An Empirical Study of Discriminative Sequence Labeling Models for Vietnamese Text Processing

Phuong Le-Hong*, Minh Pham Quang Nhat†, Thai-Hoang Pham†, Tuan-Anh Tran§, Dang-Minh Nguyen‡

*College of Science, Vietnam National University, Hanoi, Vietnam
†FPT Technology Research Institute, FPT University, Hanoi, Vietnam
§Alt Vietnam
‡FPT Technology Innovation, FPT Corp., Vietnam

Abstract—This paper presents an empirical study of two widely-used sequence prediction models, Conditional Random Fields (CRFs) and Long Short-Term Memory Networks (LSTMs), on two fundamental tasks for Vietnamese text processing, including part-of-speech tagging and named entity recognition. We show that a strong lower bound for labeling accuracy can be obtained by relying only on simple word-based features with minimal hand-crafted feature engineering, of 90.65% and 86.03% performance scores on the standard test sets for the two tasks respectively. In particular, we demonstrate empirically the surprising efficiency of word embeddings in both of the two tasks, with both of the two models. We point out that the state-of-the-art LSTMs model does not always outperform significantly the traditional CRFs model, especially on moderate-sized data sets. Finally, we give some suggestions and discussions for efficient use of sequence labeling models in practical applications.

I. INTRODUCTION

Many datasets, such as text collections and genetic databases, consist of sequences of distinct values. For applications that use such datasets, we often need to predict the sequence of labels given an observation sequence. In sequence prediction problems, we attempt to predict elements of a sequence on the basis of the preceding elements. Many statistical sequence models have been developed for sequence prediction, for example hidden Markov models (HMM) [4], maximum entropy Markov models (MEMMs) [5], conditional random fields (CRFs) [6] or recurrent neural nets (RNNs) [7]. These are all powerful probabilistic tools for modeling sequential data and have been applied to many text-related tasks such as part-of-speech tagging, named entity recognition, text segmentation and information extraction. These models also support applications in bioinformatics such as expressed sequence tag finding and gene discovery.

In this paper, we present an empirical study of two prevalent discriminative sequence labeling models, CRFs and LSTMs, on two fundamental problems of text processing, namely part-of-speech tagging and named entity recognition. Experiments are carefully designed, carried out and analyzed on standard Vietnamese data sets. The main findings of this work are as follows. First, we show that we can obtain a strong performance lower bound for both part-of-speech (POS) tagging and named entity recognition (NER) by using only simple and word-based features, both with CRFs and LSTMs. For POS tagging, we achieve a test accuracy of 90.65% by using only word identities, word shapes and word embedding features on sentences less than 25 tokens. For NER with the same feature set and sentence length, we obtain about 86.03% of F1-score. Second, we show that word embeddings are very effective and beneficial for both of the two tasks. They help improve POS tagging accuracy significantly by about 4.0% when using with LSTMs or 1.35% when using with CRFs. Word embeddings are even more beneficial for NER — they help improve recognition performance by more than 5% in both of the models. Third, we show that although the LSTMs model slightly outperforms CRFs in terms of accuracy, the gap is relatively small, especially on moderate-sized data sets, with the cost of much longer training time. Finally, this paper gives some suggestions for efficient use of sequence labeling models in practical applications.

The remainder of this paper is structured as follows. Section III presents the adopted methodology. Section III describes detailed settings and experimental results. Section VI gives discussions and findings. Section VII presents related work. Finally, Section VIII concludes the paper.

II. METHODOLOGY

A. Fundamental Tasks

This subsection gives a brief description of two fundamental-sequence learning tasks investigated in this study, part-of-speech tagging and named entity recognition.

1) Part-of-Speech Tagging: POS tagging is a typical sequence prediction task, where we are interested in building a model that reads text in some languages and assigns a part-of-speech to each token (word), such as noun, verb, adjective. In general, POS taggers in computational applications use more fine-grained POS tags like common noun or proper noun. For example, each word of the following English sentence is tagged with its most likely correct part-of-speech:

Proi/NN soared/VB at/P Boeing/NP Co./NN, easily/ADJ ADVP topping/VB forecasts/VB on/P Wall/NN Street/NN, as/P their/PPOSS CEO/NN Alan/NP Mulally/NP announced/VB first/ADJ ADJ quarter/NN results/NN.

where the tags N, V, P, ADV, ADJ denotes a noun, a verb, a preposition, an adverb, an adjective, respectively.

2) Named Entity Recognition: Named entity recognition, also known as entity identification is a subtask of information extraction that aims to locate and classify elements in texts into pre-defined categories such as the names of persons, organizations, locations and so on.
For example, the named entities extracted from the same English sentence above are as follows:

Profits soared at [Organization Boeing Co.], easily topping forecasts on [Location Wall Street], as their CEO [Person Alan Mulally] announced first quarter results.

In this example, an organization name, a location and a person name have been detected and classified. Actually, NER can be formalized as a sequence tagging problem, where each token is tagged with a specific tag, for example:

Profits/O soared/O at/O Boeing/B-ORG Co./I-ORG .O easily/O topping/O forecasts/O on/O Wall/B-LOC Street/I-LOC .O as/O so/O their/O CEO/O Alan/B-PER Mulally/I-PER announced/I/O first/O quarter/O results/O.

Here, the tag O means “no entity”, the tags B-ORG and I-ORG mean “begin organization” and “in organization” respectively; similarly, the tags B-LOC and I-LOC mean “begin location” and “in location” respectively, and so on.

B. Discriminative Models

In this subsection, we give a brief description of two discriminative sequence models used in this study, including Conditional Random Fields (CRFs), and Long Short-Term Memory Recurrent Neural Networks (LSTMs).

1) Conditional Random Fields: Conditional Random Fields (CRF) [4] is a discriminative probabilistic framework, which directly model conditional probabilities of a tag sequence given a word sequence. Formally, in CRF, the conditional probability of a tag sequence $y = (y_1, y_2, \ldots, y_T)$, given a word sequence $x = (x_1, x_2, \ldots, x_T)$ is defined as follow.

$$ P(y | x) \propto \exp(\sum_j \lambda_j t_j(y_{i-1}, y_i, x_i) + \sum_k \mu_k s_k(y_i, x_i)) $$

where $t_j(y_{i-1}, y_i, x_i)$ is a transition feature function of the entire observation sequence and the labels at the position $i$ and $i - 1$ in the label sequence; $s_k(y_i, x_i)$ is a state feature function of the label at the position $i$ and the observation sequence; $\lambda_j$ and $\mu_k$ are parameters to be estimated from training data.

We can simplify the notations by writing $s(y_i, x_i) = s(y_{i-1}, x_i)$ and

$$ F_j(y, x) = \sum_{i=1}^n f_j(y_{i-1}, y_i, x_i) $$

where each $f_j(y_{i-1}, y_i, x_i)$ is either a state function $s(y_{i-1}, y_i, x_i)$ or a transition function $t(y_{i-1}, y_i, x_i, i)$. By using this notation, we can write the conditional probability as follows:

$$ P(y | x, \lambda) = \frac{1}{Z(x)} \exp(\sum_j \lambda_j F_j(y, x)) $$

$Z(x)$ is a normalization factor.

The parameters in CRF can be estimated by maximizing log-likelihood objective function:

$$ L(\lambda) = \sum_k \left[ \log \frac{1}{Z(x^{(k)})} + \sum_j \lambda_j F_j(y^{(k)}, x^{(k)}) \right] $$

Parameter estimation in CRF can be done by using iterative scaling algorithms [6], [4], [7] or gradient-based methods [8].

2) Long Short-Term Memory Networks: Recurrent Neural Networks (RNNs) have recently been widely used for sequence labelling because they can directly represent sequential structures such as word sequences, sounds and time series data. For this reason, there is a rapidly growing interest in using RNNs for practical applications as an efficient method to map input sequences to output sequences. They are computationally more powerful and biologically more plausible than other adaptive approaches such as Hidden Markov Models (no continuous internal states), Feed-Forward Neural Networks (FFNN) and Support Vector Machines (no internal states at all). Traditional RNNs of the 1990s could not learn to look far back into the past because of the vanishing or exploding gradient problems. A feedback network called Long Short-Term Memory (LSTM) [9] was proposed to overcome these problems.

We represent the word sequence of a sentence with a collection of multi-dimensional vectors, including an input gate $i_t$, a forget gate $f_t$, an output gate $o_t$, a memory cell $c_t$, and a hidden state $h_t$. The unit takes as input a d-dimensional input vector $x_t$, the previous hidden state $h_{t-1}$, the previous memory cell $c_{t-1}$, and calculates the new vectors using the following six equations:

$$ i_t = \sigma(W^i x_t + U^i h_{t-1} + b^i) $$

$$ f_t = \sigma(W^f x_t + U^f h_{t-1} + b^f) $$

$$ o_t = \sigma(W^o x_t + U^o h_{t-1} + b^o) $$

$$ c_t = i_t \cdot u_t + f_t \cdot c_{t-1} $$

$$ h_t = o_t \cdot \tanh(c_t), $$

where $\sigma$ denotes the logistic function, the dot product denotes the element-wise multiplication of vectors, $W$ and $U$ are weight matrices and $b$ are bias vectors. The LSTM unit at $t$-th word receives the corresponding word embedding as input vector $x_t$. Since the LSTM is bidirectional, we concatenate the hidden state vectors of the two directions’ LSTM units corresponding to each word as its output vector and pass it to the subsequent layer.

III. EXPERIMENTS

A. Datasets

1) Part-of-Speech Tagging: We perform experiments on Vietnamese part-of-speech tagging using the standard part-of-speech tagged corpus of the VLSP project[4]. This corpus contains 10,165 manually tagged sentences where the training set contains 9,000 sentences and the test set contains 1,165 sentences. The tagset has 21 different tags. Further details of the corpus are described in [11].

2) Named Entity Recognition: For experiments on named entity recognition, we use the standard NER corpus developed by the Vietnamese Language and Speech Processing community in late 2016. Similar to the CoNLL 2003 NER corpus for English, four named entity types are considered, including

\[1\] Many interesting details of RNNs are available online at http://people.idsia.ch/~juergen/rnn.html

\[3\] https://vlsp.hpa.vn/demo/?page=home

\[4\] http://vlsp.org.vn/


Table I: Statistics of named entities in the VLSP corpus

| Entity Types            | Training Set | Test Set |
|-------------------------|--------------|----------|
| Location                | 6,237        | 1,379    |
| Organization            | 1,213        | 274      |
| Person                  | 7,480        | 1,294    |
| Miscellaneous names     | 282          | 49       |
| All                     | 15,222       | 2,996    |

Table II: Word shape features

| Feature                          | Example                  |
|-----------------------------------|--------------------------|
| is lower word                     | “tinh”                   |
| is capitalized word               | “tinh, cn”               |
| contains all capitalized letters (allcaps) | “T/N”                  |
| is name - consecutive syllables are capitalized | “Hà_Nộ”, “Đàn_Mê_H出品” |
| is mixed case letters             | “@Phone”                 |
| is capitalized letter with period | “Th.”, “H.”, “U.S.”     |
| contains hyphen                   | “H.”                     |
| is number                         | “100”                    |
| is date                           | “20-10-1980”, “10/10”    |

persons (PER), organizations (ORG), locations (LOC), and miscellaneous entities (MISC). The data are collected from electronic newspapers published on the web. Table III shows the quantity of named entity annotated in the training set and the test set.

B. Feature Sets

1) Word Identities: The first basic feature set contains only word occurrence information extracted from the training set. All words having an occurrence frequency above a minimum threshold are kept in a vocabulary \( V \). Each word can be represented by an one-hot sparse vector of size \( |V| \).

2) Word Shapes: In addition to word identities, word shapes have been shown to be important features for improving prediction ability, especially for unknown or rare words. Common word shape features used in our experiments are shown in Table II. We used regular expressions to extract those word shape features.

3) Word Embeddings: Word embeddings are low-dimensional distributed representation of words. Each word embedding is a real-valued vector of \( d \) dimensions where \( d \) is much smaller than \( |V| \) of its one-hot sparse representation. Distributed word representations have been shown very useful for many natural language processing tasks. Many state-of-the-art language processing models are now employing word or character embeddings. In particular, some previous works have also integrated Vietnamese word embeddings to improve performance [12], [13].

To create distributed word representations, we use a dataset consisting of 7.3GB of text from 2 million articles collected through a Vietnamese news portal [10]. The text is first normalized to lower case and all special characters are removed except these common symbols: the comma, the semicolon, the colon, the full stop and the percentage sign. All numerical sequences are replaced with the special token <number>, so that correlations between certain words and numbers are correctly recognized by the neural network or the log-bilinear regression model.

Each word in the Vietnamese language may consist of more than one syllables with spaces in between, which could be regarded as multiple words by the unsupervised models. Hence it is necessary to replace the spaces within each word with underscores to create full word tokens. The tokenization process follows the method described in [14]. After removal of special characters and tokenization, the articles add up to 969 million word tokens, spanning a vocabulary of 1.5 million unique tokens. We train the unsupervised models with the full vocabulary to obtain the representation vectors, and then prune the collection of word vectors to the 65,000 most frequent words, excluding special symbols and the token <number> representing numeral sequences. We train the Mikolov’s continuous Skip-gram model using the neural network and source code introduced in [15]. The continuous skip-gram model itself is described in details in [16]. Each word is represented by a real-valued vector of 25 dimensions.

In both CRF and LSTM models, we use three kinds of features mentioned above. In the CRF model, we represent word identities, word shapes as binary features. Each dimension of a word-embedding vector is a feature and its value is the feature value. In the LSTM model, we use word shape and word embedding features as additional dimensions in vector representation for each word. Thus, each word in the LSTM model is represented by a vector of size \( |V| + 34 \) (\( |V| \) is the vocabulary size; we use 9 word shape features and 25-dimension word-embedding vectors).

C. Evaluation Method

For the POS task, our system is evaluated by the tagging accuracy on the corresponding data sets. The accuracy is the ratio of number of tokens which are correctly tagged divided by the total number of tokens in the test set. For the NER task, the performance of our system is measured with \( F_1 \) score: \( F_1 = 2 \times P \times R / (P + R) \). Precision (\( P \)) is the percentage of named entities found by the learning system, which are correct predictions. Recall (\( R \)) is the percentage of named entities present in the corpus that are found by the system. A named entity is correct only if it is an exact match of the corresponding entity in the data file. The performance of our system is evaluated by the automatic evaluation script of the CoNLL 2003 shared task [11].

D. Experimental Settings

In the experiments, we fix the minimum frequency threshold for features as 5. In other words, all words or tags which do not appear at least 5 times in the training corpus are considered unknown. In our experiments with the CRF model, we adopted CRFSuite [17], an implementation of linear-chain (first-order Markov) CRF. That toolkit allows us to easily incorporate both binary and numeric features such as word embedding features. We use default setting of CRFSuite in which the training algorithm is L-BFGS [18] and L2 regularization is used. The coefficient for L2 regularization is 1.0.

The recurrent neural networks all have one bidirectional recurrent layer of different numbers of units whose activation function is tanh. The output layer uses softmax activation function as usual. The multiclass cross entropy loss function...
is selected. The network is trained by using the stochastic gradient descent optimization algorithm with learning rate fixed at 0.01. The Xavier initializer is used for parameter initialization [19]. We use early stopping when training the network to help avoid overfitting and remove the need to manually set the number of training epoch. The training terminates either if the training score does not improve for three consecutive epochs or if the number of epochs reaches 400.

**E. Results**

This subsections presents experimental results of the models on the two tasks. We first present results of part-of-speech tagging, and then those of named entity recognition.

1) **Part-of-Speech Tagging:** In the first experiment, we train and compare performance of sequence models on sentences of length not greater than 20 tokens. There are 4,879 training sentences and 570 test sentences. The vocabulary size is 1,630.

We train different LSTMs with varying number hidden units in the range from 32 to 200. Table III shows their performance on the feature set \{word identity, word shapes\}. We see that the larger number of hidden units is, the better result the tagger can achieve on the test set. The LSTMs tagger achieves an accuracy of 90.65% on the test set when the network has 200 hidden units.

In the second experiment, we add word embeddings as features to the LSTMs model to see whether they are helpful or not. Table IV shows their performance on the feature set \{word identity, word shapes, word embeddings\}.

It is surprising that word embeddings helps improve the accuracy of the tagger significantly. With the same training parameters, we are able to boost the accuracy on the test set from 85.98% to 89.92%. This result demonstrates that in LSTMs, it is beneficial to combine both discrete features and continuous features to build a better tagger.

---

**Table III: Performance of LSTMs for PoS tagging using word identities and word shapes**

| Hidden Units | Test Acc. | Training Acc. |
|--------------|-----------|---------------|
| n ≤ 20       | n ≤ 25    | n ≤ 20        | n ≤ 25        |
| 16           | 84.85     | 86.45         | 94.91         | 96.40         |
| 32           | 83.41     | 83.89         | 94.43         | 93.18         |
| 64           | 83.82     | 85.93         | 96.04         | 96.23         |
| 100          | 84.29     | 85.83         | 93.93         | 95.74         |
| 128          | 84.67     | 86.15         | 97.49         | 97.52         |
| 150          | 85.84     | 86.33         | 97.49         | 97.27         |
| 200          | 85.98     | 87.46         | 97.27         | 97.46         |

**Table IV: Performance of LSTMs for PoS tagging using word identities, word shapes and word embeddings**

| Hidden Units | Test Acc. | Training Acc. |
|--------------|-----------|---------------|
| n ≤ 20       | n ≤ 25    | n ≤ 20        | n ≤ 25        |
| 16           | 87.02     | 88.73         | 96.60         | 97.79         |
| 32           | 88.44     | 86.39         | 98.46         | 91.41         |
| 64           | 88.29     | 88.56         | 99.06         | 98.55         |
| 100          | 89.05     | 90.24         | 99.21         | 98.84         |
| 128          | 89.24     | 89.31         | 98.79         | 98.26         |
| 150          | 89.83     | 89.65         | 99.33         | 99.35         |
| 200          | 89.92     | 90.65         | 99.71         | 99.47         |

---

**Table V: Performance of the CRF model for PoS tagging with two feature sets**

| Feature set | Test Acc. | Training Acc. |
|-------------|-----------|---------------|
| n ≤ 20      | n ≤ 25    | n ≤ 20        | n ≤ 25        |
| Word identities, word shapes | 87.62 | 88.93 | 90.75 | 91.56 |
| Word identities, word shapes, word embeddings | 88.97 | 90.26 | 91.64 | 92.29 |

---

**Table VI: Performance of the LSTM model for NER with three feature sets on the test set (sentences not longer than 20 tokens)**

| Feature Set | Precision | Recall | F1  |
|-------------|-----------|--------|-----|
| (1) = Word identities, word shapes | 77.20 | 76.17 | 77.23 |
| (2) = (1) + PoS tags | 78.92 | 82.85 | 80.68 |
| (3) = (2) + word embeddings | 85.29 | 86.17 | 86.03 |

**Table VII** shows the results of the CRF model with two feature sets: 1) \{word identity, word shapes\}; and 2) \{word identity, word shapes, word embeddings\}. The table indicates that incorporating word embedding features helps to improves the accuracy of the CRF model 1.35% from 87.62% to 88.97%. The CRF model outperformed LSTMs when we do not use word embedding features. However, its accuracy is lower than that of LSTMs when word embedding features are incorporated.

In the third experiment, we enlarge the data set by considering longer sentences. We train and compare sequence models on sentences not longer than 25 tokens. With this length, there are 6,221 training sentences and 737 test sentences in the standard data set. The vocabulary now contains 2,197 different words. The LSTM tagger achieves an accuracy of 90.65% on the test set, significantly better than its performance on the 20-token data set. Similar to the LSTM models, the performances of CRFs model are better than those on shorter sentences. This can be explained by the fact that the more training data are available, the greater number of patterns the models can learn. These results also confirm the effectiveness of word embedding features for the CRF model.

We observe that performance of the CRF model is slightly worse than that of LSTMs model, both on the test sets and on the training sets. In addition, the LSTMs model has a very good memorization capacity – its accuracy on the training set is nearly perfect on long sentences, especially when the number of hidden units in use is large enough.

2) **Named Entity Recognition:** Similar to PoS tagging experiments, we evaluate NER methods on sentences of length not greater than 20 and on sentences of length not greater than 25. In the former experiment, there are 8,968 training sentences and 1,355 test sentences; the vocabulary size is 2,525. In the latter experiment, there are 11,436 training sentences and 1,787 test sentences; the vocabulary size is 3,368.

Table VI and Table VII shows the performance of the LSTM models on sentences not longer than 20 and 25 tokens, respectively. Table VIII and Table IX shows the experimental results of the CRF model on the same data sets and feature sets as those of LSTM experiments. The results indicated the effectiveness of PoS tag and word embedding features. While using PoS tag features mainly improved recall using word embedding features helped to improve both precision and
We found that the LSTMs model obtained slightly better test accuracies and had much better memorization capacity than the CRF model in NER tasks. In this paper, we compare the LSTMs model and the CRF model in two Vietnamese text processing tasks. In our understanding, our work is the first empirical work that compares these two discriminative sequence labeling models for Vietnamese text processing tasks in a systematic way. We found that the LSTMs model obtained slightly better test accuracies and had much better memorization capacity than the CRF model in both tasks. We also showed the effectiveness of word embedding features in sequence labeling models.

Because of the space limitation, we did not include maximum-entropy Markov models (MEMM) and hidden Markov models (HMM) in our comparison. It has been shown that the CRF model overcomes limitations of MEMM and HMM and outperforms MEMM and HMM in many sequence labeling tasks. For the comparison between MEMM, HMM and CRF, we can refer to the work. In particular, the work investigated and compared MEMM and CRF.

We also did not include bidirectional LSTM-CNN-CRF, the state-of-the-art end-to-end sequence labeling model, which combine bidirectional LSTM, CNN and CRF. That work used both word- and character-level representations in the neural network. Actually, in our paper, we do not aim to obtain state-of-the-art results but to compare discriminative sequence labeling models in a basic setting.

Word representations which are learned from raw text corpora, have been shown to be effective in sequence labeling models. In, Turian et al. intensively evaluated features derived from unsupervised word representations such as Brown clusters and word vectors on NER and chunk tasks with the CRF model. They used near state-of-the-art supervised baselines, and showed that word representation features improved those baselines. Our work confirmed the benefit of using word representation features for Vietnamese language processing tasks. In this paper, although we did not use word-cluster-based features, we obtained significant improvements in both two tasks.

In our work, we limit the maximal length of sentences to 25 tokens. The reason is that the LSTMs model has very high computational cost, especially for long sentences. We need about 8 hours just to train an experiment with the LSTMs model. Considering that the main purpose of the paper is to compare two sequence labeling models in experiments with simple settings, we decided to limit the maximal length of sentences. With the same reason, we decided to just use simple unigram features in the two sequence labeling models.

We found that the LSTMs model did not really outperform the CRF model in our experiments. We suspect that the training data size we used in experiments is not large, and it affected the generalization capacity of the LSTMs model. Improving generalization capacity of deep learning models on small data is still a challenging problem in the deep learning research community. In contrast, the CRF model worked quite well even with moderate-sized training data. The lesson we leaned from the results is that in an application domain that we could not obtain large data, we may use fast sequence labeling models such as CRF and spend time designing good features that are specific and beneficial for that domain.

### IV. DISCUSSION

Discriminative sequence labeling models such as CRF or LSTMs models have been used for Vietnamese text processing tasks such as PoS tagging or named-entity recognition (NER). In this work, we compare the LSTMs model and the CRF model in two Vietnamese text processing tasks. In our understanding, our work is the first empirical work that compares these two discriminative sequence labeling models for Vietnamese text processing tasks in a systematic way. We found that the LSTMs model obtained slightly better test accuracies and had much better memorization capacity than the CRF model in both tasks. We also showed the effectiveness of word embedding features in sequence labeling models.

Because of the space limitation, we did not include maximum-entropy Markov models (MEMM) and hidden Markov models (HMM) in our comparison. It has been shown that the CRF model overcomes limitations of MEMM and HMM and outperforms MEMM and HMM in many sequence labeling tasks. For the comparison between MEMM, HMM and CRF, we can refer to the work. In particular, the work investigated and compared MEMM and CRF.

We also did not include bidirectional LSTM-CNN-CRF, the state-of-the-art end-to-end sequence labeling model, which combine bidirectional LSTM, CNN and CRF. That work used both word- and character-level representations in the neural network. Actually, in our paper, we do not aim to obtain state-of-the-art results but to compare discriminative sequence labeling models in a basic setting.

Word representations which are learned from raw text corpora, have been shown to be effective in sequence labeling models. In, Turian et al. intensively evaluated features derived from unsupervised word representations such as Brown clusters and word vectors on NER and chunk tasks with the CRF model. They used near state-of-the-art supervised baselines, and showed that word representation features improved those baselines. Our work confirmed the benefit of using word representation features for Vietnamese language processing tasks. In this paper, although we did not use word-cluster-based features, we obtained significant improvements in both two tasks.

In our work, we limit the maximal length of sentences to 25 tokens. The reason is that the LSTMs model has very high computational cost, especially for long sentences. We need about 8 hours just to train an experiment with the LSTMs model. Considering that the main purpose of the paper is to compare two sequence labeling models in experiments with simple settings, we decided to limit the maximal length of sentences. With the same reason, we decided to just use simple unigram features in the two sequence labeling models.

We found that the LSTMs model did not really outperform the CRF model in our experiments. We suspect that the training data size we used in experiments is not large, and it affected the generalization capacity of the LSTMs model. Improving generalization capacity of deep learning models on small data is still a challenging problem in the deep learning research community. In contrast, the CRF model worked quite well even with moderate-sized training data. The lesson we leaned from the results is that in an application domain that we could not obtain large data, we may use fast sequence labeling models such as CRF and spend time designing good features that are specific and beneficial for that domain.

### V. RELATED WORK

This section briefly reviews related works on Vietnamese part-of-speech tagging and named entity recognition using discriminative sequence models. In, the authors give an empirical study of MEMM for Vietnamese part-of-speech tagging with different feature sets. Their best model has a tagging accuracy of about 93.5% when all the VLSP treebank is used. We see that despite using a smaller data set with short sentences and a very simple feature set with minimal hand-crafted word shapes, we are able to achieve a tagging accuracy of more than 90%. This is a strong lower bound for this task when only raw text is available for tagging.

In VLSP 2016 workshop, several different systems have been proposed for Vietnamese NER. The F-score of the best participating system is 88.78% in that shared task. That system used many hand-crafted features to improve the performance of MEMM. Most approaches in VLSP 2016 used the CRF and maximum entropy models, whose performance is

---

6 On an IBM server with 32 GB RAM and 8-core CPU.
heavily dependent on feature engineering. Table X shows those models and their performance. We observe that although the models studied in this work only rely on word features, their performance is very competitive.

Most recently, a more advanced end-to-end system for Vietnamese NER using LSTMs was proposed [28], which achieved an F1 score of 88.59%.

VI. CONCLUSION

We have presented an empirical and comparative study of two discriminative sequence prediction models CRFs and LSTMs on two fundamental tasks of Vietnamese text processing. We have demonstrated the great benefit of integrating word embeddings which are trained by an unsupervised learning method into both of the two models. These word embeddings are able to capture semantic similarities which help improve the prediction ability of the models, thereby increase the part-of-speech tagging and named entity recognition accuracy by about 4.0% and 5%, respectively. The LSTMs model is slight better than the CRFs model in terms of accuracy but the gap is not always significant in moderate-sized data sets, with the cost of much longer training time. We have also shown for the first time that a strong accuracy lower bound for both part-of-speech tagging and named entity recognition can be obtained by relying on only simple, word-based features with a minimal hand-crafted features. Using a feature set of word identities, word shapes and word embeddings, we improve the prediction ability of the models, thereby increase character-level performance for Vietnamese named entity recognition performance on sentence not longer than 25 tokens.

Table X: Performances of NER systems at VLSP 2016

| Team         | Model | Performance |
|--------------|-------|-------------|
| Le-Hong [24] | ME    | 88.78       |
| (Anonymous)  | CRF   | 86.62       |
| Nguyen et al. | LSTM | 84.08       |
| Nguyen et al. | LSTM | 83.80       |
| Le et al.    | LSTM  | 78.40       |

REFERENCES

[1] L. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” Proceedings of the IEEE, vol. 77, no. 2, pp. 257–286, 1989.
[2] L. R. Welch, “Hidden Markov models and the Baum-Welch algorithm,” IEEE Information Theory Society Newsletter, vol. 53, no. 4, 2003.
[3] A. McCallum, D. Freitag, and F. Pereira, “Maximum entropy Markov models for information extraction and segmentation,” in Proceedings of ICML, 2000.
[4] J. Lafferty, A. McCallum, and F. Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data,” in ICML, 2001, pp. 282–289.
[5] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.
[6] J. N. Darroch and D. Ratcliff, “Generalized iterative scaling for log-linear models,” Annals of Mathematical Statistics, vol. 43, no. 5, pp. 1470–1480, 1972.
[7] J. Goodman, “Sequential conditional generalized iterative scaling,” in Proceedings of ACL, 2002, pp. 9–16.
[8] J. Nocedal and S. J. Wright, Numerical Optimization, 2nd ed. New York: Springer, 2006.
[9] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.
[10] A. Graves, A.-R. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in Proceedings of ICASSP. IEEE, 2013, pp. 6645–6649.
[11] P. T. Nguyen, L. V. Xuan, T. M. H. Nguyen, V. H. Nguyen, and P. Le-Hong, “Building a large syntactically-annotated corpus of Vietnamese,” in Proceedings of ACL, Linguistic Annotation Workshop, ACL-IJCNLP, Suntec City, Singapore, 2009, pp. 182–185.
[12] P. Le-Hong, T.-M.-H. Nguyen, T.-L. Nguyen, and M.-L. Hu, “Fast dependency parsing using distributed word representations,” in Trends and Applications in Knowledge Discovery and Data Mining, ser. Lecture Notes in Artificial Intelligence. Springer, 2015, vol. 9441.
[13] C. Vu-Manh, A.-T. Luong, and P. Le-Hong, “Improving Vietnamese dependency parsing using distributed word representations,” in Proceedings of SoICT. ACM, 2015, pp. 54–60.
[14] P. Le-Hong, T. M. H. Nguyen, A. Roussanaly, and T. V. Ho, “A hybrid approach to word segmentation of Vietnamese texts,” in Language and Automata Theory and Applications, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2008, pp. 240–249.
[15] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in Advances in Neural Information Processing Systems 26, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 3111–3119.
[16] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” in Proceedings of Workshop at ICLR, Scottsdale, Arizona, USA, 2013.
[17] N. Okazaki, “Crfsuite: a fast implementation of conditional random fields (crfs),” 2007. [Online]. Available: http://www.chokkan.org/software/crfsuite/
[18] J. Nocedal, “Updating quasi-newton matrices with limited storage,” Mathematics of computation, vol. 35, no. 151, pp. 773–782, 1980.
[19] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, vol. 9, Sardinia, Italy, 2010, pp. 249–256.
[20] P. Le-Hong, X.-H. Phan, and T. T. Tran, “On the effect of the label bias problem in part-of-speech tagging,” in The 10th IEEE RIVF. Hanoi, Vietnam: IEEE, 2013, pp. 103–108.
[21] X. Ma and E. Hovy, “End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF,” in Proceedings of the ACL, Berlin, Germany, August 2016, pp. 1064–1074.
[22] J. Turiyan, L. Ratinov, and Y. Bengio, “Word representations: A simple and general method for semi-supervised learning,” in Proceedings of ACL, Uppsala, Sweden, 2010, pp. 384–394.
[23] P. Le-Hong, A. Roussanaly, T. M. H. Nguyen, and M. Rossignol, “An empirical study of maximum entropy approach for part-of-speech tagging of Vietnamese texts,” in Actes de Traitement Automatique des Langues, Montreal, Canada, 2010, pp. 50–61.
[24] P. Le-Hong, “Vietnamese named entity recognition using token regular expressions and bidirectional inference,” in Proceedings of VLSP, Hanoi, Vietnam, 2016.
[25] T. C. V. Nguyen, T. S. Pham, T. H. Vuong, N. V. Nguyen, and M. V. Tran, “DSKTLAB-NER: Nested named entity recognition in Vietnamese text,” in Proceedings of VLSP, Hanoi, Vietnam, 2016.
[26] T. S. Nguyen, L. M. Nguyen, and X. C. Tran, “Vietnamese named entity recognition at VLSP 2016 evaluation campaign,” in Proceedings of VLSP, Hanoi, Vietnam, 2016.
[27] T. H. Le, T. T. Nguyen, T. H. Do, and X. T. Nguyen, “Named entity recognition in Vietnamese text,” in Proceedings of VLSP, Hanoi, Vietnam, 2016.
[28] T.-H. Pham and P. Le-Hong, “End-to-end recurrent neural network models for Vietnamese named entity recognition: Word-level vs. character-level,” in Proceedings of PACLING, Yangon, Myanmar, 2017.