# Comparative Analysis of Deep Learning Models for Myanmar Text Classification

Myat Sapal Phyu(✉) and Khin Thandar Nwet(✉)

Faculty of Computer Science, University of Information Technology, Yangon, Myanmar
{myatsapalphyu,khinthandarnwet}@uit.edu.mm

**Abstract.** Text classification is one of the major research areas for Natural Language Processing (NLP). Long Short Term Memory (LSTM), Convolutional Neural Networks (CNN), and their combination models have been applied in many NLP tasks. This paper presents a joint CNN with no max-polling layer and Bidirectional LSTM to fulfill the requirements of each model. The proposed model takes advantage of CNN to extract features and Bi-LSTM to capture long term contextual information from past and future contexts. The proposed model is compared with CNN, Bi-LSTM, RNN, and CNN-LSTM models with pre-trained word embedding on five article datasets in Myanmar language.

**Keywords:** Text classification · Myanmar language · Deep learning · Pre-trained word embedding · CNN · RNN · CNN-RNN · CNN-LSTM · Bi-LSTM

## 1 Introduction

In the age of information, people are wasting a lot of time finding their interesting information. Consequently, it is crucial to effectively and quickly extract the most relevant information from a wide range of information. Text classification can negotiate with these problems. It is one of Natural Language Processing (NLP)'s main research areas. Text classification is the arrangement of text into their respective categories such as spam filtering, articles, sentiment analysis, posts, and hate speech identification. Recently, the use of word embedding with a deep learning method has attracted considerable interest in text classification due to their ability to capture semantic relationships of words [2, 6, 8]. Words are considered as basic unit in most of the NLP for implementing continuous word vector representation. This paper focuses in particular on the Myanmar text classification. There is no rule to determine word boundaries for Myanmar language. Since Myanmar language is rich in morphology, it is difficult to learn good representation of words because many word types seldom occur in the training corpus. In order to classify Myanmar text by means of deep learning models, several steps are taken to pre-process Myanmar text such as extracting massive amounts of Myanmar text, removing unnecessary characters, determining words boundaries and converting words into word vectors that keep the context information. Grave et al. [3] published pre-trained word vectors for two hundred forty-six languages trained on common crawl and Wikipedia. They proposed bag-of-character n-grams based on skip-gram that could capture sub-word


N. T. Nguyen et al. (Eds.): ACIIDS 2020, LNAI 12033, pp. 76–85, 2020.
https://doi.org/10.1007/978-3-030-41964-6_7

information to enrich word vectors. The pre-trained sub-word vectors for two hundred seventy-five languages were also released by Heinzerling et al. [5]. Their works are very helpful in resource-scarce languages and can be applied to specific NLP tasks by transferring learning. This paper applies the deep learning models for text classification and pre-trained word embedding trained on Wikipedia for the construction of embedding matrix.

The next sections are as follow, Section 2 addresses the related work of the text classification for both the English and Myanmar languages. Section 3 discusses the pre-processing steps before an embedding layer. Section 4 explains the proposed model. Section 5 explains the experimental section containing the dataset collection, comparison models and experiment results and the paper is concluded in the Sect. 6.

## 2 Related Work

Conneau et al. [2] have proposed very deep convolutional neural networks (VDCNN) that use twenty nine layers of convolution. VDCNN operated directly on character-level and performance is measured by using eight datasets. Joulin et al. [6] developed a text classification system that is efficient and simple and is denoted as fastText. This model's accuracy is similar to other deep learning classifiers, but using a regular multicore CPU, it takes less than ten minutes for training more than one billion words. Song et al. [13] introduced a context-LSTM-CNN model to use LSTM-based long-range dependencies and used the convolution layer and max-pooling layer to extract local features at specific points. Lai et al. [10] applied bi-directional RNN to capture meaning and max-pooling to capture key components in texts. Kim [8] showed that the use of a single convolution layer in the simple CNN and proposed variations of the CNN models CNN-rand, CNN-static, CNN-non-static, and CNN-multichannel. These models were experimented on seven publicly available datasets and improved the state-of-the-art methods on four out of seven datasets. Zhang et al. [15] compared character-level convolutional networks with word-level ConvNets and RNN for text classification in the English language. In Myanmar language text classification, we also investigated previous research work, such as news classification, spam filtering, and sentiment analysis. Aye et al. [1] improved the accuracy of prediction on informal Myanmar text by considering objective and intensifier words for Myanmar's food and restaurant text reviews. Khine et al. [7] showed the comparison of Naïve Bayes and k-Nearest Neighbors (KNN) algorithms for Myanmar news classification. The experiment showed that KNN is higher in recall and accuracy than Naïve Bayes on 1,200 documents datasets with four categories. Yu et al. [14] developed a corpus annotated with sentiment polarity for Myanmar news. The N-gram model is used to choose features and the Naïve Bayes algorithm is to identify emotions. Kyaw et al. [9] constructed a spam filtering corpus and proposed a Naïve Bayes-based learning algorithm for spam or harm classification. According to the literature review, some deep learning models were improved and explored for Automatic Speech Recognition as in [12]. Most of the Myanmar text classification tasks are performed in lexicon-based and approaches because the challenge of text classification in Myanmar language is the need for huge resources to train in deep learning models. Using pre-trained word vectors can address such resource-requiring problems. In our previous work [12], we performed

the comparative analysis of CNN and RNN both on syllable and word level by using three pre-trained vectors and also collected and annotate six Myanmar articles datasets. We use the pre-trained vector that is trained on the skip-gram model in the embedding layer. This paper presents a joint CNN and Bi-LSTM model and compares with most of the baseline deep learning models and their combination models for Myanmar text classification on five datasets.

## 3 Pre-processing

Pre-processing steps is cruel for Myanmar language because of its nature. Firstly, we extract sentences from text documents. Pre-processing steps contain removing the non-Myanmar character, punctuation marks, and numbers. As this work focus on Myanmar text classification, we remove non-Myanmar characters that do not contain in the Unicode range between [U1000-U104F]. The numbers [U1040-U1049] and the punctuation marks [U104A-U104B] are also removed. Myanmar language has rule to determine the boundary of words. In this work, the BPE tokenizer[1] is used to define the word boundary. Algorithm 1 and 2 show the step by step procedure of preprocessing task. Algorithm 1 shows the step-by-step process to remove the unnecessary characters from the text dataset. Table 1 shows the sample of pre-processing steps for sample input text " ဆေးဝါးလိုအပ်သူများအတွက် အခမဲ့Medicine Box ပဲခူးဆေးရုံကြီး၌ စတင်ထားရှိ။". In this sample text, non-Myanmar characters "Medicine Box" and punctuation marks " ။" are removed and the remaining text string " ဆေးဝါးလိုအပ်သူများအတွက် အခမဲ့ ပဲခူးဆေးရုံကြီး၌စတင်ထားရှိ" is segmented as " ဆေးဝါး_လိုအပ်_သူများအတွက်_အခမဲ့_ပဲခူးဆေးရုံကြီး၌_စတင်_ထားရှိ" by the tokenizer.

**Table 1.** Pre-processing steps for sample input text

| | |
|---|---|
| Input Text | ဆေးဝါးလိုအပ်သူများအတွက် အခမဲ့ Medicine Box ပဲခူးဆေးရုံကြီး၌ စတင်ထားရှိ။ |
| Word Segmentation | ဆေးဝါး_လိုအပ်_သူများအတွက်_အခမဲ့_ပဲခူး ဆေးရုံကြီး၌_စတင်_ထားရှိ |

[1] https://github.com/bheinzerling/bpemb.

**Algorithm 1:** Removing Unnecessary Characters

**Input** : Raw text document
**Result** : Text documents D without unnecessary characters
**Initialization** : Character $c_i$, Character code $cc_i$,
Myanmar Unicode, MU = [u1000-u104f],
Myanmar Digit, MD = [u1040-u104b],
Punctuation Marks, PM = [u104a-u104b], Text String T extracted from D; i = {0,1,...,n}, i = 0

```
foreach c_i ∈ T do
    if cc_i ∉ MU then
        remove c_i
        i++
    end
    if cc_i ∈ MD then
        remove c_i
        i++
    end
    if cc_i ∈ PM then
        remove c_i
        i++
    end
end
```

### 3.1 Pre-trained Vector

In this work, we use the pre-trained vector trained on the fastText Skip-gram model[2]. The number of word vectors in this pre-trained vector is 91,497 and the dimension is 300. The pre-trained vectors file is used as vocabulary to convert words into word vectors. Algorithm 2 shows the conversion of segmented words to embedding matrix. Figure 1 shows the step-by-step process before the embedding layer. Table 2 the sample result of embedding matrix for each segmented word.

**Table 2.** Sample result of embedding matrix for each segmented word

| Segmented Word | Sample Embedding Matrix (300 Dimension) |
|---|---|
| ဆေးဝါး | 0.963450 1.260167 -0.309332 0.324999 ……..…0.454490 |
| လိုအပ် | -0.303027 0.528527 -0.898522 -0.326885 ……0.060485 |
| သူများအတွက် | -1.601050 -0.07043 0.098535 -0.363543 …….. 0.178379 |
| အခမဲ့ | -0.720556 0.595658 -0.711430 0.183147 ……0.5865110 |
| ပဲခူး | 0.800076 0.450307 -0.803149 -0.686676 …….-0.876465 |
| ဆေးရုံကြီး၌ | -1.635552 -0.716141 -0.154168 -0.289999 ……0.643253 |
| စတင် | 0.058164 0.033736 -0.705663 0.482859…….....0.5694375 |
| ထားရှိ | -0.803120 -0.225860 1.441452 1.634341…….. 0.587558 |

[2] https://fasttext.cc/docs/en/pretrained-vectors.html.

**Algorithm 2:** Word Embedding Matrix

**Input** : text documents D without unnecessary characters
**Result** : Word embedding matrix for embedding layer
**Initialization** : Words $w_i$,
where, i = {0,1,...,n}
Word embedding matrix, embmatrix[i],
Text String T extracted from D, Vocabulary in
Pretrained Vectors, V,
i = 0,

```
segment T into wi by BPE tokenizer
  remove duplicate wi;
  foreach w_i ∈ V do
    if w_i ∈ V then
      embmatrix[i] = vector(w_i)
      i++
    end
  end
```

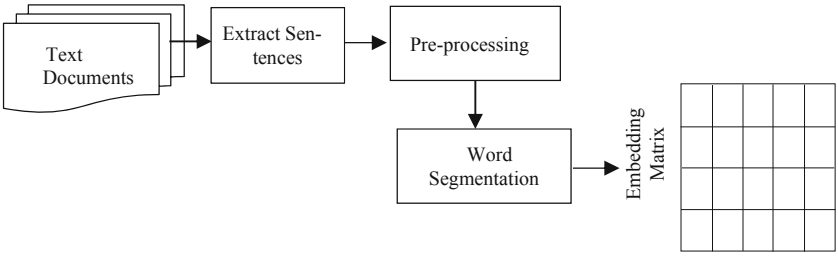


**Fig. 1.** Pre-processing steps before the embedding layer

## 4 Model

A joint CNN-Bi-LSTM model is illustrated in Fig. 2. It is basically composed of the following layers.

**Embedding Layer:** After pre-processing steps, the segmented words are matched with the vocabulary in the pre-trained word vector that is trained on the skip-gram model. Each word in the vocabulary attaches with their corresponding vectors and it can catch context information.

**Convolution Layer:** Convolution layer performs the convolution process with stride size 1 by using the ReLU activation function $f(x) = \max(x, 0)$. The convolution layer is used to extract features from the embedding matrix and discard the pooling layer because it only captures the most important information and lost the context information.

**Bi-LSTM Layer:** Bi-LSTM layer is applied as an alternative of pooling layer to capture long term semantic information from both past and future contexts.

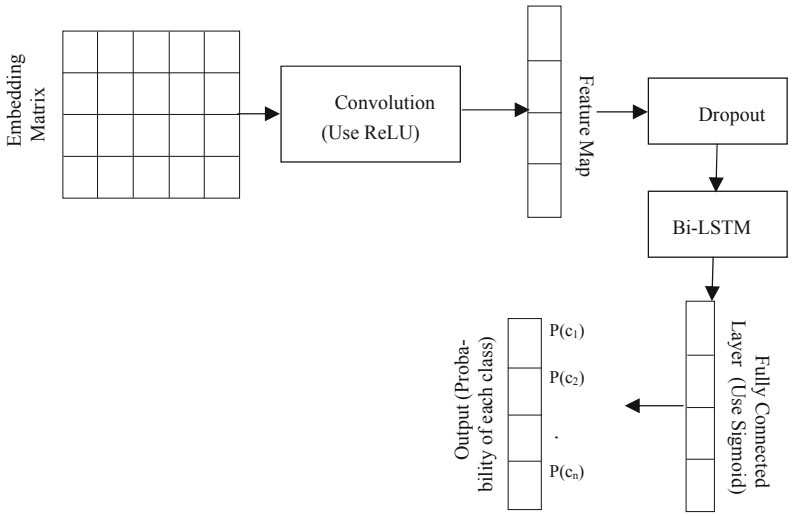


**Fig. 2.** A joint CNN and Bi-LSTM model

**Fully Connected Layer:** Fully connected layer used sigmoid activation function to calculate the probabilities of each class. The sigmoid function of $p(c_n)$ is

$$\mathrm{p}(c_n) = \frac{1}{1 + e^{-c_n}} \tag{1}$$

The probability of a class does not depend on all other classes' probabilities. It can handle the multi-label problem. Binary cross-entropy is used as a loss function and Adam optimization function with 0.5 dropouts and 16 batch size on 10 epochs are set as hyper-parameters. In addition, the bias and kernel regularizer set ($l2 = 0.01$) in output layer for reducing overfitting problem.

## 5 Experiment

### 5.1 Datasets

Empirical exploration is conducted in Myanmar language on five news datasets. These datasets are collected from five daily news websites [12]. Text data are converted into Unicode font by using rabbit converter[3]. Each line represented a sentence annotated with corresponding label. Text data are and shuffled and split into 75% and 25% for training and testing datasets. The first dataset is collected from the 7Day Daily[4] website with 10,884 and 3,628 sentences for train and test sets. The second dataset is collected from the DVB[5] website and it includes five subjects, with 8,201 and 2,733 sentences for the training and testing dataset. The third dataset is collected from The Voice[6] news

[3] https://www.rabbit-converter.org/.
[4] http://7daydaily.com/.
[5] http://burmese.dvb.no/.
[6] http://thevoicemyanmar.com/.

website, which covers five subjects with 7,660 and 2,586 sentences for training and testing dataset. The fourth dataset from the Thit Htoo Lwin[7] news website and includes five subjects with 12,299 and 4,099 sentences for testing and training datasets. The last dataset is collected from the Myanmar Wikipedia[8] website that contains four topics with 11,299 and 3,766 sentences for testing and training set. Table 3 summarizes these datasets.

**Table 3.** Myanmar text classification datasets

| Dataset | Train | Test | Classes |
|---|---|---|---|
| 7Day Daily | 10,884 | 3,628 | 5 |
| DVB | 8,201 | 2,733 | 5 |
| The Voice | 7,760 | 2,586 | 5 |
| Thit Htoo Lwin | 12,299 | 4,099 | 5 |
| Myanmar Wiki | 11,299 | 3,766 | 4 |

### 5.2 Comparison Models

In this work, we performed the comparative analysis of a joint CNN and Bi-LSTM model with CNN, RNN, Bi-LSTM, CNN-LSTM models.

**Convolutional Neural Networks (CNN):** It is an artificial neural network feed-forward, most widely used for visual image analysis. This model has recently achieved significant success in the tasks of text classification. It has three basic components, convolution, pooling, and fully connected layer. ReLU activation functions f(x) = max(x, 0) is used in convolution layer and it can have several layers of convolution. The pooling layer extracts the most important features. The pooling layer mostly applies Max-pooling. The fully connected layer is the model's output layer and it predicts the class of the input sentences. The fully connected layer commonly uses the Softmax function. Softmax function of $f(x)_i$ is $\frac{e^{x_i}}{\sum_j^C e^{x_j}}$, the probability of a class depends on the probabilities of all other classes.

**Recurrent Neural Networks (RNN):** RNN is a generalization of feedforward neural networks with the distinction that it has an internal memory that keeps information to persist. It performs the same function for all input data by learning from the previous data. RNN produces the output $y_t$ as in Eq. (2).

$$y_t = f\left(W_y h_t\right) \tag{2}$$

$$h_t = \sigma\left(W_h h_{t-1} + W_x x_t\right) \tag{3}$$

[7] http://www.thithtoolwin.com/.
[8] https://my.wikipedia.org/.

**Bidirectional LSTMs:** It is an extension of the LSTM model that can learn from the past and future information for a specific task.

**CNN-LSTM:** (Hassan A, 2018) proposed a joint CNN and LSTM framework to produce the feature map by CNN and to capture long term dependencies by LSTM.

### 5.3 Experimental Result and Discussion

The experiment is accomplished on Google Cloud Laboratory[9] that does not require to configure the Jupyter notebook by using, Keras[10], a model-level library. The performance of the CNN-Bi-LSTM model is compared with comparison models described in Sect. 5.2 as listed in Table 4. The highest performance sores for each dataset are highlighted in bold. According to the experiments, the proposed model improves accuracy in four out of five datasets. The CNN model performs equally with the proposed model in two datasets. The CNN-LSTM combined model performs better in two out of five datasets. We also measure the training time of each model. According to the measurement results, the CNN model requires the minimum training time because we used only one convolution layer. Although CNN-Bi-LSTM model performed better in three datasets than the remaining models, it requires more time for training than CNN-LSTM, RNN and CNN models. Average training time of each model are listed in Table 5.

**Table 4.** Comparison of average testing accuracy

| Datasets | Bi-LSTM | CNN | CNN-Bi-LSTM | CNN-LSTM | RNN |
|---|---|---|---|---|---|
| 7Day Daily | 98.07 | 98.13 | 98.11 | **98.14** | 97.74 |
| DVB | 91.92 | 90.24 | **92.14** | 91.58 | 82.97 |
| The Voice | 95.39 | **95.67** | **95.67** | 95.06 | 94.98 |
| Thit Htoo Lwin | 96.12 | **96.49** | **96.49** | 96.29 | 94.95 |
| Myanmar Wiki | 94.01 | 93.99 | 93.80 | **94.21** | 86.76 |

**Table 5.** Comparison of average training time

| Datasets | Bi-LSTM | CNN | CNN-Bi-LSTM | CNN-LSTM | RNN |
|---|---|---|---|---|---|
| 7Day Daily | 18 min 6 s | 1 min 47 s | 17 min 6 s | 9 min 9 s | 5 min 22 s |
| DVB | 19 min | 1 min 20 s | 13 min 22 s | 6 min 50 s | 3 min 23 s |
| The Voice | 13 min 19 s | 1 min 11 s | 11 min 6 s | 6 min 23 s | 3 min 14 s |
| Thit Htoo Lwin | 19 min 55 s | 1 min 34 s | 18 min 1 s | 9 min 53 s | 5 min 9 s |
| Myanmar Wiki | 18 min 55 s | 1 min 31 s | 16 min 56 s | 9 min 43 s | 5 min 11 s |

[9] https://colab.research.google.com.
[10] https://keras.io/.

## 6 Conclusion

This paper presents a joint CNN-Bi-LSTM model that take advantages of CNN to extract feature and Bi-LSTM to capture long term context information from both past and future information. A series of the experiment is performed by comparing the pro-posed model with CNN, Bi-LSTM, RNN, CNN-LSTM models in term of accuracy on five Myanmar articles datasets. According to the experiment, the proposed system per-forms better in three out of five datasets. The CNN model requires minimum training time than the remaining models and CNN-Bi-LSTM model takes more time than CNN, RNN, and CNN-LSTM models.

**Acknowledgement.** We deeply thank the anonymous reviewers for sharing their precious time to check our manuscript. We greatly thank the researchers who released pre-trained vectors publicly and these resources helpful for low resources languages. We greatly thank the friends who assist to collect and annotate Myanmar text datasets.

## References

1. Aye, Y.M., Aung, S.S.: Enhanced sentiment classification for informal Myanmar text of restaurant reviews. In: 16th International Conference on Software Engineering Research, Management and Applications (SERA), pp. 31–36. IEEE (2018). https://doi.org/10.1109/SERA.2018.8477231
2. Conneau, A., Schwenk, H., Barrault, L., Lecun, Y.: Very deep convolutional networks for text classification. In: The European Chapter of the Association for Computational Linguistics, EACL 2017 (2017). https://doi.org/10.18653/v1/e17-1104
3. Grave, E., Bojanowski, P., Gupta, P., Joulin, A., Mikolov, T.: Learning word vectors for 157 languages. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC-2018 (2018)
4. Hassan, A., Mahmood, A.: Convolutional recurrent deep learning model for sentence classification. IEEE Access **6**, 13949–13957 (2018). https://doi.org/10.1109/ACCESS.2018.2814818
5. Heinzerling, B., Michael, S.: BPEmb: tokenization-free pre-trained sub-word embeddings in 275 languages. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC-2018, pp. 31–36 (2018). https://doi.org/10.11588/data/V9CXPR
6. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pp. 427–431 (2017). https://doi.org/10.18653/v1/e17-2068
7. Khine, A.H., Nwet, K.T., Soe, K.M.: Automatic Myanmar news classification. In: 15th International Conference on Computer Applications, ICCA 2017, pp. 401–408 (2017)
8. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1746–1751 (2014). https://doi.org/10.3115/v1/D14-1181
9. Kyaw, T.N., Nyo, N.N.: Myanmar spam filtering based on Naïve Bayesian learning algorithm (MSFNBLA). In: 14th International Conference on Computer Applications, ICCA 2016 (2016)

10. Lai, S., Liheng, X., Kang, L., Jun, Z.: Recurrent convolutional neural networks for text classification. In: The Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
11. Mon, A.N., Pa, W.P., Thu, Y.K.: Exploring the effect of tones for Myanmar language speech recognition using convolutional neural network (CNN). In: Hasida, K., Pa, W.P. (eds.) PACLING 2017. CCIS, vol. 781, pp. 314–326. Springer, Singapore (2018). https://doi.org/10.1007/978-981-10-8438-6_25
12. Phyu, S.P., Nwet, K.T.: Article classification in Myanmar language. In: The Proceeding of 2019 International Conference on Advanced Information Technologies (ICAIT), pp. 188–193. IEEE (2019). https://doi.org/10.1109/AITC.2019.8920927
13. Song, X., Petrak, J., Roberts, A.: A deep neural network sentence level classification method with context information. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 900–904 (2018). https://doi.org/10.18653/v1/D18-1107
14. Yu, T., Nwet, K.T.: Annotation and sentiment analysis for Myanmar news. In: 16th International Conferences on Computer Applications, ICCA 2018
15. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. In: Advances in Neural Information Processing Systems, pp. 649–657 (2015)