44	95.41	96.02	92.44	94.2
Proy	92.12	93.52	92.81	94.25	95.4	94.82	94.74	96.24	**95.48**	94.07	93.83	93.94
Vy	82.68	80.19	81.39	85.3	88.95	87.03	88.31	86.12	**87.18**	86.55	82.98	84.72
Vb	95.21	88.79	91.86	95.78	92.92	**94.32**	96.01	92.43	94.15	95.56	89.77	92.55
Nby	89.11	78.43	83.41	88.79	88.6	88.67	92.42	87.38	**89.83**	92.74	85.72	89.08
Q	82.35	66.01	73.19	71.84	78.23	74.52	84.39	82.47	**83.32**	78.14	67.3	72.11
Auxy	92.7	93.12	92.89	90.81	96.79	93.7	94.92	95.33	**95.1**	91.97	94.96	93.41
Nu	87.07	74.65	80.22	85.29	81.35	**83.18**	85.83	76.89	81.04	84.62	72.1	77.85
Adjb	90.55	80.06	84.93	90.6	90.18	90.17	94.73	88.01	**91.11**	94.73	82.74	88.25
G	91.44	77.1	83.56	86.76	79.33	82.82	95	83.81	**89.01**	89.83	76.42	82.54
Ib	95.77	92.84	94.18	96.18	98.75	**97.43**	94.35	98.3	96.2	91.79	92.8	92.23
NNPy	91.29	61.18	73.07	80.96	82.33	81.37	91.95	80.04	**85.33**	91.25	78.52	84.04
Gy	94.3	72.8	81.61	88.79	87.16	87.86	93.33	88.64	**90.86**	88.73	85.03	86.74
Adjy	63.49	29.04	39.3	66.52	54.91	59.52	83.42	52	**63.43**	78.62	41.11	53.81
Cy	82.91	71.71	76.27	87.91	82.29	84.71	90.19	86.28	**88.13**	85.38	88.31	86.77
FW	32.5	24.23	27.74	33.33	2.08	3.92	64.29	29.1	**32.7**	50	9.38	15.79
Other	61.67	22.38	31.6	57.79	35.48	43.56	72.45	40.02	**49.99**	66.67	43.11	49.06
Vby	86.76	81.05	83.23	89.99	83.72	85.8	88.46	86.09	**86.47**	89.68	78.85	82.37
Numy	60	16.19	25	96	83.81	**88.24**	100	72.14	81.6	66.67	27.78	38.89

Table 5: Experimental results of each POS tag on the four best models of the four approaches. (in %).

93.36% in the accuracy scores, respectively. The *DNN_2++* 's accuracy is 1.6% higher than *CRF_4* and 2.7% higher than *DNN_2*. General speaking, it can be said that hand-crafted features complement neural nets for the POS tagging task. The best model *DNN_2++* also yielded slightly better performance than the BERT-based model by 0.94% in the accuracy score.

Based on the best models of the four tagging methods, which are *CRF_4, BERT, DNN_2*, and *DNN_2++*, we also measured the tagging performance on each POS tag as shown in Table 5. We can see that the four models could produce the very good performance on most POS tags with more than 80% in the F1 score. There are several POS tags whose tagging performances are quite low, for example the tags of *FW, Others*, and *Adjy*. The reason might be that their number of instance is remarkably smaller than others'. Among these four models, *DNN_2++* and *BERT* received the best or the second best F1 scores on all POS tags. Specifically, *DNN_2++* achieved the highest F1 scores on 27 POS tags. For the remaining eight POS tags, the model *BERT* yielded the highest performance.

5.2.5. Learning curve with different sizes of training data set

Figure 4 shows the graph drawing the learning curve of our best model, *DNN_2++*, when the size of training data is gradually increased from 10% upto 100%. This graph illustrates the relationship between the number of the training samples and the performance of the best model. The graph clearly indicates that the performance consistently enhances as the size of training data set increases and still continues to improve even when the size of the training data reaches 100%. This result suggests that it is possible for us to boost the performance of the POS tagging model by increasing the annotated dataset in the training phase.

5.2.6. Error Analysis

From the best POS tagging model *DNN_2++*, we performed analyzing typical errors generated. Table 6 shows the statistics about these errors. The second column of the table presents the wrong tags predicted for the gold tags observed

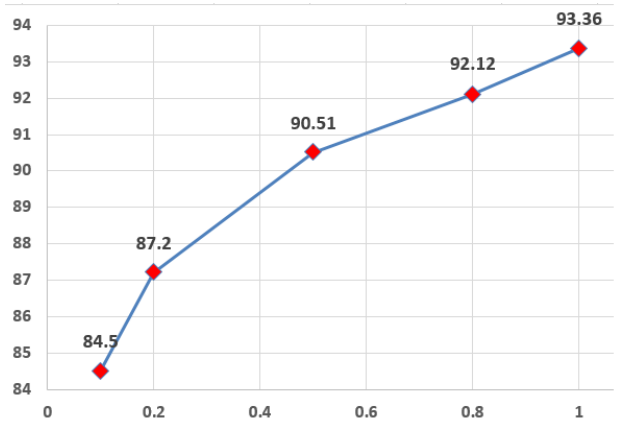

Figure 4: The relationship between the accuracy of the *DNN_2++* and the different sizes of training dataset.

over 5 folds. We listed the three most popular wrongly-predicted tags along with their rates. We acknowledge several typpical error types as follows:

- The most popular errors were incorrectly assigning POS tags of words into the most frequent tags in the dataset, e.g. the POS tags of *V, N, Adj, and Adv.*
- The tag *NNP* was usually confused with *N* because most *NNP*'s instances in the dataset were in lowercases.
- Most words belonging to the tag *FW* (with a low accuracy score of 32.7%) were assigned to nouns because they occurred few times in the dataset.

6. Conclusion

This paper introduced a new large-scale dataset about Vietnamese POS tagging on online conversational texts. After carefully investigating the raw texts, we proposed a new tagging schema of 36 POS tags which includes exclusive tags to describe some special phenomena of conversational texts such as abbreviation and borrowed words. This corpus of 16.310 social sentences will be published to the NLP research community.

Based on this public dataset, we conducted a wide range of experiments using SOTA machine learning methods for POS tagging including methods done by previous researches as well as the recent advanced machine learning method, the BERT-based method. The experimental results from extensive experiments suggested that using the same sequence labelling architecture, namely CRFs, we can significantly enhance the performance of the POS tagger by two ways. The first one is to integrate more and more features by both manually designing and automatically learning via deep neural architectures. In addition, these results can be improved more by enriching the training dataset. We also acknowledged that the pre-trained BERT model is quite robust. It achieved much higher performance in comparison to previous deep neural approaches. This model is even strongly competitive with all the deep neural networks integrated rich handcrafted features. The best model of *DNN_2++* yielded the best accuracy of 93.36%, and the best F1 score on most POS tags of the dataset.

In the future, we will investigate more models to boost the performance of tagging. In addition, we will also verify the effectiveness of this POS tagging model on other downstream tasks in NLP such as named entity recognition or intent detection tasks.

Gold Tags	Wrong Tags			Gold Tags	Wrong Tags		
V	N 40.9	Adv 17.5	Adj 12.1	Vy	Advy 52.5	V 16.7	Ny 12.4
N	V 41.0	Adj 16.3	Pro 6.3	Vb	Nb 43.9	V 21.1	N 12.3
Pro	N 40.9	V 15.4	Aux 15.4	Nby	N 37.1	Nb 17.1	V 10
Adv	V 51.7	N 20	Adj 9.9	Q	Num 31.3	N 16.3	Adv 12.5
Aux	V 22	Pro 18.9	C 11.5	Auxy	Proy 50	Cy 18.8	Aux 12.5
Nb	N 53.3	V 14.3	Adj 8.7	Nu	N 40	Advy 15	Numx 10
Adj	N 41.1	V 36.9	Adv 8.3	Adjb	Nb 27.6	Ib 20.7	N 17.2
Advy	Vy 66.4	V 6.9	Ny 5.3	G	V 34.5	Adv 20.7	Adj 17.2
C	V 48.5	Adv 18.1	N 13.3	Ib	Adjb 66.7	I 33.3	
Ny	N 25.9	Vy 16.1	Aux 13.3	NNPy	N 45.2	NNP 12.9	Nb 12.9
Pre	V 55.9	C 16	N 15.5	Gy	Advy 38.5	Ny 23.1	V 15.4
Num	Numx 28.6	N 28.6	Nu 14.3	Adjy	Advy 41.5	N 15.1	Adj 15.1
Nc	N 53.2	V 30	Nb 6.4	Cy	Advy 45.5	Proy 36.4	N 9.1
NNP	N 47.7	V 15.4	Nb 12.6	Vby	V 44.4	Vb 22.2	Nby 22.2
Numx	Num 75	Nby 12.5	N 6.3	FW	N 29.8	V 22.5	NNP 8.1
I	N 26.4	V 23.6	Aux 14	Numy	Numx 80	N 20	
Proy	Ny 24	Auxy 21.7	N 21.7	X	SYM 31.6	NNP 26.3	Numx 10.6

Table 6: The most wrongly-predicted tags by the best POS tagging model.

7. Bibliographical References

Bach, N., Van, P., Tai, N., and Phuong, T. (2015). Mining vietnamese comparative sentences for sentiment analysis. In Proceedings of the 7th international conference on Knowledge and Systems Engineering (KSE), pages 162–167. IEEE.

Bach, N., Linh, N., and Phuong, T. (2018). An empirical study on pos tagging for vietnamese social media text. *Computer Speech and Language*, 50:1–15.

Bach, N., Duy, T., and Phuong, T. (2019). A pos tagging model for vietnamese social media text using bilstm-crf with rich features. In Proceedings of the 16th Pacific Rim International Conferences on Artificial Intelligence (PRICAI), Part III, pages 206–219.

Brown, P., deSouza, P., Mercer, R., Pietra, V., and Lai, J. (1992). Class-based n-gram models of natural language. *Journal of Computational Linguistics*, 18(4):467–479.

Cohen, K. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Derczynski, L., Ritter, A., Clark, S., and Bontcheva, K. (2013). Twitter part-of-speech tagging for all: Overcoming sparse and noisy data. In Proceedings of the Inter-

national Conference Recent Advances in Natural Language Processing RANLP, pages 198—-206. INCOMA Ltd. Shoumen, BULGARIA.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186.

Gimpel, K., Schneider, N., O'Connor, B., D., D., M., D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., and Smith, N. (2011). Part-of-speech tagging for twitter: annotation, features, and experiments. In HLT '11 Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2, pages 42–47. Association for Computational Linguistics.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Journal Neural Computation*, 9(8):1735–1780.

Horsmann, T. and Zesch, T. (2016). Building a social media adapted pos tagger using flextag – a case study on italian tweets. In Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian - EVALITA 2016.

Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: probabilistic models for segmenting and labeling sequence data. In Proceedings of the ICML, pages 282–289.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named entity recognition. In 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 260–270. Association for Computational Linguistics.

LeCun, Y. and Bengio, Y. (1998). Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks, pp. 255-258. MIT Press Cambridge, MA, USA.

Meftah, S. and Semmar, N. (2018). A neural network model for part-of-speech tagging of social media texts. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018).

Neunerdt, M., Trevisan, B., Reyer, M., and Mathar, R. (2013). Part-of-speech tagging for social media texts. In Lecture Notes in Computer Science book series (LNCS, volume 8105), pages 139–150. Springer-Verlag Berlin Heidelberg.

Owoputi, O., O'Connor, B., Dyer, C., Gimpel, K., Schneider, N., and Smith, N. (2013). Improved part-of-speech tagging for online conversational text with word clusters. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 380–390, Atlanta, Georgia. Association for Computational Linguistics.

Proisl, T. (2018). Someweta: A part-of-speech tagger for german social media and web texts. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). European Language Resources Association (ELRA).

T., H. and Zesch, T. (2015). Effectiveness of domain adaptation approaches for social media pos tagging. In Proceedings of the Second Italian Conference on Computational Linguistics CLiC-it 2015, pages 166–170. Accademia University Press.

Tran, O. and Luong, T. (2020). Understanding what the users say in chatbots: A case study for the vietnamese language. *The International Scientific Journal Engineering Applications of Artificial Intelligence*, 87:1–10.

Wang, D., Fang, M., Song, Y., and Li, J. (2019). Bridging the gap: Improve part-of-speech tagging for chinese social media texts with foreign words. In Proceedings of the 5th Workshop on Semantic Deep Learning (SemDeep-5), pages 12–20. Association for Computational Linguistics.

Yang, J. and Zhang, Y. (2018). NCRF++: An open-source neural sequence labeling toolkit. In Proceedings of ACL 2018, System Demonstrations, pages 74–79, Melbourne, Australia, July. Association for Computational Linguistics.