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.

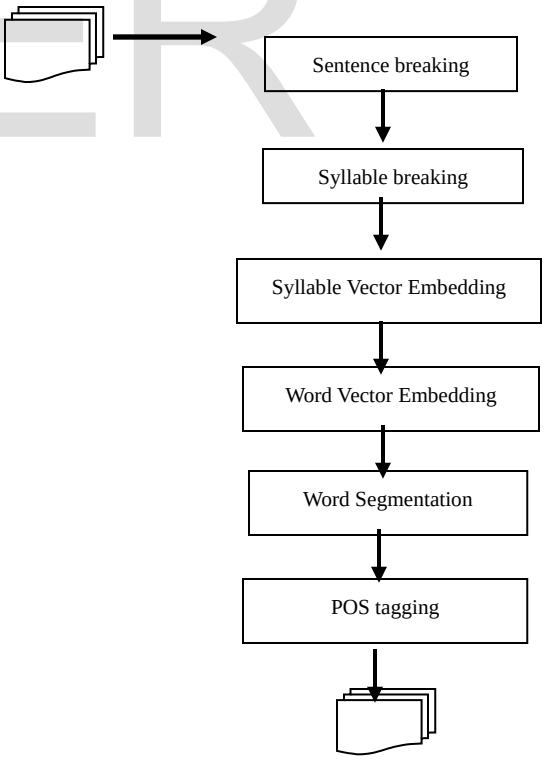

Fig 1: Proposed System Architecture

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, '**ရေ** , *[ray']*) and (e.g, '**အိုး** , *[aoe]'*) forms a meaningful Myanmar words, ('**ရေ** , *[ray]'*, water in English) and ('**အိုး** , *[aoe]'*, pot in English) bust also more than one syllable are combined to form a meaningful word such as '**ရေအိုး** , *[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 (**ဈေး** , *[hcyaayy]*, market in English) these two syllables combination is transform to another meaning (**မုန့်သည်** , *[mone snay]*, snack seller in English) and (**ေဈးသည်** , *[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.

Myanmar Words	Phonetics	English Meaning
ရေ (one syllable)	*(ray)*	water
အိုး (one syllable)	*(aoe)*	pot
ရေအိုး (two syllables)	*(rayaoe)*	water pot
ဝါး (one syllable)	*(warr)*	bamboo
အိပ်ရာ (two syllables)	*(aiutrar)*	bed
သည် (one syllable)	*(sany)*	(None)
ဈေးသည်(two syllables)	*(hcyaayysany)*	seller
မုန့် သည် (two syllables)	*(monesany)*	Snack seller

Table 1: : Myanmar Word's Phonetics and English Meanings

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-occurence statistics in a form of syllable/word co-ocurrence matrix X. (Each element X_{ij} of such matrix represents how often i^{th} syllable/word appears in context of j^{th} 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 :

$$J = \sum_{i=1}^{V}\sum_{j=1}^{V} f(X_{ij})(W_i^T W_j + b_i + b_j - logX_{ij})^2$$

Where, f is weighing function and $f(X_{ij})$ is defined by

$$f(X_{ij}) = \begin{cases} (\frac{X_{ij}}{x_{max}})^{\alpha} & if(X_{ij}) > Xmax \\ 1 & otherwise \end{cases}$$

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-ocurrencnce 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.

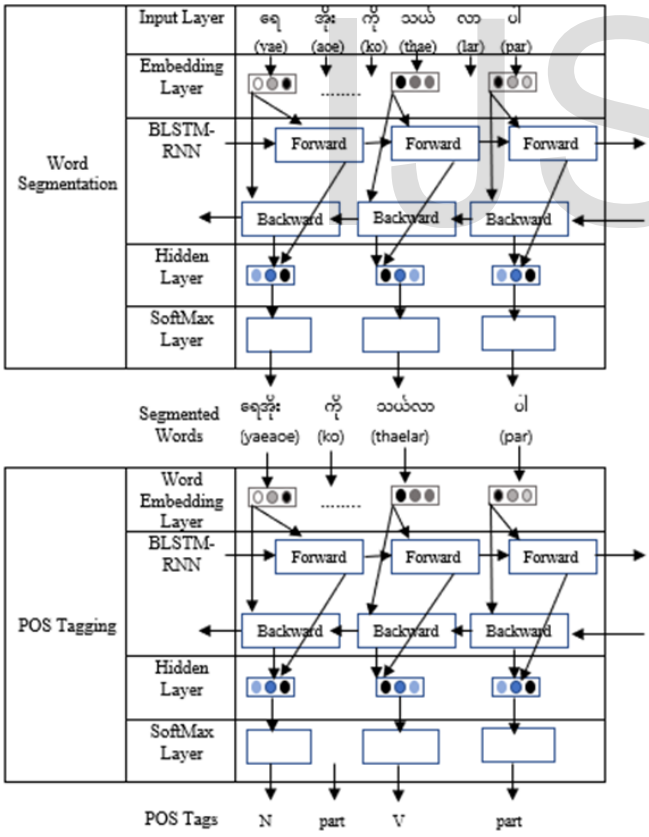

Fig 2: Segmentation and POS tagging Using Bi-LSTM-RNN

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, ..., w_n is given, a standard RNN computes the output vector y_t of each word wt by iterating the following two equations from t=1 to n .

$$h_t = H(W_{xh}x_t + W_{hh}h_{t-1} + b_h)$$
$$y_t = W_{hy}h_t + b_y$$

Where, h_t is the vector of hidden states
W is weight matrix
W_{xh} is weight between input and hidden layer
b_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.

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i)$$
$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f)$$
$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$$
$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o)$$

Where σ is the logistic sigmoid function and the output of LSTM hidden layer ht given input x_t is computed as following composite functions:

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

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

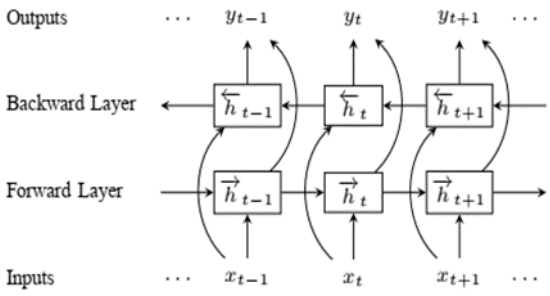

Fig 3: Layeres of Bi-directional RNN

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 $\vec{h}_t$ by using the equation, $\vec{h}_t = H(W_{x\vec{h}}x_t + W_{\vec{h}\vec{h}}\vec{h}_{t-1} + b_{\vec{h}})$ and the

backward hidden sequence $\overleftarrow{h_t}$ by using the equation, $\overleftarrow{h_t} = H(W_{x\overleftarrow{h}}x_t + W_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h}_{t+1} + b_{\overleftarrow{h}})$ respectively, and then pass the value of $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ to the fully connected layer (f_c) and (f_c) 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_c) layer, the two outputs are computed by using the equation (1) and the logit layer used the equation (2).

$$f_c = relu\ (W_{fc}[\overrightarrow{h_t} + \overleftarrow{h_t}] + b_{fc}) \quad \text{---(1)}$$

$$logit = relu\ (W_l f_c + b_l) \quad \text{---(2)}$$

$$output = softmax\ (logit) \quad \text{---(3)}$$

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

$$targets * [-\log(sigmoid\ (logit)) * P_{weight}] + (1 - targets) * [-\log(1 - sigmoid(logits))]$$

where, P_{weight} is the weight to control the amount of false positive and false negative. False negative count would be decreased if $P_{weight} < 1$ and false positive count would be decreased if $P_{weight} > 1$. As the proposed system emphasize on precision, the value of 0.5 is set to P_{weight}. Optimization to cost function is done by Adam Optimizer with learning rate 0.0001.

Although the basic architecture of POS tagger is the same as the segmentor (bidirectional LSTM with peephole connection), cost function of segmentor is only intended to binary classification. And therefore, cost function of POS tagger is different from segmentor. And Softmax cross entropy is used as for softmax layer of the POS tagger. Moreover, according to the corpus statistics, learning process become difficult due to the unfair distribution of each class (for example, proportion to Noun by Interjection is around 666). Therefore, class weight is proposed for smoothing unfair distribution problem and cost function is smoothed by multiplying class weight to original softmax cross entropy as shown in the following equation.

$$Cost = classweight * SoftmaxCrossEntropy(Output, Target)$$

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.

POS Tags	Definition	Occurrence in Corpus
abb	Abbreviation	311
Adj	Adjective	7200
Adv	Adverb	2886
Conj	Conjunction	11696
Fw	Foreign Word	2531
Int	Interjection	98
N	Noun	65237
Num	Number	3940
Part	Particle	52394
tn	Text Number	2351
Ppm	Post Positional Marker	38765
Pron	Pronoun	2684
Punc	Punctuation	15840
V	Verb	33628

Table 2: POS Tags Used in Proposed System

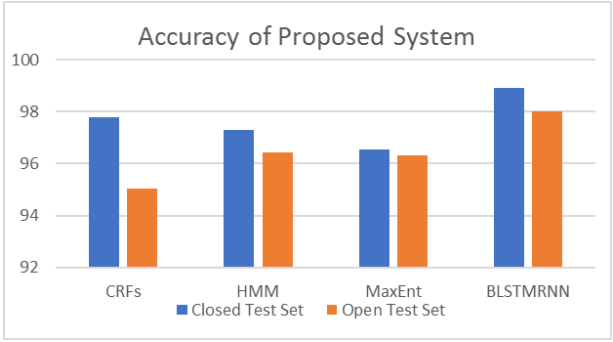

Fig 4: Accuracy of Proposed System

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

[1] H. Tang , "Bidirectional LSTM-CNNs-CRF Models for POS Tagging", 2018.
[2] T.P. Tan, B.R. Malançon , L. Besacier, Y.L. Yeong, K. H. Gan, and E. K. Tang "Evaluating LSTM Networks, HMM and WFST in Malay Part-of-Speech Tagging" , School of Computer Sciences, Universiti Sains Malaysia, Penang, Malaysia.
[3] Dmitriy Selivanov, "GloVe Word Embeddings", 2017.
[4] M. Nielsen, "Neural networks and deep learning," Online: http://neuralnetworksanddeeplearning.com/, 2017.
[5] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in Proc. INTERSPEECH, Singapore, 2014, pp.338-342.
[6] Y.K.Thu, A. Tamuray, A.Finchy, E.Sumitay, Y.Sagisaka "Unsupervised POS Tagging of Low Resource Language for Machine Translation", 2014.
[7] R.N. Patel, P.B. Pimpale, M. Sasikumar " Recurrent Neural Network based Part-of-Speech Tagger for Code-Mixed Social Media Text" , 2016.
[8] Natural Language Processing with Deep Learning. Online: http://web.stanford.edu/class/cs224n/.
[9] K.W.W.Htike, Y.K.Thu, W.P.Pa, "Comparison of Six POS Tagging Methods on10K Sentences Myanmar Language (Burmese) POS Tagged Corpus."
[10] Myanmar Grammar, Department of the Myanmar Language Commission. 2005. Myanmar Grammar. Ministry of Education, Myanmar.
[11] Z.Sun, Z.H.Deng "Unsupervised Neural Word Segmentation for Chinese via Segmental Language Modeling".