Myanmar Language Part-of-Speech Tagging Using Deep Learning Models

Thinn Thinn Wai

Abstract — Part-of-speech (POS) tagging is one of the most important processes in Natural Language Processing (NLP). It is useful in many areas of linguist research such as information retrieval, natural language translation, word sense disambiguation and sentiment analysis. The goal of this process is to correctly assign the POS tags for each word in a sentence. Moreover, it is also the essential process for Myanmar Language Translation. Although there are many approaches in Myanmar Language POS tagging, Deep Learning Models are especially proposed in this paper. In this work, Recurrent Neural Network (RNN) with Bi-directional Long Short-Term Memory (Bi-LSTM RNN) is especially applied in Myanmar Word segmentation and POS tagging process. Moreover, GloVe is also used to perform syllable vector representation and word vector representation.

Index Terms — Part-of-Speech tagging, Natural Language Processing, Word Segmentation, Recurrent Neural Network, Bi-directional Long Short-Term Memory, GloVE

1 INTRODUCTION

Recently, deep learning has been gaining popularity in the field of Artificial Intelligence and Natural Language Processing. The advancements are due to the breakthrough in the algorithms that learn and recognize very complex patterns using deep layers of neural networks or commonly known as the deep neural networks (DNN) [1], and the introduction of different types of neural network such as convolutional neural network and recurrent neural network (RNN). Although there are many well-known projects which use Deep Learning in English language translation having human level performance, applying deep learning on Myanmar Language translation is still rare because of the language structure and word composition. Moreover, there are many preprocessing steps needed to carry out to translate Myanmar language to another language. Among these process, part-of-speech tagging is one of the essential processes in Myanmar language translation. In addition to this, there is no space between words and no certain rule for spacing in writing style of Myanmar Language. And so, part-of-speech tagging does not perform directly on Myanmar sentences and syllabification and word segmentation are needed to carry out in advance. In this study, GloVe (global vector for word representation) is used to perform syllable level vector representation and word level vector representation. Moreover, two recurrent neural networks with long short-term memory (LSTM) are used for word segmentation and part-of-speech tagging.

2 PROPOSED SYSTEM ARCHITECTURE

The architecture of proposed RNN-based Myanmar Language POS tagger is shown in Fig 1. As shown in this figure, when input Myanmar text is enter to the system, preprocessing steps such as sentence breaking, and syllable breaking are needed to carry out. For sentence breaking, input texts are broken as line by line sentences by using sentence marker symbol (‘။’) called ‘pope ma’. After sentence breaking, syllable breaking is performed using regular expression rules generated for each Myanmar syllable. Then, syllable-level embedding, and word-level embedding is carried out by using the corpus with the help of GloVe algorithm. After that, segmentation process is performed by using Bi-LSTM recurrent neural network (Bi-LSTM RNN) model which can predict upcoming space. Finally, POS tagger is also built by using Bi-LSTM RNN to predict the tags of words in the input sentence. In this work, public corpus, myPOS corpus created by Y.K. Thu and his et al [9] is used as the corpus resource.
3 SYLLABLE LEVEL AND WORD LEVEL EMBEDDING USING GLOVE

Like all other neural networks, deep-learning models don’t take as input raw text: they only work with numeric tensors. Vectorizing text is the process of transforming text into numeric tensors. This can be performed by tokenizing words or characters of the text. This tokenized words or characters are encoded to embed into a vector. This is also called token embedding or word embedding. Based on the concept of word embedding and character embedding, syllable level embedding is carried out for Myanmar Language.

According to the Myanmar Language’s characteristics, not only one Myanmar syllables such as (e.g., 'ဗီ [ray'] and (e.g., 'ဗုး [aoe]', pot in English) but also more than one syllable are combined to form a meaningful word such as ‘ဗီဗုး [rayaoe], water pot in English’.

Moreover, Myanmar Syllable (‘သည် [sany]’) have more than one meaning. If this syllable is not combined with another Myanmar syllables, it has no meaning and it is used as post position marker of Myanmar sentences. If this syllable is combine with other syllables such as (မုန့် [mone], snack in English) and (ဗုးဗုး [hcyaayy], market in English) these two syllables combination is transform to another meaning (မုန့်သည် [mone sany], snack seller in English) and (ဗုးသည် [hcyaayy sany], seller in English) and it can be tagged as noun in Myanmar POS tagging.

Because of the multiple meanings and multiple usages of each syllable, only syllable vector usage is not efficient to get the meaningful segmented words and correct POS tags. Therefore, Word vector is also built using GloVe in order to get the meaningful segmentation and this segmentation result is used as the input for POS tagger. In this work, 13779 words from corpus are embedded into word vectors. Moreover, examples of Myanmar words with one syllable and two syllables are shown in Table 1.

Table 1: Myanmar Word’s Phonetics and English Meanings

<table>
<thead>
<tr>
<th>Myanmar Words</th>
<th>Phonetics</th>
<th>English Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ဗီ (one syllable)</td>
<td>[ray]</td>
<td>water</td>
</tr>
<tr>
<td>ဗုး (one syllable)</td>
<td>[aoe]</td>
<td>pot</td>
</tr>
<tr>
<td>ဗီဗုး (two syllables)</td>
<td>[rayaoe]</td>
<td>water pot</td>
</tr>
<tr>
<td>သည် (one syllable)</td>
<td>[sany]</td>
<td>bamboo</td>
</tr>
<tr>
<td>ဗုးသည် (two syllables)</td>
<td>[hcyaayy sany]</td>
<td>seller</td>
</tr>
</tbody>
</table>

3.1 GloVe Model

For syllable embedding and word embedding, GloVe model is trained on the non-zero entries of a global syllable-syllable or word-word co-occurrence matrix, which tabulates how frequently syllable or words co-occur with one another in a given corpus. Populating this matrix requires a single pass through the entire corpus to collect the statistics. Glove Algorithm consists of the following steps.

Step 1: Collect syllable/word co-occurrence statistics in a form of syllable/word co-occurrence matrix X. (Each element Xij of such matrix represents how often ith syllable/word appears in context of jth syllable/word)

Step 2: Define soft constraints for each syllable/word pair by using the equation

\[ W_i^T W_j + b_i + b_j = \log(X_{ij}) \]

Where, \( W_i \) - vector for the main syllable/word
\( W_j \) - vector for the context syllable/word
\( b_i, b_j \) are scalar biases for the main and context syllable/words

Step 3: Define cost function :
\[ J = \sum_{i=1}^{n} \sum_{j=1}^{n} f(x_{ij})(W_{ij} + b_{i} + b_{j} - \log x_{ij})^2 \]

Where, \( f \) is weighing function and \( f(x_{ij}) \) is defined by

\[ f(x_{ij}) = \begin{cases} \frac{x_{ij}}{x_{\text{max}}} & \text{if } (x_{ij}) > x_{\text{max}} \\ 1 & \text{otherwise} \end{cases} \]

In this work, context window size of 3 is used in GloVe because there are many particles in Myanmar language and large value of context window size in co-occurrence will lead to high similarity between syllables.

4 SEGMENTATION AND POS TAGGING USING BI-LSTM RNN MODEL

In this work, bi-direction LSTM RNN is used in two processes. The first one is word segmentation process and the second one is part-of-speech tagging process. The architecture of Bi-LSTM RNN model for word segmentation and POS tagging is illustrated in Fig 2.

\[ \text{Where, } h_t = H(W_{\text{ph}} x_t + W_{\text{hh}} h_{t-1} + b_{\text{h}}) \]
\[ y_t = W_{\text{hy}} h_t + b_{\text{y}} \]

Where, \( h_t \) is the vector of hidden states
\( W \) is weight matrix
\( W_{\text{ph}} \) is weight between input and hidden layer
\( b_{\text{h}} \) is bias of hidden layer

LSTM network is formed like the standard RNN except that the self-connected hidden units are replaced by special designed units called memory blocks. In order to store the information in the cell of LSTM memory block over long periods of time and avoid the vanishing gradient, multiple gates such as, input gate \( (i_t) \), forget gate \( (f_t) \), output gate \( (o_t) \) and cell activation vectors \( c_t \) are worked out according to the following equations.

\[ i_t = \sigma(W_{ix} x_t + W_{hi} h_{t-1} + W_{ri} c_{t-1} + b_i) \]
\[ f_t = \sigma(W_{fx} x_t + W_{hf} h_{t-1} + W_{rf} c_{t-1} + b_f) \]
\[ c_t = f_t c_{t-1} + i_t \tanh(W_{xc} x_t + W_{hc} h_{t-1} + b_c) \]
\[ o_t = \sigma(W_{ox} x_t + W_{ho} h_{t-1} + W_{ro} c_t + b_o) \]

Where \( \sigma \) is the logistic sigmoid function and the output of LSTM hidden layer \( h_t \) given input \( x_t \) is computed as following composite functions:

\[ h_t = o_t \tanh(c_t) \]

Moreover, the illustration of the layers of Bi-directional RNN are described in Fig 3.

4.1 Bi-directional LSTM RNN Model (Bi-LSTM RNN)

Recurrent neural network (RNN) is a type of artificial neural network that consists of cyclic connections to model contextual information dynamically. If an input word sequence \( w_1, w_2, \ldots, w_n \) is given, a standard RNN computes the output vector \( y_t \) of each word \( w_t \) by iterating the following two equations from \( t=1 \) to \( n \).

\[ y_t = W_{hy} h_t + b_{y} \]

As shown in fig 3, BRNN involves two separate hidden layers, forward layer and backward layer that can access both preceding and succeeding contexts to offers an effective solution. BRNN first computes the forward hidden sequence \( h_t \) by using the equation, \( h_t = H(W_{hph} x_t + W_{hph} h_{t-1} + b_{h}) \) and the
backward hidden sequence $h_t$ by using the equation,
$$h_t = H(W_{r_t}x_t + W_{r_t}h_{t-1} + b)$$
respectively, and then pass the value of $h_t$ and $h_{t-1}$ to the fully connected layer ($f_c$) and ($f_t$) compute the truth output from these layers and the result is passed to the logic layer that used the sigmoid function. Finally, the result obtained from sigmoid function is normalized by using cost function of the SoftMax layer and the desired segmented words or POS tags are produced. In ($f_t$) layer, the two outputs are computed by using the equation (1) and the logit layer used the equation (2).

$$f_t = \text{relu} (W_{f_t}h_t + b_{f_t}) \quad \text{(1)}$$
$$\text{logit} = \text{relu} (W_{f_t}f_t + b_t) \quad \text{(2)}$$
$$\text{output} = \text{softmax} (\text{logit}) \quad \text{(3)}$$

For segmentation, cost function of the softmax layer is defined by using the following equation.

$$\text{Cost} = \text{classweight} \times \text{SoftmaxCrossEntropy} (\text{Output}, \text{Target})$$

5 EXPERIMENTS

Proposed POS tagger can classify the words in 14 categories such as Abbreviation, Adjective, Adverb, Conjunction, Foreign Word, Interjection, Noun, Number, Particle, Postpositional Marker, Pronoun, Punctuation, Text number, Verb defined by myPOS copus created by Y. K. Thu [9]. These 14 POS tags used in this work with their class occurrence within the training corpus is described in Table 2. Proposed POS tagger is tested on open test-set and closed test-set. Open test set contains 4 K sentences and tagger achieved 98.02 % accuracy on open test-set. Compared with other POS tagging approaches, the accuracy of proposed tagger is illustrated in Fig 4.

Table 2: POS Tags Used in Proposed System

<table>
<thead>
<tr>
<th>POS Tags</th>
<th>Definition</th>
<th>Occurrence in Corpus</th>
</tr>
</thead>
<tbody>
<tr>
<td>abb</td>
<td>Abbreviation</td>
<td>311</td>
</tr>
<tr>
<td>Adj</td>
<td>Adjective</td>
<td>7200</td>
</tr>
<tr>
<td>Adv</td>
<td>Adverb</td>
<td>2886</td>
</tr>
<tr>
<td>Conj</td>
<td>Conjunction</td>
<td>11696</td>
</tr>
<tr>
<td>Fw</td>
<td>Foreign Word</td>
<td>2331</td>
</tr>
<tr>
<td>Int</td>
<td>Interjection</td>
<td>98</td>
</tr>
<tr>
<td>N</td>
<td>Noun</td>
<td>65237</td>
</tr>
<tr>
<td>Num</td>
<td>Number</td>
<td>3940</td>
</tr>
<tr>
<td>Part</td>
<td>Particle</td>
<td>52394</td>
</tr>
<tr>
<td>tn</td>
<td>Text Number</td>
<td>2351</td>
</tr>
<tr>
<td>Ppn</td>
<td>Post Positional Marker</td>
<td>38765</td>
</tr>
<tr>
<td>Pron</td>
<td>Pronoun</td>
<td>2684</td>
</tr>
<tr>
<td>Punc</td>
<td>Punctuation</td>
<td>15840</td>
</tr>
<tr>
<td>V</td>
<td>Verb</td>
<td>33628</td>
</tr>
</tbody>
</table>

Fig 4: Accuracy of Proposed System
6 CONCLUSION

In the proposed system, POS tagging is performed by using deep learning model, Bi-LSTM-NN model. Although the proposed POS tagger can outperform over some traditional POS tagging approaches, there are still limitations compare to human tagger. To overcome these limitations, more training will be carried out in the future. Moreover, POS tagging process is the one of the necessary processes for Natural Language Processing. Therefore, POS tagged resulted from the proposed system can be used as the useful inputs of the Natural language Processing, Machine Translation and Information retrieval system.

REFERENCES


IJSER © 2019
http://www.ijser.org