JOINT WORD SEGMENTATION AND STEMMING FOR MYANMAR LANGUAGE

Yadanar Oo

UNIVERSITY OF COMPUTER STUDIES, YANGON

OCTOBER, 2019

Joint Word Segmentation and Stemming for Myanmar Language

Yadanar Oo

University of Computer Studies, Yangon

A thesis submitted to the University of Computer Studies, Yangon in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

October, 2019

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

....

Date Yadanar Oo

ACKNOWLEDGEMENTS

First and foremost, I would like to thank His Excellency, the Minister for the Ministry of Education, for providing full facilities during the Ph.D. Course at the University of Computer Studies, Yangon.

Secondly, a very special gratitude goes to Dr. Mie Mie Thet Thwin, Rector of the University of Computer Studies, Yangon, for allowing me to develop this research and giving me general guidance during the period of my study.

I would also like to extend my special appreciation and thanks to the external examiner, Professor Dr. Tin Htar Nwe, Professor, University of Computer Studies (Magway), for her patience in critical reading the thesis, the useful comments, advice and insight which are invaluable to me.

I am also very grateful to Dr. Khine Moe Nwe, Professor and Course-coordinator of the Ph.D. 9th Batch, University of Computer Studies, Yangon, for her valuable advice, moral and emotional support in my research work.

I sincerely would like to express my greatest pleasure and the deepest appreciation to my supervisor, Dr. Khin Mar Soe, Professor, University of Computer Studies, Yangon. Without her excellent ideas, guidance, caring, and persistent help, this dissertation would not have been possible.

It is with immense gratitude that I acknowledge the support, many insightful advice and suggestions of Dr. Win Pa Pa, Professor, the University of Computer Studies, Yangon.

I deeply would like to express my respectful gratitude to Daw Aye Aye Khine, Associate Professor, Head of English Department, for her valuable supports from the language point of view and pointed out the correct usage not only through the Ph.D. course work but also in my dissertation.

My sincere thanks also go to all my respectful Professors for giving me valuable lectures and knowledge during the Ph.D. course work.

I also thank my friends from Ph.D. 9th Batch for providing support, care, and true friendship along the way.

Last but by no means least, I must express my very profound gratitude to my family, especially my mother for always believing in me, for providing me with unfailing support and continuous encouragement, for their endless love throughout

my years of Ph.D. study and through the process of researching and writing this dissertation. This accomplishment would not have been possible without they.

ABSTRACT

Due to the powerful development of internet use, the amount of unstructured Myanmar text data has increased excessively. It is necessary to retrieve exact data for user query. The effectiveness of searching is obviously related to the stemming process. Identifying the stem word in a given text is an important aspect of any Natural Language Process. In Myanmar language, texts typically contain many different forms of a basic word. Morphological variants are generally the most common problem in mis-spellings, wrong translation and irrelevant retrieval query.

Since Myanmar written language does not use blank spaces to indicate word boundaries, segmenting Myanmar texts becomes an essential task for Myanmar language processing. Besides word segmentation, it is necessary to identify the stem words in the sentence. Stemming refers to the process of marking each word in the word segmentation result with a correct word type, for example, root word, single word, prefix, suffix, etc. The segmentation and stemming process are denoted as morphological analysis. During the process of word segmentation, two main problems occur: segmentation ambiguities and unknown word occurrences. There are basically two types of segmentation ambiguities: covering ambiguity and overlapping ambiguity. These ambiguities are dealt with known words. An unknown word is defined as a word that is not found in the system dictionary. In other words, it is an out-of-vocabulary word. For any languages, even the largest dictionary will not be capable of registering all geographical names, person names, organization names, technical terms and some duplication words, etc. Named entity recognition (NER), refers to recognizing entities that have specific meanings in the identified text, including persons, locations, organization, etc.

Normally, stemming is considered as a separate process from segmentation. In this new approach, segmentation, stemming and named entity detection are integrated as a lexical analysis system. This research contributes to integrate segmentation, stemming and named entity detection that would benefit in all these process. Although many stemmers are available for the major languages, there is no stemmer for Myanmar Language. The main reason is to produce Myanmar stemmer and it also solves the word segmentation problem and detects the named entities. This is the first work on joint Myanmar word segmentation, stemming and named entity detection.

Nowadays, deep learning approaches have become more and more popular in NLP tasks. This system proposes BiLSTM-CNN-CRF network architecture that jointly learns three processes. In this approach, stemming and named entity detection are considered as a typical sequence tagging problem over segmented words, while segmentation also can be modelled as a syllable-level tagging problem that identify the word boundaries via predicting the labels. This approach is an effective joint neural sequence labelling which predicts the combinatory labels of segmentation boundaries and stemming and named entity detection tag at the syllable level.

This research presents BiLSTM-CNN-CRF architecture that learns both character and syllable-level features, presenting the first evaluating of such architecture on Myanmar language evaluation datasets. This research also evaluates over different network architecture and many hyper parameters optimization such as pre-trained embedding, dropout rate, learning rate and different optimizers.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

CHAPTER 1
INTRODUCTION

A natural language is the preferred medium of communication for people and it can be in a spoken or written form, which is difficult to be simply understood by the computers. This needs a mechanism with enough information of the language including its word grammar and sentence structure to be understood by the computers. The processing of this information by a computer is known as natural language processing (NLP). NLP is used for both generating human understandable information from computer systems and adapting human language into more formal structures that a computer can understand. Natural Language generation and natural language understanding are key areas in the domain of natural language processing (NLP) and recent research has included areas like computational linguistics, bilingual transformation between others. These are subsets of the larger research area coined Artificial Intelligence (AI). [32] It means that computer performs many tasks like humans. It is a field of study which consists of different levels of linguistics analysis such as segmentation, stemming, syntactic and semantic analysis, and the basic levels are the segmentation and morphological stemming to different NLP applications.

1.1 Morphological Stemming

Morphological stemming is a process of segmenting words into morphemes, the assignment of grammatical information to grammatical categories and the assignment of the lexical information to particular lexeme or lemma [25]. It retrieves the grammatical features and properties of an inflected word. The stemmer breaks the word into minimal meaning bearing morphemes and produces the morph syntactic features such as the root, tense, person and number. A morphological stemmer is an essential and basic tool for building any language processing application in natural language for example, Machine Translation and it is an essential technology for most text analysis applications like information retrieval (IR) and text summarization.

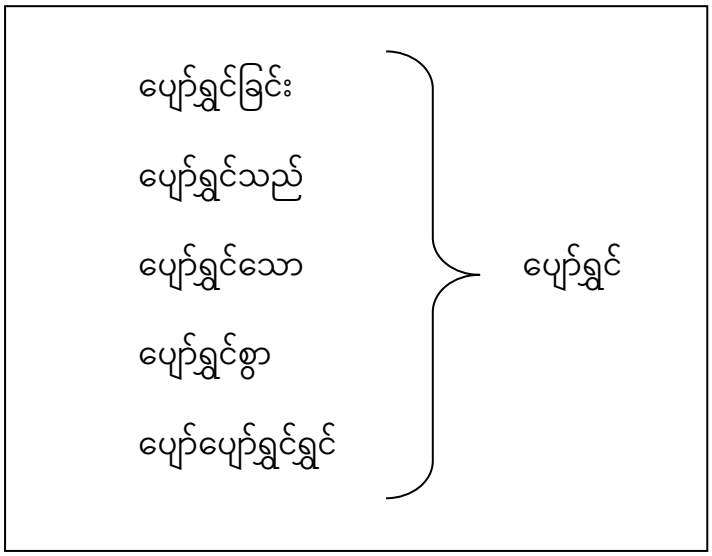

Figure 1.1 Example of Morphological Stemming

1.2 Problem Definition

Nowadays, enormous amount of data is available online. So, retrieval of accurate user query becomes the essential task. Stemming has been widely used to enhance the efficiency of Information Retrieval System [58]. In Linguistic morphology, stemming is the process of reducing inflected words to their root form. [39] In information retrieval system, stemming acts as an important tool to increase the retrieval accuracy. [50] In Myanmar language, stemming is performed by stripping suffix and affix from the given sentence. Texts typically contain many different forms of a basic word. Morphological variants are generally the most common problem in mis-spellings, wrong translation and irrelevant retrieval query. The effectiveness of searching is definitely related to the stemming process.

Myanmar written language does not have word boundaries. All texts need to be separated into syllable, words, sentences and paragraphs in order to explore the meaning of the document. [63] Word segmentation is the process of determining word boundaries in a piece of text. In English language, because of the presence of white spaces or punctuation between words, word boundaries can be simply determined. In Myanmar Language, segmenting sentences into words is a challenging task because sentences are clearly delimited by a sentence end marker, but words are not always delimited by spaces. [2] Spaces may sometimes be inserted between words and even between a root word and the associated post-position. It is because there are no indicators such as blank spaces to show the word boundaries in Myanmar text.

The same phenomenon does not happen only to Myanmar language but also many other Asia languages such as Japanese, Chinese and, Thai. In order to find the

root word in Myanmar text, it is necessary to cut the sentences into word segments. Although it sounds easy to cut a sentence into a word sequence, however, from the past experience, it is not a trivial task.

During the process of Myanmar word segmentation, two main problems are encountered: segmentation ambiguities and unknown word occurrence. Segmentation ambiguities are dealt with known words, for example, words found in the dictionary or in the corpus. An unknown word is not found in the dictionary or in the training corpus. In other words, it is an out-of-vocabulary word. For any languages, even the largest dictionary will not be able to cover all geographical names, organization names, technical terms, person names and some duplication words. Name entity detection is one of the issues in Asian Language that has traditionally required large amount of feature engineering to achieve high performance. Normally, segmentation is considered as a separate process from stemming and named entity detection. In this approach, word segmentation, stemming and named entity recognition are implemented as a joint process.

Traditional work used for stemming is affix removal method that removes suffix or prefix from words by using the rules, and converts them into a common stem form. In recent years, machine learning approach achieves good or state-of-the art results. Commonly used statistical approach are Hidden Markov Model and Conditional Random Fields with handcraft. Later, deep learning approach improves performance. This research has proposed a neural sequence labeling model that jointly learn word boundary and extract the stem word and named entity.

Moreover, words are considered as independent entities without any direct relationship among morphologically related word. So, some rare words are poorly estimated and unknown words are represented as only a few vectors. Word embedding is a good generalization to unseen words and that can capture general syntactic as well as semantic properties of word. Furthermore, deep learning approaches have become more and more prominent in NLP tasks and pre-trained embedding layers have been applied to enhance the efficiency of neural network architectures for many NLP applications because many machine learning algorithms and most of the deep learning architectures cannot process the raw form of strings or plain texts. Therefore, pre-trained embedding layers have been applied to improve the performance of neural network architectures for NLP tasks. The main target of word embedding model is to convert word to the form of numeric vectors. Most existing

word embedding results are generally trained on data source such as news pages or Wikipedia articles. In this system, different pre-trained embeddings are also evaluated.

The system is intended to find the stem word and named entity. It also detects the boundary of the word that are basic requirements for Natural language processing applications. Without word segmentation, other processing cannot be done. Stemming is also an essential step in Myanmar NLP application. Due to the dramatic growth of internet use, the amount of unstructured Myanmar text data has increased enormouly. Stemming has been extensively used in various Information Retrieval Systems to increase the retrieval accuracy. [58] Stemming is a method that reduces morphological similar variant of word into a single term called stems or roots without doing complete morphological analysis. In English, a word like "children" to its root "child" is an obvious necessity.

The importance of morphology, however, is even greater in a language like Myanmar, Japanese, Chinese or Korean. Asian text is written with limited or no space separations. The task of segmenting the initial text into a sequence of words is fully associated to the stemming process. Named Entities (NE) have a unique status and indicate particular concepts and things in the world which are not listed in grammar or lexicons. The purpose of this research is to introduce stemmer in Myanmar news data and to identify word boundary and named entity based on Myanmar morphological grammar. In doing so, I have designed and evaluated syllable-based tagging on Neural Network architecture.

1.2 Objectives of the research

In Myanmar Language, "word" is difficult to define normally, to produce the stem word or NE, word segmentation task is a preprocessing stage of stemming and so far, segmentation is considered as a separate process from stemming. In this system, the new approach is being integrated that would benefit in all processes. This research, focuses on syllable-based boundary tagging and proposes an approach for stemming and then recognizes the name entity at the same time

Throughout this work, the following objectives are pursued:

- To propose joint process for segmentation and stemming in Myanmar language

- To build stem word corpus for Myanmar language to be useful in NLP application
- To detect the named entity on joint process
- To use neural network architecture for joint word segmentation and stemming
- To support Text Categorization, Information Retrieval, Information Extraction, Text Summarization system and Machine Translation

1.3 Motivation of the Research

With the violent growth of online data, it is difficult to access relevant information from the internet at a short period of time. There are lots of approaches used to increase the effectiveness of online data retrieval. Stemming has been voluminously used in various Information Retrieval Systems to raise the retrieval accuracy. Stemming became an active field of research in both Information Retrieval (IR) and Natural Language Processing (NLP) communities. Asian text is written with limited or no space separations and segmentation is essential pre-processing requirement for many NLP applications.

Segmentation error would cause translation mistakes directly. Stemming also influences in accuracy of text categorization, IR and text summarization. Many word stemmers are available for the major languages, but they do not exit for Myanmar. The current named entity recognition (NER), which is a subtask of NLP, plays a vital role to achieve human level performance on specific documents such as newspapers to effectively identify entities. Myanmar word segmentation and stemming of this research aim to support Information Retrieval and Myanmar NLP applications.

1.5 Contributions of the Research

This research proposes a joint model that has stronger capabilities for Myanmar word segmentation and stemming. As far as we know, this is the first work on joint Myanmar word segmentation, stemming and named entity detection. The results of this research help to support basic requirements of later NLP processes in Myanmar Language.

The main contributions of the proposed system are as follows:

i. Propose a joint process. (Published in [P2])
ii. Build customized tag sets for segmentation, stemming and NE

iii. Build the corpus for joint word segmentation and stemming

iv. Compare the effectiveness of neural sequence labelling architectures that relies on two sources of information about syllable- and character-level representation, by using LSTM, CNN and GRU in joint process. (Published in [P2])

v. Explore the various hyper parameters and compare the experimental results. (Published in [P3] [P4])

During the neural training process, optimizers are key pieces that adjust and change the parameter of model to minimize the loss function and make predictions as possible as it is. Moreover, Overfitting is an unneglectable problem in deep learning, which can be effectively reduced by regularization.

vi. Give practical evaluations of different optimization functions and dropout rate.

vii. Evaluate the performance on different pre-trained embedding and take advantage of better pre-trained embedding.

1.6 Organization of the Research

This dissertation is organized with seven chapters. This chapter includes an introduction, the motivation of the thesis, the problem statements, objectives, motivation, focuses and contribution of the research work. Chapter 2 surveys the challenges and approaches of word segmentation, stemming and named entity detection on literature that deals with the dissertation. Chapter 3 explains introduction of Myanmar language and nature of Myanmar word and proposed tagging scheme for word segmentation, stemming and named entity detection. The theoretical background of the neural network architecture and neural sequence labeling model and the architecture of the proposed system is discussed in Chapter 4. The design and implementation of the proposed system are represented in Chapter 5. Chapter 6 describes the evaluation of the experimental results by using different architecture of the network design and different configuration to improve the performance of the joint model. Finally, Chapter 7 draws with the conclusion extracted from this research work and presents the future research lines.

CHAPTER 2
LITERTATURE REVIEW AND RELATED WORK

This chapter will describe the seven steps of machine learning and it also discusses the different approaches for Myanmar word segmentation and joint process for neural sequence labeling approaches in Asian language and English. This chapter has been divided into three parts. The first part is about the seven steps of machine learning. The second part is different approaches for Myanmar word segmentation, and the last part is other related research with neural sequence labeling approaches.

2.1 Introduction to Machine learning

Nowadays, machine learning is having a deep important broad area of applications for text understanding, image and speech recognition, to health care and genomics. [52] Machine learning is an artificial intelligence's application that supports systems the capability to automatically receive information and develop from experience without being explicitly programmed. In order to identify the faces in images, to predict weather, to detect skin cancer or to translate languages, machine learning allow computer system entirely new abilities. [21] There are seven steps in Machine Learning:

1. Data Collection
2. Preparing that Data
3. Choosing a Model
4. Training
5. Evaluation
6. Hyper-parameter Tuning
7. Prediction

2.1.1 Data Collection

Data collection is a dominant barrier in machine learning and an active research topic in numerous communities. [52] There are generally two facts data collection which has recently become a vital issue. First, as machine learning is developing more broadly-used, new applications are searched which do not naturally have enough labeled data. Second, unlike traditional machine learning where feature engineering is

the barrier, deep learning approaches automatically generate features, but instead require large amounts of labeled data. Firstly, a model is built. And a process called training is done. In order to train a model, it is needed to collect a data. In these data, there are a lot of features such as color, shape, etc. [71] So, the first step of machine learning is collecting data. This step is very prominent because a quality and quantity of data collected will directly concerned with creation of a model. Much of the current success in machine learning is as a result of better performing framework and large amounts of training data. There are many challenges in machine learning; data collection is becoming one of the vital barriers. It is known that the main part of the time for operating machine learning end-to-end is spent on preparing the data, which includes collecting, cleaning, understanding, and feature engineering. While all of these steps are time-consuming, data collection has recently become a challenge. [52] However, as machine learning requires to be executed on large amount of training data, data management concern with:

1. in what way large datasets are collected,
2. in what way data labeling is performed and
3. in what way the quality of large amount of existing data are promoted to become more appropriate

2.1.2 Data Preparation

Second step is data preparation. Machine learning helps us find pattern in data so we use our data into a suitable place and prepare it for use in our machine learning train process. Data preparation is the process of reconstructing raw data through machine learning algorithm to reveal observation or make prediction. [7l] The hardest problems to solve in machine learning are getting the right data in the right format. Suitable data preparation generates clean and well-curated data that introduces to more practical, accurate model outcomes. [72] Getting the right data indicates collection or classifying the data that correlates with the outcomes you want to predict; i.e. data that contains a signal about events you care about. We also need to split the data into two sets: training set and testing set. The first part is used in training model and it will be the major and largest part of the dataset. [72] Running a training set through machine learning approach teaches how effect different features, adjusting them coefficients according to their likelihood of minimizing error in the outcome.

These coefficients are parameters that will contain in the model because they encode a model of the data they train on.

The second part will be used in for evaluating our trained model's performance. Some data needs other forms of adjusting and manipulating such as normalization, error correction etc.

2.1.3 Choosing a Model

The next step is selecting an appropriate model. There are many models that researchers and data scientists have created over many years. [71] Some are well-suited for image, other for voice sequence, and some for numerical data and then other for text-based data. First of all, Machine learning tasks can be classified into:

1. Supervised learning
2. Unsupervised learning
3. Semi-supervised learning
4. Reinforcement learning

Supervised learning is the task of inferring a function from labeled training data. [73] The most optimal model parameters are found by fitting the labeled training set to predict unknown labels on test set. If the label is a real number, it is called task regression. If the label is limited number of values, it is called classification.

In unsupervised learning, there is less information about training data because data set has no labels associated with them. The goal of unsupervised learning is to organize data with its structure. It means that grouping it into cluster or finding some similarities between groups. [74] The major differences between supervised and unsupervised learning is that supervised learning algorithms are trained on datasets that are labeled by data scientist that guide the algorithm to understand which features are important. On the other side, unsupervised learning is trained on unlabeled data and determine important feature on their inherent pattern in the data.

Semi-supervised learning algorithms are trained on a combination of labeled and unlabeled data. Labeled data is used to identify specific groups of data types. The algorithm is then trained on unlabeled data to decide the boundaries of data types and identify new types of data that were undefined in the existing labeled data set.

Reinforce learning differs from other tasks in a way that other learning approach have labeled or unlabeled training dataset which has the answer key with it so the model is trained with correct answer itself whereas in Reinforces, there is no

answer but the reinforces agent decide what to perform the given task. It is about taking appropriate process to maximize reward in a particular condition. In the absence of training dataset, it is certainly learning from its experience. It is utilized by various software and machines to find the best possible solution in a specific situation.

2.1.4 Training

In the training process, our model's ability will be incrementally improved by using our training data. Training a model simply means learning (determining) optimal value for all the weights and the bias from labeled dataset. [71] Training process requires initializing some random values for W and b and attempting to predict the output with these values. These values are training parameters and adjust these values to control the learning algorithm make correct predictions. This process then repeats. Each iteration updating the weights and biases is called one training step.

2.1.5 Evaluating

Once training is complete, we should check whether the model is good or not by using evaluating the testing data. Evaluating the machine learning algorithm is an essential part. It allows us to test our model against data that are not including in training data. This measurement allows us to see how the model performs well on data that have never seen. The dataset is divided into training and testing set in order to check accuracies and precision. The portion is to be divided, mostly 80% of the data for training and the rest for testing. But, it is not essential and it completely depends on the dataset being used and the task to be completed. In machine learning, the bigger the dataset, the better the training.

2.1.6 Hyper-parameter Tuning

Hyper-parameters are configuration variables that are external to the model and whose values cannot be predicted from data. It can directly learn from data in the model training. After doing evaluation, the training model can be improved by tuning hyper-parameters. With the right values of hyper-parameters will eliminate the chances of overfitting and underfitting. One example is how many iterations are needed to train the dataset. This value plays a role in higher accuracies. Another hyper-parameter is learning rate. The learning rate hyper-parameters determines how fast or slow of model train. The adaptation or tuning of these hyper-parameters is an

experimental process that fully depends on the specifics of dataset, model and training process.

2.1.7 Prediction

Traditionally, machine learning models have not included awareness of why or how they arrived at an outcome. Prediction or inference refers to the process of inferring things about the world by applying model to new data. The importance of this phase is to minimize the scope of the model by removing any parts not necessary to make this prediction.

2.2 Myanmar Word Segmentation

In Myanmar language, there is no space between words. Word segmentation plays a vital role in later Natural Language Processing. There are a lot of research in Myanmar word segmentation. Most of the Myanmar NLP processes such as segmentation, stemming and POS tagging performance has been improved significantly, from the earliest Maximal Matching (dictionary-based) approaches to CRF approach. This section aims to categorize the various approaches that improve Myanmar word segmentation.

2.2.1 Challenges in dictionary-based longest matching

This section briefly reviews related works on longest matching approach on word segmentation. This paper is a very first attempt of segmenting Myanmar sentences into words [23]. In Myanmar language, dictionaries and other lexical resources are not yet widely accessible in electronic form. So, they firstly attempt to create a word hypothesizer that include stop word list and syllable level n-gram. And then, sentences are split by using the stop word removal and syllable n-gram. The accuracy is about 65%.

In [22] the author introduced Myanmar word segmentation. Firstly, they have collected 4550 syllables from available sources of 2,728 sentences and build the words lists from available sources including dictionaries and by generating syllable n-grams as possible words, a total of 800000 words. Secondly, word segmentation is carried out with the longest syllable word matching using their 800000 strong stored word list. There are two weaknesses. The segmentation error can occur in out-of-vocabulary word because there always exist oov words such as new derived word,

new compound words, morphological variation of existing words, technical term and named entity. Another limitation is the segmentation error can occur because of left-to-right processing. In order to fit these weaknesses, they introduce hybrid approaches. Example of segmentation error in due to left-to-right limitation is shown in Figure 2.1.

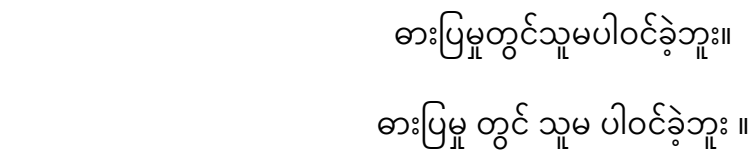

Figure 2.1 Example of Segmentation Error in left-to-right Limitation

A hidden Markov model is a tool for representation probability distributions over sequences of observations. [15] Markov Models have been applied in part-of-speech tagging for Myanmar language. Since English texts consist of blank spaces to indicate the word boundaries, the only problem is to assign the POS tags. However, in our language, Myanmar having no spaces to mark the word boundaries, so segmentation of words is done by maximum matching algorithm using dictionary.

The Hidden Markov Model is a finite set of states, each of which is correlated with probability distribution. [12] Transitions among the states are driven by the set of probabilities called transition probabilities. In a particular state observation can be generated, according to the associated probability distribution. Observation refers to the data we know and can observe. It is outcome but states are not visible and hidden to an external observer thus the name Hidden Markov Model.

In [42], customized tagsets are proposed for POS tagging in Myanmar language. And word segmentation is performed as a prerequisite phase. In order to annotate basic POS tags, a lexicalized HMM-based approach is applied. Moreover, normalization rules are applied in standard POS tagging and rule-based chunker is proposed. The evaluation and experimental results show that the proposed approaches achieve the high accuracy, over 90%, for most of the structured Myanmar sentences.

2.2.2 Challenges in Hybrid Approaches

In [63], the author proposed a method which has two phases: syllable segmentation and syllable merging. A rule-based heuristic approach was adopted for syllable segmentation and a dictionary-based statistical approach for syllable merging.

They proposed six rules for syllable segmentation and tested over 32,567 syllables but no error was occurred. So, syllable segmentation achieves 100% accuracy. In syllable merging, they proposed a dictionary-based statistical approach. Segmented syllable is merged in to all possible combination and the mutual information of two syllable are pre-calculated with the corpus and used to calculate the collocation strength of a sentence or phrase and then select the combination with the biggest collocation strength of merged words. They use mixture of Dictionary, Corpus, Longest Matching approach and Statistical approach. During the syllable merging process, three types of errors can occur. First, missing common words in dictionary. Second, spelling errors and unknown proper noun such as names of people and place are not listed in dictionary. Third, an infrequent occurrence of Pali word.

In [46], word segmentation system consists of four components, sentence splitting, tokenization, initial segmentation by Maximum Matching Algorithm and statistical combined model (bigram model and modified word juncture model) for final segmentation. They use the combination of Corpus, Statistical approach and Longest Matching approach.

2.2.3 Statistical-based Approaches

Conditional Random Fields (CRF) is undirected graphical models trained to maximize a conditional probability of the whole graph structure. [28] A common case of a graph structure is a linear chain, which correlates with a finite state machine, and is relevant for sequence labeling. A linear chain CRF with parameters $\lambda = \{\lambda_1,\ldots..,\lambda_k\}$ defines a conditional probability for a label sequence $y = y_1,\ldots,y_T$ given an input sequence $x = x_1,\ldots.x_T$ to be:

$$P_\lambda = \frac{1}{Z_x}\exp\left(\sum_{t=1}^{T}\sum_{k}\lambda_k f_k(y_{t-1}, y_t, x)\right) \qquad (2.1)$$

where Z_x is the normalization factor that makes the probability of all state sequences sum to one; $f_k(y_{t-1}, y_t, x)$ is a feature function, and λ_k is a learned weight associated with feature f_k. The feature function measures any aspect of a state transition, $y_{t-1} \longrightarrow y_t$ and the entire observation sequence, x . Large positive values

for λ_k indicate a preference for an event and large negative value make the event unlikely.

The most probable label sequence for an input x;

$$y^* = argmax_y \, P_\lambda(y|x) \tag{2.2}$$

can be effectively determined using the Viterbi algorithm.

CRFs are trained using maximum likelihood estimation, i.e., maximizing the log-likelihood L_λ of a given training set $T = <x_i,y_i>_{i-1}^N$,

$$L_\lambda = \sum_i logP_\lambda(y_i|x_i) \tag{2.3}$$

$$= \sum_i (\sum_{t=1}^{T} \sum_k \lambda_k f_k(y_{t-1}, y_t, x) - logZ_{x_i} \tag{2.4}$$

In this implementation, quasi-Newton method is used as the learning algorithm for parameter optimization, which has been shown to converge much faster. To avoid over-fitting, log-likelihood is penalized with Gaussian prior. CRFs are discriminative models and can capture many correlated features of the inputs. Therefore, it is suitable in many tasks in NLP for sequence labeling. Since they are discriminatively-trained, they are often more accurate than the generative models, even with the same features.

The CRF model has focused on the labeling bias issue and eliminated two unreasonable hypotheses in HMM. Of course, the model has also become more complicated.

HMM directly creates the transition probability and phenotype probability, and computes the probability of co-occurrence. Hence, it is a generative model. MEMM is not a generative model, but a model with finite states depends on state classification. MEMM provides the probability of co-occurrence depend on the transition probability and the phenotype probability. It computes the conditional probability, and only follows the local variance normalization, making it easy to fall into a local optimum. HMM and MEMM are a directed graph, while CRF is an

undirected graph. MEMM follows local variance normalization while CRF follows global variance normalization. CRF is a discriminant model. CRF computes the normalization probability in the global scope, instead of the local scope as is the case with MEMM. It is the best global solution and settles the labeling bias issue in MEMM.

In [47], Conditional random field is used to identify Myanmar word boundaries within a supervised framework. CRF approach is compared against a baseline based on maximum matching using dictionary from Myanmar Language Commission Dictionary (word only) and manually segmented subset of the BTEC1 corpus.

In [8], this paper studied Khmer word segmentation using a conditional random fields based approach. In order to train the segmenter, manually-segmented corpus is created. During the manual annotation, by using the human annotators, it provided details of a set of word segmentation strategies. The trained CRF segmenter was compared to a baseline maximum matching approach.

[69] In this paper described a Chinese word segmentation system based on Conditional random fields approaches. In the system, a character in the given sequence is labeled by a tag which stands for its position in the word that the character belongs to.

In [70], this paper concerned with Chinese word segmentation, which is regarded as a character based tagging under conditional random field framework. In this approach, both of feature template selection and tag set selection are considered instead of focusing only on feature template.

In [31], the author of this paper studied on word boundary decision (WBD) approach and implements it as a 2-tag character tagging with CRF approach. With a help of tag transition features, WBD with CRF segmentation approach achieves comparative performances compared to 4-tag character tagging approach.

In [44], this paper reported a careful investigation of two successful statistical learning method CRF and SVM. Then, CRF and SVM models are trained on the corpus applying different feature settings and their performances are analysed and compared with each other to determine the effect of feature selection as well as the generalization ability of CRFs and SVMs on the segmentation accuracy.

2.2.4 Deep learning Approaches

A machine learning algorithm is used to parse data, learn from that data, and generate knowledgeable decisions based on what it has learned. [75] Mostly, deep learning is used in layers to create an Artificial Neural Network that can learn and generate intelligent decisions on its own. So, deep learning is a subset of machine learning. Machine learning focuses on solving real-world problems. It also takes a few ideas from Artificial Intelligence. Machine learning goes through the Neural Networks that are designed to human decision making capabilities. Machine learning approaches are two key narrow subsets that only focus on deep learning.

In the recent research literature, neural models can be challenging. [65] the author explored three neural model designs: character sequence representation, word sequence representation and inference layer. Experiments show that character information improves model performance.

In [51], they evaluated different network design choice and selection hyperparameters optimization for neural network. Experiments revealed that the choice of word embedding, the selected optimizer, the classifier used as last layer, and the dropout mechanism has a high impact on the achieved performance.

Artificial intelligence community has increasingly used word embedding for optimization of neural architecture; see for example [35], in this paper, they firstly used convolutional neural networks (CNNs) to encode character-level information of a word into its character-level representation. Then, they combined character- and word-level representations and feed them into bi-directional LSTM (BLSTM) to model context information of each word. Finally, sequential CRF is jointly used to decode labels for the whole sentence. They achieved competing performance against traditional models. In order to test the importance of pre-trained word embedding, experiments perform with different sets of word embedding, as well as a random sampling method, to initialize the model.

In [57], they proposed a character-based model for joint segmentation and POS tagging for Chinese that use bidirectional RNN-CRF architecture with novel vector representations of Chinese characters that capture rich contextual information and sub-character level features. In addition to utilizing the pre-trained character embedding, they proposed a concatenated n-gram representation of the characters. They converted rich local information in the character vectors via utilizing the incrementally concatenated n-gram representation.

In [40], they evaluated the word representations (with and without postprocessing) using four different neural network architectures (CNN, vanilla-RNN, GRU-RNN and LSTM-RNN). They used Word2Vec and Glove and allow the parameter D for dimension of choice to vary between 300 to 1000.

In [24], the author paper proposed a variety of LSTM-based model for sequence tagging (such as LSTM network, BI-LSTM network, LSTM-CRF, BI-LSTM-CRF). They showed that BI-LSTM-CRF model is more efficient because they use both past and future input features.

In [68], this paper showed that joint models have a stronger capability for Chinese word segmentation and POS tagging. It also presented a simple yet effective sequence-to-sequence neural model for the joint task, based on LSTM neural network structure. Moreover, by using well-trained character-level embedding, the proposed neural joint model got the best performance.

In [34], to overcome the poorly elimination of rare and complex word and all unknown words were represented by using a few vectors, this paper proposes a novel model that is capable of building representations for morphologically complex words from their morpheme. They combined recursive neural networks (RNNs) with neural language models (NLMs) to consider contextual information in learning morphologically aware word representations. Their main advantage was having such a distributed representation over word classes can capture various dimensions of both semantic and syntactic information in a vector where each dimension corresponds to a latent feature of the word.

2.3 Challenges in Morphological Stemming

With the enormous growth of World Wide Web, stemmer plays a critical part in natural language processing applications for lexical analysis, information retrieval or language specific search engine in order to find user specified keywords in indexing database more quickly. Although many stemmers are available for major languages, but they are not much proficient for the less computerized languages including Myanmar. Myanmar stemmer is a difficult task in natural language processing since the nature of Myanmar script is complex rather than the English language. Myanmar sentences do not have white space to specify words boundaries and hence word segmentation is essential as a first step for Myanmar stemmer. Moreover, there is lack of resources for Myanmar text such as WordNet, ontology

representation, parsing the keywords and their part-of-speech. So, the overall task of information retrieval becomes complex and time consuming. Myanmar documents always carry suffixes which may cause the problem in accurate information retrieval.

For some international languages like Hebrew, Portuguese, Hungarian, Czech and French and for many Indian languages like Bengali, Marathi and Hindi, stemming has been extensively used to raise the accuracy of Information Retrieval Systems. [58] There are four automatic approaches namely Affix Removal Method, Successor Variety Method, n-gram Method and Table lookup method.

In [16], this system, input is randomly picked Marathi document and output is sequence of root words. They have created corpus containing more than 3000 possible stop words and suffixes for Marathi language. The corpus performs very critical part for filtration of input document. Performing stemming and removing inflections of the word is very important task to retrieve relevant and detailed information from the document.

In [39], stemmer used the Hybrid approach (combination of brute force and suffix removal approach). Brute force search is also known as exhaustive search. This approach applies a lookup table which contains relations between root words and inflected words. In this approach, large number of inflected word along with their corresponding root word are needed to store. It also reduces the problem of over-stemming and under-stemming which was found in A light weight stemmer for Hindi.

In [1], the author proposed algorithm uses stemming operation to prepare the word for use, and then it is compared to a list of different morphological weights. A list of different morphological weights to examine the proposed algorithm has been prepared to ensure selection of the maximum amount of morphological inflictions. The proposed method was characterized by not using any special roots to compare with, but based on a series of steps to find the root.

Stemming algorithm can be broadly classified into two categories, namely Rule based and Statistical Approach.

2.3.1 Rule-Based Approach

In a rule-based approach, language specific rules are encoded and based on these rules, stemming is performed. In this approach, there are various conditions to convert a word to its derivational stem. A list of all valid stems are given and also there are some exceptional rules which are used to handle the exceptional cases.

In [49], M. F. Porter composed of a set of conditional rules. These conditions are either applied on the stem or on the suffix or on the stated rules. In [33], J.B. Lovins proposed an approach for stemming. In this approach, stemming comprises of two phases: In the first phase, the stemming algorithm retrieves the stem from a word by removing its longest possible ending by matching these endings with the list of suffixes stored in the computer and in the second phase spelling exceptions are handled.

Benefits:

1. Rule-based stemmers are fast and to find a stem, it requires less computation time.
2. In English language, by using Rule-based stemmer, the retrieval results are very high.

Weakness:

1. But rule based stemmer requires to have extensive language expertise to make them.
2. To store rules for stem extraction from the words and some exceptional cases, it requires large amount of storage.
3. These stemmers may apply over stemming and under stemming to the words.

2.3.2 Statistical Approach

Statistical stemming is an effective and popular approach and statistical stemmers are good alternatives to rule-based stemmer. Additionally, they do not require language expertise. Rather they employ statistical information from a large corpus of a given language to learn morphology of words. In [3], M. Bacchin, N. Ferro, and M. Melucci took a step forward from the graph-based stemming algorithm just described and to introduce a probabilistic framework which models the mutual reinforcement between stems and derivations. By concatenating prefixes or suffixes, stemming is considered as the inverse of a machine which generate words. This paper shows how the estimation of the probabilities of the model relates to the notion of mutual reinforcement and to the discovery of the communities of stems and derivations.

In [36] P. Majumder, M Mitra, S.K. Parui, and G. Kole (ISI), P. Mitra (IIT), and K.K. Dutta proposed an approach for statistical stemmer. In this approach, a set of string distance measure was defined and complete linkage clustering was used to discover equivalence classes from the lexicon. The string distance measure was used to check the similarity between two words by calculating the distance between two strings, the distance function maps a pair of string to a real value where a smaller value indicates greater similarity between two strings.

Benefits:

1. Statistical stemmers are useful for language that have limited amount of resources. For example, in Asian languages, they heavily used in Asian Sub Continent but very less research is done on these languages.
2. In morphologically more complex language such as French, Hindi, Portuguese and Bengali, statistical stemmer generates the best retrieval results.

Weakness:

1. Most of the statistical stemmer does their statistical analysis based on some sample of the actual corpus. As sample size decreases, the possibility of covering most morphological variants will also decrease.
2. Statistical stemmers are time consuming because for these stemmers to work we need to have complete language coverage, in terms of morphology of words, their variants, etc.

2.4 Myanmar Named Entity Detection

During the word segmentation, two problems are encountered: segmentation ambiguities and unknown word occurrence. Unknown words are not found in dictionary or in training data. Most of the Myanmar named entity are not found in dictionary or in lexicon. NER systems have been developed for resource-rich language like English with very high accuracies. NER for Myanmar language is a challenging task since Myanmar is very rich in morphology. Currently, Myanmar NLP is at a fundamental phase and lexical resources usable are highly restricted. On the other hand, it has both complicated and rich morphology as well as uncertainty. These facts can lead to wrong word segmentation.

In [37] The earlier introduced approaches for NER based on classified into dictionary-based or rule-based and traditional statistical sequence labeling approaches and hybrid. Those approaches need linguistic information and feature engineering.

Former effort on Myanmar NER had been accomplished by rule-based and statistical research works. A method for Myanmar Named Entity Identification using a hybrid method was proposed by [62] Thi Thi Swe, Hla Hla Htay. Their method was an aggregation of rule-based and statistical N-grams based method and name database was used as well.

In [43] Thida Myint, Aye Thida, proposed a Myanmar Named Identification algorithm that defines the names by using some of the POS information, NE identification rules and clues words in the left and/or the right contexts of NEs that carry information for NE identification. Their limitation is that input sentence must be with specified POS tags. As a weakness, sematic implication of proper names is ineffective. Moreover, those approaches required linguistic information and feature engineering. The main disadvantages of these rule based method is that they need vast experience and grammatical knowledge of the particular language or domain and these systems are not easily adaptable to other domains or languages.

Recent systems rely on machine learning approaches, but their performance is highly dependent on size and quality of training data. In [48] Shamima Parvez introduced NER system in Bengali news data to identify events of specified things in running text based on regular expression and Bengali grammar. In doing so, they have designed and evaluated part-of-speech (POS) tags to recognize proper nouns. Hidden Markov Model (HMM) has been applied for developing NER system from Bengali news data. In [4] this paper proposes the Named Entity Recognition (NER) system for Punjabi language using a hybrid approach in which rule based approach and machine learning approach Hidden Markov Model (HMM) is combined.

In [14] Asif Ekba, Rejwanul Haque, Sivaji Bandyopadhyay proposed the system which use the statistical Conditional Random Fields (CRFs) for the development of NER system for Bengali. The system made use of the different contextual information of the words along with the variety of features that are helpful in predicting the various named entity (NE) classes. It is the current trend in NER is to use the machine-learning approach, which is more attractive in that it is trainable and adoptable and the maintenance of a machine-learning system is much cheaper than that of a rule-based one.

The authors Xinnian Mao, Yuan Dong, Saike He, Sencheng Bao, Haila Wang in [38] introduced Chinese word segmentation (CWS), named entity recognition (NER) and part-of speech tagging is the lexical processing in Chinese language. In order to enhance the low recall in NER system; they applied non-local features and mitigate class imbalanced distribution on NER dataset to progress the recall and keep its relatively high precision. Some other post-processing measures such as consistency checking and transformation-based error-driven learning were used to improve word segmentation performance. Their systems participated in most CWS and POS tagging evaluations and all the NER tracks.

In [59] the authors N. Sobhana, P. Mitra, S.K. Ghosh, proposed a Named Entity Recognition (NER) system for Geological text using Conditional Random Fields (CRFs). This system used the different kind of features to identify NE's. The features aids in deciding to which class a named entity belongs. The main features for the NER task have been identified based on the different possible combination of available word and tag context. The features also included prefix and suffix for all words.

Although statistical approaches such as CRFs have been widely applied to NER tasks, those approaches heavily rely on feature engineering. In [9] "Jason P.C. Chiu, Eric Nichols" introduced two layer of neural network architecture for NER that use no features. This paper presented a novel neural network architecture that automatically detects word- and character-level features using a hybrid bidirectional LSTM and CNN architecture, eliminating the need for most feature engineering. The author employed bi-directional recurrent neural network with long short-term memory units to transform word features into named entity tag scores. For each word they employed a convolution and a max layer to extract a new feature vector from the per-character feature vectors such as character embeddings.

The authors [29] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami and Chris Dyer introduced two new neural architectures—one based on bidirectional LSTMs and conditional random fields, and the other that constructs and labels segments using a transition-based approach inspired by shift-reduce parsers. introduced two new neural architectures—one based on bidirectional LSTMs and conditional random fields, and the other that constructs and labels segments using a transition-based approach inspired by shift-reduce parsers. Their models relied on two sources of information about words: character-

based word representations learned from the supervised corpus and unsupervised word representations learned from unannotated corpora.

In [55], the authors presented Segment-level Neural CRF, which combines neural networks with a linear chain CRF for segment-level sequence modeling tasks such as named entity recognition (NER) and syntactic chunking. Their method applied segment lattice constructed from the word-level tagging model to reduce the search space.

In [18], Andrej Zukov-Gregoric, Yoram Bachrach and Sam Coope proposed a new parallel recurrent neural network model for entity recognition. They showed that rather than using a single LSTM component, as many other recent architectures have, they instead resort to using multiple smaller LSTM units. This has the benefit of reducing the total number of parameters in their model.

In [26], Zhenyu Jiao, Shuqi Sun, Ke Sun proposed Chinese Lexical Analysis with Deep Bi-GRU-CRF Network. They introduced a deep Bi-GRU-CRF network that jointly models word segmentation, part-of-speech tagging and named entity recognition tasks. Their main purpose was jointly accomplishing three tasks. The model worked in a full end-to-end manner and it is effective and efficient.

2.5 Summary

In this chapter, there are four parts. The first part is the introduction to machine learning. In this part, seven essential steps in machine learning have been explained briefly. In the second part, Myanmar word segmentation and challenges in different approaches with related papers have also been explained. Moreover, word segmentation problem in Asian language and different approaches such as statistical-based and deep learning approaches have been described.

In the third part, challenges in morphological stemming and different related papers are described. Different between rule-based approaches and statistical approaches are also analyzed. In the last part, Myanmar named entity and named entity detection for other languages are also explained.

Traditional work used for stemming is affix removal method that removes suffix or prefix from words so as to convert them into a common stem form. In recent year, machine learning approach achieve good or state-of-the art results. Commonly use statistical approach are Hidden Markov Model and Conditional Random Fields with handcraft. Later, deep learning approach improves performance. This research

proposes segmentation and stemming as a joint sequence labelling problem by using deep learning approach. Moreover, we also added parameter tuning to improve the performance.

CHAPTER 3
MYANMAR WORD SEGMETATION AND STEMMING

This chapter briefly describes introduction to segmentation, stemming and named entity detection approaches and the analysis of Myanmar Language, and the proposed tagging scheme for joint word segmentation, morphological stemmer and named entity detection in Myanmar Language. This chapter is divided into three parts: in the first part introduction of segmentation, stemming, and named entity detection are briefly described. In the second part, eight part-of-speech classes of Myanmar words are briefly explained, and morphological tagging scheme for stemming is described as in the third part. In order to search the morphological stem word, it is needed to segment the sentence into meaningful word. Firstly, syllable must be identified and then word boundary is detected and stem word and named entity can be identified by using a syllable tagging scheme.

The proposed joint word segmentation, morphological stemming and named entity detection employs the customized tag sets which is divided into two groups. Hence, the first part is for word boundary detection and the second part is for morphological stemmer and named entity detection. The proposed joint process is trained with neural sequence labeling model. Syllable segmentation is performed as the preprocessing phase.

3.1 Introduction

In the applications of Natural Language Processing(NLP), morphological analysis of the sentences is one of the important tasks for machine translation, text summarization, text categorization and information retrieval. For a complete morphological analysis of the sentences, word segmentation, stemming and named entity detection are needed to perform. Myanmar sentence structure is subject, object and verb, and most of the sentences consist of named entity. To perform morphological analysis of a sentence, word segmentation is a fundamental task.

3.1.1 Segmentation

Word segmentation is the essential part in natural language processing (NLP). [19] Most NLP applications need given sentences to be segmented into single

meaningful words before other processing. For instance, in machine translation, words must be first segmented into a series of terms and then analyzed it grammatically and interpreted into another language. [54] In information retrieval, word documents or word queries are entered as input and these input documents also need to first segmented into single words. The processed terms are then arranged into an inverted file index data structure for rapidly retrieve the query. In speech synthesis process, the tokenized words are segmented further into syllables, which are then developed into phoneme units.

Word segmentation system can be categorized into two specific approaches: dictionary-based (DCB) and machine learning-based (MLB) [19]. DCB approaches depend on sets of stored words in dictionary to parse and segment input sentences. DCB approaches use a set of words from a dictionary for parsing and segmenting input text into word tokens. During the parsing process, it discovers series of characters in the dictionary to observe equivalent words. The effectiveness of DCB approaches rely on the size and nature of the dictionary. Although DCB approaches are almost easy and simple, two problems are encounter with this approach. The first one is concerned with unknown word. [20] Unknown words are word that do not include in the dictionary. The second problem is ambiguity in parsing. Ambiguity problem happened a given character sequence can be segmented more than one way. Ambiguity can be solved by selection of heuristics such as choosing the longest word (longest matching) or choosing the segmentation yielding the minimum number of word tokens (maximal matching).

However, MLB approaches depend on statistical models predicted from training corpus using machine learning techniques [19]. Some popular machine learning approaches are: decision tree, Naive Bayes (NB), Support Vector Machine (SVM), Conditional Random Field (CRF) and Neural Network (NN). MLB approaches intend to solve the shortcomings of DCB approaches. Machine learning approaches use the tagged corpus in which word boundary are tagged by the special annotation. And then machine learning algorithm produces the statistical models based on the surrounding characters features. The identities and categories of characters within an n-gram of character surrounding word boundaries are most common features in machine learning approach. For word segmentation, character types are characteristics.

In MLB approaches, the process of word segmentation is considered as a binary classification. In this classification task, each character is denoted as B or I in the corpus. A character is labeled as "B" if it is beginning of the word. Otherwise, "I" if it is intra-word character. In MLB approaches, dictionaries are not required so it is the main convenience. To achieve correct classification, the unknown word and ambiguity problems are handled in principle by selecting adequately rich contextual information from the n-gram and by developing adequately large set of training examples.

The performance of the MLB approaches rely severely on the size of the training corpus and the characteristics of the document domains. [19] It is the main drawback of MLB approaches. For instance, a model might not effect well on documents from other domain if it is developed based on a corpus from one specific domain. The machine learning approaches such as SVM, CRF require linguistic knowledge and feature engineering. NN models have the advantage of minimizing the effort in feature engineering, since deep layer of neural networks can discover relevant features to tasks.

As a consequence of the existence of white spaces or punctuation among word, word boundaries are easily determined in English text [63]. Inversely, because of no inter-word spacing or other delimiter in the text, word segmentation task is not straightforward in Asian languages. For word segmentation, Myanmar words could cause a problem because of the absence of definite description. If there is no agreement on word segmentation, a training corpus will not be very useful. [63] Because of the different person segment the word in different way so it is not compatible or sometimes in conflict. Actually, word segmentation could be inconsistent even though it is carried out by the same person.

For Asian languages, most research on this task has addressed the segmentation and morphological analysis of Myanmar, Chinese, Japanese, and Vietnamese for which the standard, state-of-the-art technique using conditional random fields has accomplished adequate results [11].

3.1.2 Stemming

Stemming is the approach of generating morphological derived form of a root word. Stemming is a factor of linguistic analysis in morphology and information

retrieval and extraction. Stemming algorithm extracts meaningful information from vast sources like big data and search engines. It is a pre-processing step in information retrieval and also a common requirement of many other NLP applications. The fundamental desire of stemming is to reduce different grammatical forms or word forms of a word like noun, adjective, verb, adverb, etc. to its root form. Stemming is usually done by extracting any attached suffixes and prefixes (affixes) from index terms. [6] A stemming algorithm reduces the words “computer”, “computing”, “computation” to the root word, “comput”, and “retrieval”, “retrieved”, “retrieves” reduce to the root word “retrieve”.

Two main errors encounter in stemming – over stemming and under stemming. Over-stemming means when two words with disparate stems are stemmed to the equivalent root. This is also known as a false positive. Under-stemming means when two words that should be stemmed to the same root are not. This is also known as a false negative. [41] Over-stemming errors can be reduced by light-stemming but increases the under-stemming errors. On the other hand, heavy stemmers reduce the under-stemming errors while increasing the over-stemming errors.

Generally, stemming algorithms can be classified into three groups: truncating methods, statistical methods, and mixed methods. Each of these groups has a common approach of finding the stems of the word variants.

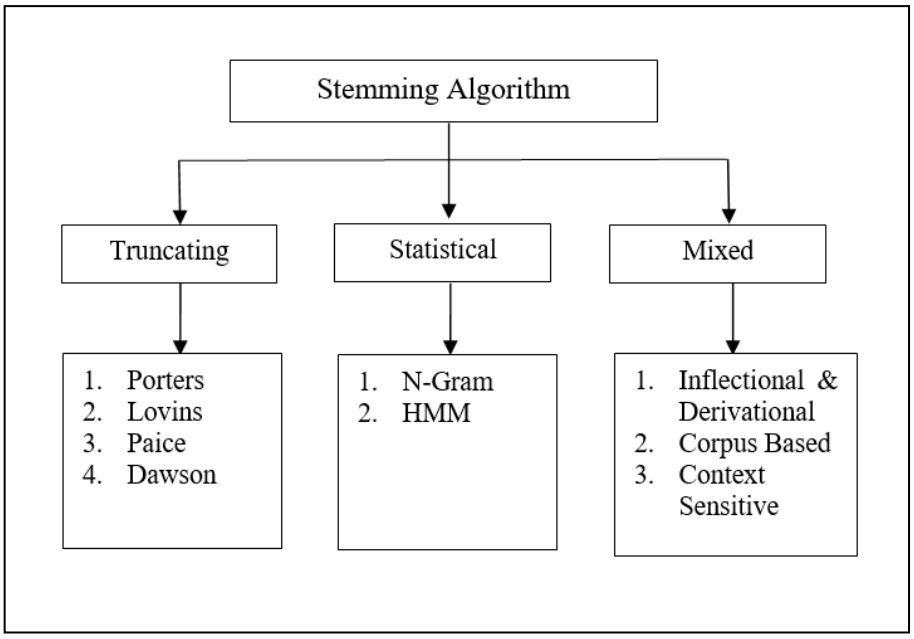

Figure 3.1 Types of Stemming Algorithm

3.1.3 Named Entity Detection

Named entity detection is an important task that has generally needed large amounts of knowledge in the form of feature engineering and lexicons to attain high performance. In most languages and domains, there is only an extremely small amount of supervised training data available [60]. On the other hand, there are few constraints on the kinds of words that can be names, so generalizing from this small sample of data is difficult. Moreover, there are two main problems in segmentation: word ambiguity and unknown word occurrence. Unknown words mean that they are not found in dictionary or in training data. In this approach, unknown words are considered as a named entity. Named-entity detection refers to a data extraction task that is responsible for finding different categories such as person name, organization name, location from the given sentences.

There are two main approaches for NER: rule-based approaches and statistical approaches.

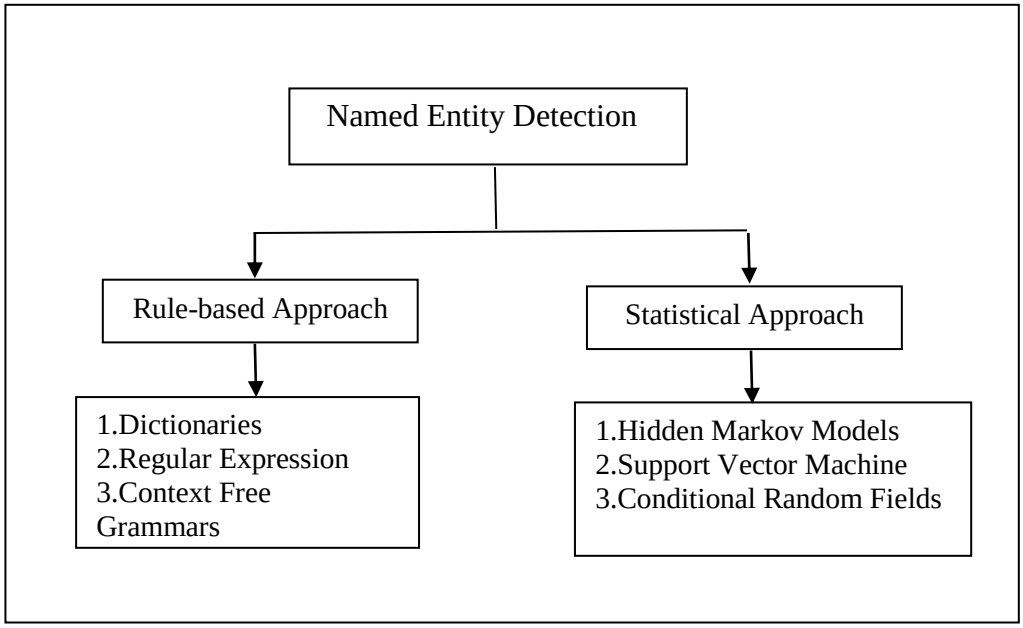

Figure 3.2 Types of Named Entity Detection Approaches

3.2 Introduction to Myanmar Language

The Myanmar language, Burmese, belongs to the Tibeto-Myanmar language group of the Sino-Tibetan family. In Myanmar Language, there are 33 consonants and 9 vowels. It is written from left to right. It is also morphologically rich and agglutinative language. It is written from left to right. A Myanmar sentence is the

combination of two or more phrases. Two or more words come together to form a phrase. A word consisting of two or more stems joined together is a compound word. Myanmar has both complex morphology and orthography, where stems are typically derived from a closed set of roots to which affixes such as coordinating conjunctions, determiners, and pronouns are attached to form words. Myanmar words are postpositionally inflected with various grammatical features [2].

Segmenting sentences into words is a challenging task in Myanmar language because sentences are obviously delimited by a sentence boundary marker but words are not always delimited by spaces. Spaces may sometimes be inserted between words and even between a stem word and the associated post-position. Words can be combined to make the new words. Moreover, a word can combine with two or more suffix to form a compound word. And then, prefix can also join together in the stem word. Segmenting and stemming Myanmar words into their constituent parts is important for a variety of natural language processing applications. For example, segmentation has been shown to improve the effectiveness of information retrieval and machine translation.

3.2.1 Myanmar Sentence

Basically, there are two types of Myanmar sentences: formal and informal. Formal sentences are used in official letters, textbooks, newspapers, online news, etc. and can be called as written form. Informal sentences are used in spoken language and can be defined as conversational form. In Myanmar language, there is no white space between words, but the sentences are delimited by sentence end marker called "။" pote-ma. In order to process any NLP tasks, the sentences are firstly separated by using sentence end marker.

3.2.2 Myanmar Word

There are two kinds of word according to the grammatical structure of Myanmar language. They are single word (SW) and compound word (CW). In Myanmar Language, segmenting sentences into words is a challenging task because there is no space separation between words. Spaces may sometimes be inserted between words and even between a stem word and the associated post-position. Therefore, to find the stem word in the Myanmar text, it is needed to cut the sentence

into word segments. [30] There are eight Part-of-Speech classes for all Myanmar words. These are Noun, Verb, Adjective, Adverb, Conjunction, Postpositional Marker, Particles, and Interjection.

3.2.2.1 Noun

Noun is the content word that can be used to refer to a person, place, thing. Noun is the stem word in a sentence. "**နိုင်ငံရေးသမား**", "**ဘောလုံးသမား**", "**ရေအိုးစင်**" are nouns. But Noun in Myanmar language can be combined with particles to form plural by suffixing the particle "**တွေ**" [-twei] or "**များ**" [-myar] e.g., "**ကျောင်းသားများ**" is the plural form of Noun "Student". Moreover, noun can suffix with "**ကြီး**", "**တိုင်း**", etc. for example, "**အသင်းကြီး**", "**နေရာတိုင်း**". In the word "**မြို့နှင့်ဆိုင်သော**", the stem word is "**မြို့**" and "**နှင့်ဆိုင်သော**" is the suffix. For the words, "**ဝင်လာသူ**" and "**ဘောင်းဘီဝတ်သူ**" are also nouns. Some of the noun has prefixed, for example, "**အပြုံး**". The word "**အ**" is prefix and the word "**ပြုံး**" means [smile].

There are four type of nouns: they are proper noun, abstract noun, common noun, and collective noun.

1. Proper noun – person name, city name, organization name, country name, etc. "**ဦးမြင့်ဦး**", "**မစ္စအစ်ဂ်ဘာ**", "**ရန်ကုန်မြို့**". Proper noun cannot combine with suffix or prefix. But, some of the proper noun has suffix. For example, "**တရုတ်များ**" means many [Chineses]. "**တရုတ်**" [Chinese], that combined with "**များ**" [-myar]. So, "**တရုတ်**" is tagged with named entity in this case, "**များ**" is a suffix because proper noun can combine with suffix or prefix. In the case of "**ဝါဘိုသူလေး**" [-WarBo Lady], "**ဝါဘို**" [WarBo] is the township name. "**ဝါဘိုသူလေး**" is the girl in that township. So, it is the compound word, it cannot separate as "**ဝါဘို**" [-WarBo] and "**သူလေး**" [Lady]. But, in this case, "**သူ**" [-thu] that index the girl and "**လေး**" is the suffix.

2. Abstract noun - refers to something with which a person cannot physically interact: it is just a feeling "**အချစ်**" [love], "**အမုန်း**" [hate], "**သတ္တိ**" [courage].
3. Common noun – words referred to name generic items instead of specific ones. People in general are named using common nouns. "**ကျောင်း**" [school], "**လမ်း**" [street], "**ဆရာမ**" [teacher], "**စာအုပ်**" [book].
4. Collective noun – words referred to single things that are combined group of people, thing, class or place. "**မိသားစု**" [family], "**ကုမ္ပဏီ**" [company], "**ဌာန**" [department].

Nouns by four types of constitution

- Original and indivisible noun – Example: "**စာအုပ်**" [book], "**ခဲတံ**" [school], "**မိသားစု**" [family].
- Compound noun – Example: "**ရေအိုး"** = (**ရေ**+**အိုး**) [water pot], "**လက်ကိုင်အိတ်**" = (**လက်**+**ကိုင်**+**အိတ်**) [hang bag].
- Verb modification noun – Example "**စားစရာ**" = (**စား**+**စရာ**) [food] "**စား**" [eat] verb combine with particle "**စရာ**", "**သောက်စရာ**" = (**သောက်**+**စရာ**) [drink].
- Qualitative noun – Example "**ကောင်းမှု**" [merit] = "**ကောင်း** (good) + "**မှု**" (particle)", "**လှပမှု**" [beauty] = "**လှပ**" [beauty]+ "**မှု**" (particle)

3.2.2.2 Verb

Stem verbs are always suffixed with at least on particle to form a tense, politeness, mood, etc. The stem of the Verbs remains unchanged when they have the particle suffix to them. These suffixes are "**ရ**" "**တွေ့မြင်ရ**", "**တွေ့မြင်**" is stem word and "**ရ**" is suffix, "**နေ**" "**ရပ်တည်နေ**", "**မယ်**" "**စားမယ်**", "**ခဲ့**" "**ပေါ်ထွက်ခဲ့**", "**နေမည့်**" "**ကြိုးစားနေမည့်**", "**ရာ**" "**လုပ်ဆောင်ရာ**" etc. For instance, "**ဖွံ့ဖြိုးဆဲ**" [developing];

“ဖွံ့ဖြိုးပြီး” [developed]; have different verb particles. But they have the same stem verb is “ဖွံ့ဖြိုး” [develop]. And, Verbs are negated by the particle “မ”[-ma], which is prefix to the verbs to form the negative verb. And then some verbs also negated by particle and also have suffix but which also unchanged the meaning of the stem verb. These verbs are between the particle “မ” and “ဘဲ”, “မ” and “ခင်” etc. For example, “မပြုလုပ်ဘဲ” [not do], the stem verb is “ပြုလုပ်” [do], “မရှိရုံ” [not have], the stem verb is “ရှိ” [have] etc. Some of the verbs are segmented as a passive meaning, for example, “ကုသမှုခံ” is the stem word. Some of the verb, negative meaning “မ” is not used as a prefix of the stem word. This verb is negated with “မ” in the middle of the verb. For example, “ခွင့်မပြု” [not allowed], “လက်မှတ်မထိုးခဲ့” [not signed], “သဘောမတူ” [not agreed]. This kind of verb cannot separate as “ခွင့်” “မ” “ပြု” and the whole word becomes negative verb. In question verb, the markers “သလား” “ရောက်ခဲ့သလား”, “မလား” “သွားမလား” is used. These questions verb “ရောက်ခဲ့သလား” [Have you been?] and “သွားမလား”.

3.2.2.3 Adjective

Adjective is used to modify the noun. Myanmar adjectives can be formed by combining verb and particles. For example, ‘ပေါ်ထွက်လာခဲ့သော’[appeared] is the adjective that combines the verb “ပေါ်ထွက်” [appear] and adjective suffix “လာခဲ့သော”. The adjective word “အနိုင်ရခဲ့သော” [won] that also combines the verb “အနိုင်ရ” [win] and adjective suffix “ခဲ့သော”. Some of the adjective are combined with “အ” “ဆုံး”, for example, “အကြီးဆုံး” [the largest] and “အလှဆုံး” [the most beautiful].

3.2.2.4 Adverb

A word that modifies verb is adverb. Myanmar adverbs are always before verb and there can be more than one adverb for one verb. Adverb also has suffix “စွာ” [-

swar] “**ကျန်းမာစွာ**” [healthily]. Their stem form remains unchanged when suffix removal. And then, Reduplication occurs in Myanmar sentences and most of the reduplicated words are Adverbs and their stem forms are Adjectives. Many Myanmar words, especially adjectives or verbs with two syllables, such as “**ရိုးသား**” [honest] (verb) **“ခိုင်မာ”** [firm](verb)**,** can be reduplicated as **“ရိုးရိုးသားသား”** [honestly](adverb) or **“ခိုင်ခိုင်မာမာ”** [firmly](adverb).

Some of the adverb are between “**တ**” “**တ**” “**တညီတညွတ်**”, “**တရင်းတနှီး**”, “**အ**” “**တ**” “**အကျွမ်းတဝင်**”, “**အရောတဝင်**”, “**အ**” “**အ**” “**အပေးအယူ**”, “**အသွားအလာ**”, “**မ**” “**မ**” “**မဆိုင်းမတွ**”, etc. This kind of adverb “**အကျွမ်းတဝင်**” is stem word, “**အ**” and “**တ**” are not stem words but these words cannot separate and it is defined as a compound word. The word “**တညီတညွတ်**” is also a stem word. In some cases, adverb is combined with “**တ**” and reduplicate the stem word, for example, “**တငြိမ့်ငြိမ့်**” has the prefix word “**တ**” and reduplicate the main stem word “**ငြိမ့်**” and “**တရွေ့ရွေ့**” the main stem word is “**ရွေ့**”.

3.2.2.5 Postpositional Marker

Post-positional markers are words suffixed to a noun, pronoun to designate it as the subject, object and to a verb to indicate time, mood, etc. [Link6] There are two types of post-positional marker: Noun marker and Verb marker. Noun markers “**ကို**”, “**အား**”, “**တိုင်တိုင်**”, “**ထက်တိုင်**”, “**နှင့်အညီ**”, “**အနက်**” are used to indicate time, mood, object, subject, etc. These markers are single words because they are not meaningful words or suffix or prefix of the stem word, for instance, the word “**၃ရက်တိုင်တိုင်**” “**၃**” is numerical number “**ရက်**” [day] means day and then “**တိုင်တိုင်**” is the postpositional marker and “**ဥပဒေနှင့်အညီ**” [abiding the law] “**ဥပဒေ**” [law] is the stem word and “**နှင့်အညီ**” is postpositional marker. “**ဘယ်**”, “**ဘာ**”, “**ဒီ**” are also postpositional markers, for example, “**ဒီနှစ်**” [this year], “**ဘယ်အိမ်**” [which house?] and “**ဘာအလုပ်လုပ်လဲ**” [what do you do?], etc.

3.2.2.6 Particles

Particles are words serving to qualify a noun, pronoun, adjective, verb, and adverb. Some of the particles are "**ရက်**" "**လုပ်ရက်**", "**တော့**" "**သွားတော့**", "**ဖို့**", "**ခွင့်ပြုဖို့**", "**ကုန်**" "**နေကြကုန်**", "**သာ**" "**လုပ်သာလုပ်**", "**သရွေ့**", etc. These particles are suffixes to stem words. Their stem form remains unchanged when suffix is removed. Some of the particles are used as type classifiers, for example, "**၃၉ယောက်**" [39 persons], "**တစ်ဂိုး**" [one goal], "**တစ်ရွက်**" [a piece of paper], "**တစ်ခု**" [one piece], etc. In this case, some types of classifiers are particle to the noun for instance "**၃၉ယောက်**" means 39 people, "**၃၉**" is numerical number and "**ယောက်**" is suffix. But, some type of classifiers is not particle that are relating to noun, they are also noun for instances, "**၂၀၁၉ခုနှစ်**", "**ခုနှစ်**" [year] is the noun. Some particles are numerical modifiers, for examples, "**ခြောက်ကြိမ်မြောက်**", "**နှစ်ဆ**", etc. In these words, "**ကြိမ်မြောက်**", "**ဆ**", are numerical modifier particles that added to the number.

If particles are used after the verb words, they serve as suffixes to verbs and they classify the tenses of the verb and what types of sentences are statements or questions. Verb markers can show the tense of the verb.

Past tense – "**ခဲ့သည်**", "**ခဲ့ပြီ**" = "**ကျောင်းသွားခဲ့သည်**" "**ထမင်းစားခဲ့ပြီ**"

Present tense – "**သည်**" "**ပြီ**" = "**ကျောင်းသွားသည်**" "**ထမင်းစားပြီ**"

Future tense – "**မည်**" "**လိမ့်မည်**" = "**ကျောင်းသွားမည်**" "**ထမင်းစားလိမ့်မည်**"

3.2.2.7 Conjunction

Myanmar conjunction is used to connect words, phrases or clauses. Some conjunctions are "**မရှိရုံ သာမက**" [not only just not have], "**ကဲ့သို့သော**" [as], "**သောအခါ**" [when], "**ထို့ပြင်**" [Moreover], "**သို့မဟုတ်**" [or], etc. In segmentation, conjunction will be segmented as a single word.

3.2.2.8 Interjection

Interjections express sudden emotions which may find utterance. The expression of feeling is one of those: admiration, delight, dislike, angry or desire, etc. Interjections are more frequently used in Myanmar language. For example, "**အလိုလေး**", "**အောင်မလေး**", "**ဘုရားဘုရား**", etc. If the interjections are found, they are segmented as a whole word.

3.3 Syllable Tagging Scheme

The task of joint word segmentation and stemming is to assign word type label to every syllable in a sentence. A single word could span several syllables within a sentence. In order to indicate the word boundaries, BIO format is represented where every syllable is labelled as B-label if the syllable is the beginning of a word, I-label if it is inside a word but not the first token within the word, or O otherwise.

The sentence is first segmented into syllable. Then, from the output, syllable boundary tagging is used to classify the word type and detect the boundary of words. In segmentation, words are segmented as stem word, for example,

Table 3.1 Example of Segmented words and Tagged words

No	Words	Segmented words	Tagged words
1.	**ကျေးရွာသူကျေးရွာသားတွေ**	"**ကျေးရွာသူ**" "**ကျေးရွာသားတွေ**"	"**ကျေး**/B-R **ရွာ**/I-R **သူ**/I-R **ကျေး**/B-R **ရွာ**/I-R **သား**/I-R **တွေ**/B-Suf"
2.	"**လက်တွေ့စမ်းသပ်ခန်း**"	"**လက်တွေ့**" "**စမ်းသပ်ခန်း**"	"**လက်**/B-R **တွေ့**/I-R **စမ်း**/B-R **သပ်**/I-R **ခန်း**/I-R"
3.	"**အလယ်ဗဟို**"	"**အလယ်**" "**ဗဟို**"	"**အ/B-R လယ်/I-R ဗ/B-R ဟို/I-R**"
4.	"**ပြည်ပပို့ကုန်**"	"**ပြည်ပ**" "**ပို့ကုန်**"	"**ပြည်/B-R ပ/I-R ပို့/B-R ကုန်/I-R**"
5.	"**သတ်ဖြတ်သူ**"	"**သတ်ဖြတ်သူ**"	"**သတ်/B-R ဖြတ်/I-R သူ/I-R**"

“ကျေးရွာသူကျေးရွာသားတွေ” [villagers] is segmented as “ကျေးရွာသူ” “ကျေးရွာသားတွေ”, “လက်တွေ့စမ်းသပ်ခန်း” [laboratory] is segmented as “လက်တွေ့” “စမ်းသပ်ခန်း”, “အလယ်ဗဟို” [center] is segmented as “အလယ်” “ဗဟို”, “ပြည်ပပို့ကုန်” [export] is segmented as “ပြည်ပ” “ပို့ကုန်”. For example, “ကျေး/B-R ရွာ/I-R သူ/I-R ကျေး/B-R ရွာ/I-R သား/I-R တွေ/B-Suf”, “လက်/B-R တွေ့/I-R စမ်း/B-R သပ်/I-R ခန်း/I-R”.

But in this case, “သတ်ဖြတ်သူ” [killer], this word is not separate as “သတ်ဖြတ်” and “သူ”. If we separate as “သတ်ဖြတ်” and “သူ”. There will be meaningless. So, it is tagged as one word. For example, “သတ်/B-R ဖြတ်/I-R သူ/I-R”

For stemming, each syllable is tagged with one of five word types.

Table 3.2 Syllable Tag Sets

No	Description	Tag Name	Example
1.	Root Word	R	**စာ/B-R အုပ်/I-R**
2.	Single Word	S	**မှာ/B-S တွင်/I-S, သို့/B-S**
3.	Prefix	Pre	**မ/B-Pre , ဒေါ်/B-Pre**
4.	Suffix	Suf	**ခဲ့/B-Suf သည်/I-Suf**
5.	Named Entity	NE	**ရ/B-NE တ/I-NE နာ/I-NE ဦး/I-NE**

3.3.1 Root words

Root words can be Noun, Verb, Adjective and Adverb but it is a common form of word without any suffix or prefix. For example, in noun word “ကျောင်းသားများ” [students] the root word is “ကျောင်းသား” [student] and “များ” is suffix. So the word “ကျောင်းသားများ” [students] is tagged as “**ကျောင်း/B-R သား/I-R များ/B-Suf**”. In verb “ဖွံ့ဖြိုးခဲ့သည်” [developed] the root word is “ဖွံ့ဖြိုး” [develop] and “ခဲ့” and “သည်” are suffixes. So the word “ဖွံ့ဖြိုးခဲ့သည်” [developed] is tagged as “**ဖွံ့/B-R ဖြိုး/I-R ခဲ့/B-Suf သည်/I-Suf**”. In adjective “လှပသော” [beautiful] the root word is “လှပ” [beauty]

and "သော" is suffix. So, the word "**လှပသော**" [beautiful] is tagged as "**လှ/B-R ပ/I-R သော/B-Suf**" and the word "**လှသောမိန်းကလေး**" [beautiful girl] is divided into "**လှသော**" and "**မိန်းကလေး**", it is tagged as "**လှ/B-R သော/B-Suf မိန်း/B-R က/I-R လေး/I-R**". Moreover, the word "**အလှဆုံး**" is tagged as "**အ/B-Pre လှ/B-R ဆုံး/B-Suf**". The root word is "**လှ**" and "**အ**" and "**ဆုံး**" are prefix and suffix of a word. In adverb **"ခိုင်မာစွာ" [**firmly**]** the root word is **"ခိုင်မာ" [**firm**]** and **"စွာ"** is suffix**.** So, the word "**ခိုင်မာ စွာ**" is tagged as "**ခိုင်/B-R မာ/I-R စွာ/B-Suf**". In duplicate word "**ယုံယုံကြည်ကြည်**", the root word is "**ယုံကြည်**" and the word "**ယုံ**" and "**ကြည်**" are prefix and suffix of the root word. The word "**ယုံယုံကြည်ကြည်**" is tagged as "**ယုံ/B-Pre ယုံ/B-R ကြည်/I-R ကြည်/B-Suf**". The word "**မိဘနှင့်အတူ**" [with parents] "**မိဘ**" is the noun and "**နှင့်အတူ**" is the postpositional marker. So, it is tagged as "**မိ/B-R ဘ/I-R နှင့်/B-S အ/I-S တူ/I-S**". In this case, the reduplication word, "**အပြည်ပြည်ဆိုင်ရာ**" [international] is assigned as a root word "**အ/B-R ပြည်/I-R ပြည်/I-R ဆိုင်/I-R ရာ/I-R**".

Table 3.3 Example of Segmented words and Tagged words for Root Word

No	Words	Segmented words	Tagged words
1.	"**ကျောင်းသားများ**"	"**ကျောင်းသားများ**"	"**ကျောင်း/B-R သား/I-R များ/B-Suf**"
2.	"**ဖွံ့ဖြိုးခဲ့သည်**"	"**ဖွံ့ဖြိုးခဲ့သည်**"	"**ဖွံ့/B-R ဖြိုး/I-R ခဲ့/B-Suf သည်/I-Suf**"
3.	"**လှပသော**"	"**လှပသော**"	"**လှ/B-R ပ/I-R သော/B-Suf**"
4.	"**လှသောမိန်းကလေး**"	"**လှသော**" "**မိန်းကလေး**"	"**လှ/B-R သော/B-Suf မိန်း/B-R က/I-R လေး/I-R**"
5.	"**အလှဆုံး**"	"**အလှဆုံး**"	"**အ/B-Pre လှ/B-R ဆုံး/B-Suf**"
6.	**"ခိုင်မာစွာ"**	**"ခိုင်မာစွာ"**	"**ခိုင်/B-R မာ/I-R စွာ/B-Suf**"
7.	"**ယုံယုံကြည်ကြည်**"	"**ယုံယုံကြည်ကြည်**"	"**ယုံ/B-Pre ယုံ/B-R ကြည်/I-R ကြည်/B-Suf**"

8.	“**မိဘနှင့်အတူ**”	“**မိဘ**” “**နှင့်အတူ**”	“**မိ/B-R ဘ/I-R နှင့်/B-S အ/I-S တူ/I-S**”
9.	“**အပြည်ပြည်ဆိုင်ရာ**”	“**အပြည်ပြည်ဆိုင်ရာ**”	“**အ/B-R ပြည်/I-R ပြည်/I-R ဆိုင်/I-R ရာ/I-R**”

3.3.2 Simple words

Simple words are Conjunction, Interjection, and Post Postpositional Marker. Like stop word, these words appear so frequently that their usefulness is limited. In Information Retrieval ignores stop words at the time of searching a user query. These single words are tagged with S. “**ထို့ကြောင့်**” is tagged as “**ထို့/B-S ကြောင့်/I-S**”, “**အလို၌**” is tagged as “**အ/B-S လို/I-S ၌/I-S**”, “**ကြောင့်**” is tagged as “**ကြောင့်/B-S**”, “**သဖြင့်**” is tagged as “**သ/B-S ဖြင့်/I-S**”. Interjection word “**အလိုလေး**”, “**အောင်မလေး**” also assign as a single word, “**အ**/B-S **လို**/I-S **လေး**/I-S”, “**အောင်**/B-S **မ**/I-S **လေး**/I-S” because these word are not meaningful word. Moreover, some of the adverb is assigned as a single word “**အလွန်**”, “**နည်းနည်း**”, “**လုံးဝ**”. For instance: “**အလွန်ကောင်း**” [very good] is an adverb but we assign “**ကောင်း**” [good] as a root word and “**အလွန်**” [very] is assigned as a single word. This kind of adverb is denoted as a stop word.

And, time indicator adverb such as “**မကြာခဏ**”, “**အမြဲတစေ**”, “**ဘယ်တော့မှ**” is assigned as a single word “**မ/B-S ကြာ/I-S ခ/I-S ဏ/I-S**”, “**အ/B-S မြဲ/I-S တ/I-S စေ/I-S**”, “**ဘယ်/B-S တော့/I-S မှ/I-S**”. Furthermore, comparative adjective “**ထက်ပို**” [more than], “**သာ၍**” [being more] are assigned as a single word “**ထက်/B-S ပို/I-S**” “**သာ/B-S ၍/I-S**”. For instance: “**သာ၍ ကောင်းသော**” is tagged as “**သာ/B-S ၍/I-S ကောင်း/B-R သော/B-Suf**”.

Table 3.4 Example of Segmented words and Tagged words for Single Word

No	Words	Segmented words	Tagged words
1.	“**ထို့ကြောင့်**”	“**ထို့ကြောင့်**”	“**ထို့/B-S ကြောင့်/I-S**”
2.	“**အလို၌**”	“**အလို၌**”	“**အ/B-S လို/I-S ၌/I-S**”
3.	“**ကြောင့်**”	“**ကြောင့်**”	“**ကြောင့်/B-S**”

4.	“သဖြင့်”	“သဖြင့်”	“သ/B-S ဖြင့်/I-S”
5.	“အလိုလေး”	“အလိုလေး”	“အ/B-S လို/I-S လေး/I-S”
6.	“အောင်မလေး”	“အောင်မလေး”	“အောင်/B-S မ/I-S လေး/I-S”
7.	“မကြာခဏ”	“မကြာခဏ”	“မ/B-S ကြာ/I-S ခ/I-S ဏ/I-S”
8.	“အမြဲတစေ”	“အမြဲတစေ”	“အ/B-S မြဲ/I-S တ/I-S စေ/I-S”
9.	“ဘယ်တော့မှ”	“ဘယ်တော့မှ”	“ဘယ်/B-S တော့/I-S မှ/I-S”
10.	“ထက်ပို”	“ထက်ပို”	“ထက်/B-S ပို/I-S”
11.	“သာ၍”	“သာ၍”	“သာ/B-S ၍/I-S”
12.	“သာ၍ ကောင်းသော”	“သာ၍” “ကောင်းသော”	“သာ/B-S ၍/I-S ကောင်း/B-R သော/B-Suf”

3.3.3 Prefix

Verbs are negated by the particle “**မ**” [-ma], which is a prefix to the verbs to form the negative verb and which also unchanged the root of verb. “**အ**” [-a] by affixing as “**လုပ်**” [work] can change verb form to noun “**အလုပ်**” [job] without changing the meaning. The word “**အလုပ်**” [job] is tagged as “**အ**/B-Pre **လုပ်**/B-R”. In name entity, “**ဦး**” [U], “**ဒေါ်**” [Daw], “**ကို**” [Ko], “**မ**” [Ma] are tagged as a prefix. For example: in the name “**ဦးသန့်**” [U Thant], the actual name is only “**သန့်**” [Thant] and “**ဦး**” [U] is the prefix of the word and it is used to separate that the person is male or female. For female: “**ဒေါ်**” [Daw] is used as a prefix. For example, “**ဒေါ်အောင်ဆန်းစုကြည်**” [Daw Aung San Su Kyi], the actual name is “**အောင်ဆန်းစုကြည်**” [Aung San Su Kyi] and “**ဒေါ်**” [Daw] is the prefix for female. The word “**ကို**” [Ko] and “**မ**” [Ma] also used as a prefix for young boy and girl.

Table 3.5 Example of Segmented words and Tagged words for Prefix

No	Words	Segmented words	Tagged words
1.	“**အလုပ်**”	“**အလုပ်**”	“**အ**/B-Pre **လုပ်**/B-R”
2.	“**ဦးသန့်**”	“**ဦးသန့်**”	“**ဦး/B-Pre သန့်/B-NE**”
3.	“**ဒေါ်အောင်ဆန်းစုကြည်**”	“**ဒေါ်အောင်ဆန်းစုကြည်**”	“**ဒေါ်/B-Pre အောင်/B-NE ဆန်း/I-NE စု/I-NE ကြည်/I-NE**”

3.3.4 Suffix

Most of the Suffix words are Particle that derive a new word form by attaching to the root word but their stem form remains unchanged when suffix removal. In Myanmar language, there are many derivational morphemes that change verbs to nouns, verb to adverb by adding suffix to the root word. “**များ**”, “**တို့**”, “**သော**”, “**သည့်**”, “**မည့်**”, “**ဖို့**”, “**ရန်**” are particles and they are assigned as the suffixes.

ဆရာများ = “**ဆရာ**” [teacher] (root) + “**များ**” [particle] (suffix)

ကလေးတို့ = “**ကလေး**” [child] (root) + “**တို့**” [particle] (suffix)

ကြင်နာသော = “**ကြင်နာ**” [kind] (root) + “**သော**” [particle] (suffix)

စာရေးသည့် = “**စာရေး**” [write] (root) + “**သည့်**” [particle] (suffix)

နေထိုင်မည့် = “**နေထိုင်**” [live] (root) + “**မည့်**” [particle] (suffix)

သွားဖို့ = “**သွား**” [go] (root) + “**ဖို့**” [particle](suffix)

စားသောက်ရန် = “**စားသောက်**” [eat] (root) + “**ရန်**” [particle] (suffix)

Some of the Myanmar verb has different form of suffix. But they have the same meaning.

Table 3.2 Example of Suffix

Myanmar Verb	Main Verb	Suffixes	English meaning
ပေါ်ထွက်လာခဲ့သော	**ပေါ်ထွက်**	**လာခဲ့သော**	appeared
ပေါ်ထွက်ခဲ့သော	**ပေါ်ထွက်**	**ခဲ့သော**	appeared

ပေါ်ထွက်သည်	ပေါ်ထွက်	သည်	appear
ဦးတည်ထားရှိကြောင်း	ဦးတည်	ထားရှိကြောင်း	head to
ဦးတည်ထားရှိ	ဦးတည်	ထားရှိ	head to
ဦးတည်ထား	ဦးတည်	ထား	head to

For example, "**ပေါ်ထွက်လာခဲ့သော**" [appeared] is the adjective that combine the verb "**ပေါ်ထွက်**" [appear] and adjective suffix "**လာခဲ့သော**". The word "**ပေါ်ထွက်လာခဲ့သော**" [appeared] is tagged as "ပေါ်/B-R **ထွက်**/I-R **လာ**/B-Suf **ခဲ့**/I-Suf **သော**/I-Suf". Moreover, a word that modifies verb is an adverb. Myanmar adverbs are always before verb and there can be more than one adverb for one verb. Adverb also has suffix "**စွာ**" [-swar]. Their stem form remains unchanged when suffix is remove. The word "**ယဉ်ကျေးစွာ**" [politely], root word is "**ယဉ်ကျေး**" [polite] and "**စွာ**" is suffix.

3.3.5 Named Entity

For the person name, "**ဦးမြင့်ဦး**" [U Myint Oo], "**ဦး**" [U] is the prefix of name "**မြင့်ဦး**" [Myint Oo]. It means that "**မြင့်ဦး**" [Myint Oo] is the male. In Myanmar Language, prefix "**ဦး**" [U] and "**ဒေါ်**" [Daw], "**ကို**" [Ko] and "**မ**" [Ma] is used to separate male or female. Moreover, it shows that the person name and it cannot separate "**ဦး**" [U] and "**မြင့်ဦး**" [Myint Oo]. It is an exceptional case, so, it can be tagged with "**ဦး/B-Pre မြင့်/B-NE ဦး/I-NE**". In English name, "**မစ္စ အစ်ဒ်ဘာ**" [Mrs. Adbar], "**မစ္စ**" is prefix and "**အစ်ဒ်ဘာ**" [Adbar] is named entity and it is tagged as "**မစ္စ/**B-Pre **အစ်ဒ်/**B-NE **ဘာ/**I-NE".

Proper name can also exist in front of "**အဖွဲ့**" [organization] "**ကမ္ဘာ့ကျန်းမာရေးအဖွဲ့**" [World Health Organization] "**ကမ္ဘာ့**/B-NE **ကျန်း**/I-NE **မာ**/I-NE **ရေး**/I-NE **အ**/B-R **ဖွဲ့**/I-R", "**ဦးစီးဌာန**" [department] "**ငါးလုပ်ငန်းဦးစီးဌာန**" "**ငါး**/B-NE **လုပ်**/I-NE **ငန်း**/I-

NE ဦး/B-R စီး/I-R ဌာ/I-R န/I-R" [Fisheries Department], "**ဝန်ကြီးဌာန**" [ministry] "**ပညာရေးဝန်ကြီးဌာန**" [Ministry of Education] "**ပ**/B-NE **ညာ**/I-NE **ရေး**/I-NE **ဝန်**/B-R **ကြီး**/I-R **ဌာ**/I-R **န**/I-R", **"မြို့"** [city] "**ရန်ကုန် မြို့**" [Yangon City] is tagged as "**ရန်/**B-NE **ကုန်/**I-NE **မြို့/**B-R", etc. Moreover, some numbers can be name entity, for example, "**၃၃လမ်း**" [33th street]. In this case, 33 is not a number. It is the name of the street "**၃/**B-NE **၃/**I-NE **လမ်း/**B-R".

The word "**ကျိုက်ထီးရိုး ဘုရား**" [Kyite Htee Yoo Pagoda]is the name of the pagoda "**ကျိုက်ထီးရိုး**" [kyite htee yoo] and "**ဘုရား**" [pagoda], it is tagged as "**ကျိုက်**/B-NE **ထီး**/I-NE **ရိုး**/I-NE **ဘု**/B-R **ရား**/I-R". In addition, "**ဆူးလေဘုရားလမ်း**" [Sule Pagoda Road] is the name of the street "**ဆူးလေဘုရား**" [Sule Pagoda] and "**လမ်း**", it is tagged as "**ဆူး**/B-NE **လေ**/I-NE **ဘု**/I-NE **ရား**/I-NE **လမ်း**/B-R". The word "**ဗိုလ်ချုပ်ဈေး**" [Bogyoke Market]is the name market and "**ဗိုလ်ချုပ်**" [Bogyoke] and "**ဈေး**" [market], it is tagged as "**ဗိုလ်**/B-NE **ချုပ်**/I-NE **ဈေး**/B-R".

Table 3.5 Example of Segmented words and Tagged words for Named Entity

No	Words	Segmented words	Tagged words
1.	"**ဦးမြင့်ဦး**"	"**ဦးမြင့်ဦး**"	"**ဦး/B-Pre မြင့်/B-NE ဦး/I-NE**"
2.	"**မစ္စ အစ်ဘာ**"	"**မစ္စ အစ်ဘာ**"	"**မစ္စ/**B-Pre **အစ်/**B-NE **ဘာ/**I-NE"
3.	"**ကမ္ဘာကျန်းမာရေးအဖွဲ့**"	"**ကမ္ဘာကျန်းမာရေး**" "**အဖွဲ့**"	"**ကမ္ဘာ**/B-NE **ကျန်း**/I-NE **မာ**/I-NE **ရေး**/I-NE **အ**/B-R **ဖွဲ့**/I-R"
4.	"**ငါးလုပ်ငန်းဦးစီးဌာန**"	"**ငါးလုပ်ငန်း**" "**ဦးစီးဌာန**"	"**ငါး**/B-NE **လုပ်**/I-NE **ငန်း**/I-NE **ဦး**/B-R **စီး**/I-R **ဌာ**/I-R **န**/I-R"
5.	"**ပညာရေးဝန်ကြီးဌာန**"	"**ပညာရေး**" "**ဝန်ကြီးဌာန**"	"**ပ**/B-NE **ညာ**/I-NE **ရေး**/I-NE **ဝန်**/B-R **ကြီး**/I-R **ဌာ**/I-R **န**/I-R"
6.	"**ရန်ကုန် မြို့**"	"**ရန်ကုန်**" "**မြို့**"	"**ရန်/**B-NE **ကုန်/**I-NE **မြို့/**B-R"

7.	“၃၃လမ်း”	“၃၃” “လမ်း”	“၃/B-NE ၃/I-NE လမ်း/B-R”
8.	“ကျိုက်ထီးရိုး ဘုရား”	“ကျိုက်ထီးရိုး” “ဘုရား”	“ကျိုက်/B-NE ထီး/I-NE ရိုး/I-NE ဘု/B-R ရား/I-R”
9.	“ဆူးလေဘုရားလမ်း”	“ဆူးလေဘုရား” “လမ်း”	“ဆူး/B-NE လေ/I-NE ဘု/I-NE ရား/I-NE လမ်း/B-R”
10.	“ဗိုလ်ချုပ်စျေး”	“ဗိုလ်ချုပ်” “စျေး”	“ဗိုလ်/B-NE ချုပ်/I-NE စျေး/B-R”

3.4 Summary

This chapter has briefly discussed the different types and problems of word segmentation, stemming and named entity detection. In addition, Myanmar word nature and eight parts of speech have been explained for Myanmar language. Moreover, this chapter proposed syllable-based tagging scheme for joint word segmentation, stemming and named entity detection for Myanmar language, and explained the customized tag sets for joint process. The detailed structure of system will be discussed in chapter 5.

CHAPTER 4
THE PROPOSED SYSTEM ARCHITECTURE

There are three parts in this chapter. Firstly, this chapter presents different types of neural network. Secondly, this chapter moves to introduce framework for neural sequence labeling model. The last part describes the system architecture for joint word segmentation, stemming and Named entity detection.

4.1 Basic Architecture of Artificial Neural Network

An Artificial Neural Network (ANN) is modeled on the brain where neurons are connected in complex patterns to process data from the senses, establish memories and control the body. [77] An Artificial Neural Network (ANN) is a system based on the operation of biological neural networks or it is also defined as an emulation of biological neural system. The Neural networks are defined as the systems of interconnected neurons. Neurons or Nerve Cells are the basic building blocks of brains which are the biological neural networks. The architecture of neural network is shown below:

Any deep neural networks typically contain three types of layers [76]:

- The Input Layer
- The Hidden Layer
- The Output Layer

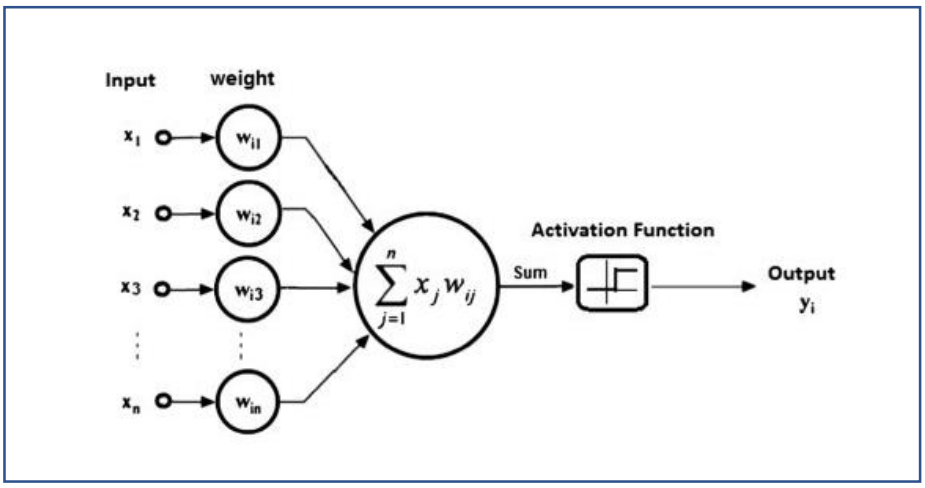

Figure 4.1 Basic Architecture of Deep Neural Network

The input layer of a neural network is composed of artificial input neurons, and takes the initial data to the system for further processing by subsequent layers of

artificial neural network. There is no computation execute in this layer, they just pass the information to the subsequent layer. [78] The input layer passes the data through the activation function before passing it on. The data is then multiplied by the first hidden layer's weights.

A hidden layer in an artificial neural network is a layer in between input layers and output layers, where artificial neuron takes in a set of weighted inputs and produce output through an activation function. [79] It is a typical part of any neural network because it is simulated the types of activity that go on in the human brain. They operate computations and then transfer the weight from the input layer to the next layer (another hidden layer or output layer).

The output layer is the last layer of neurons that produces given outputs for the program. The output layer uses an activation function that maps to the desired output format.

4.1.1 Activation

In a neural network, the input data are fed the input layer of the neuron. Each neuron has weight, and calculating the input data with the weight and which is transmitted to the output layer of the neuron. The activation function is a numerical function between the input layer and output layer. It is used to switch the neuron output on and off calculating on a rule or threshold value. Neural network uses non-linear activation functions to compute complex problems.

4.1.2 Weight

The most significant factor in adapting an input to impact the output is weight. Weights represent numerical parameters which decide how strongly each of the neurons influence on the output.

4.1.3 Bias

Bias is also important factor in deep learning. The activation function in neural network takes an input data by calculating a weight. Bias allows the neuron to shift the activation function by adding a threshold value (i.e. constant) to the input.

4.2 Different Types of Neural Network

In the first section, there are three main types in neural network: Feed Forward Neural Network (FF), Convolutional Neural Network (CNN) and Recurrent neural networks (RNN).

4.2.1 Feed Forward Neural Network

Feedforward neural network is a basic class of artificial neural networks. [45] In a feedforward neural network, the information moves only one direction through the input layer continuously and it reaches the output layer. There is no back propagation but it has a front propagated and all nodes are fully connected. There are two types of feed forward network: single-layer perceptron and multilayer perception.

- Single-layer perceptron (SLP)

  Single layer perceptron is the simplest neural network and it consists of only single hidden layer of output node. [45] In the single-layer perceptron, the sum of the multiplication of the inputs and their weights are calculated and fed to the linear activation function to obtain the output.

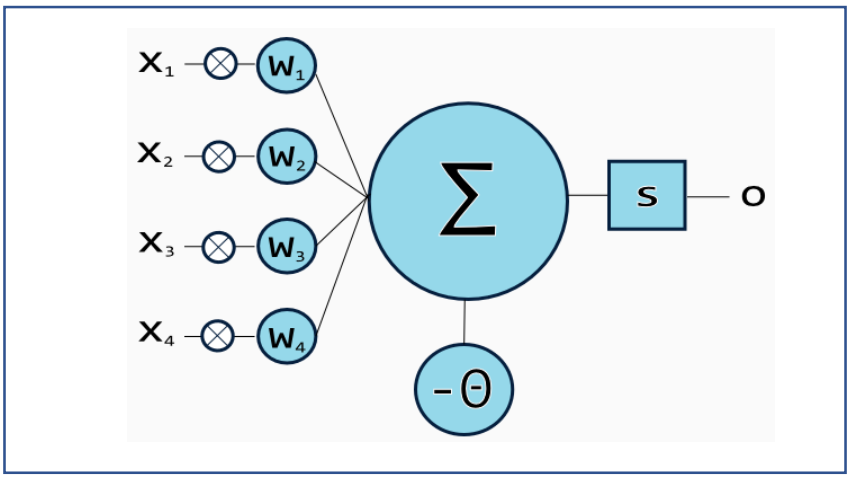

Figure 4.2 Single-layer Perceptron (SLP)

- Multilayer perceptron(MLP)

There are three or more layers contained in multilayer perceptron. It is used to classify data that cannot be separated linearly. It is a fully connected artificial neural network because every single node is connected each other. It utilizes a non-linear

activation function and it is more useful because it solves the problems that are not linearly separable.

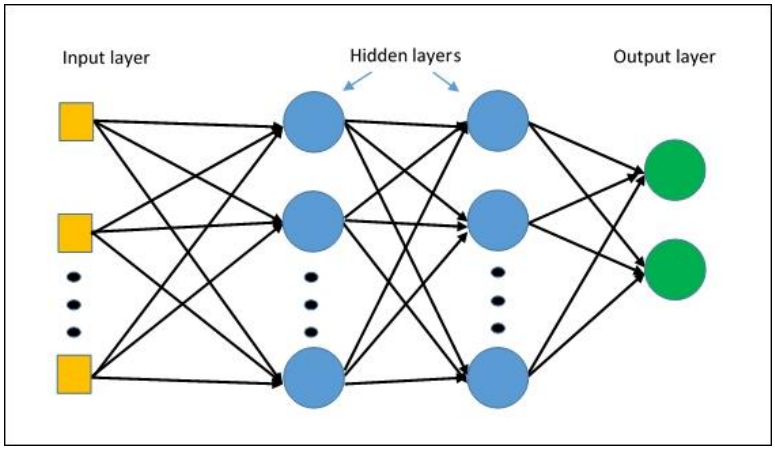

Figure 4.3 Multilayer Perceptron (MLP)

4.2.2 Convolutional Neural Network (CNN)

CNN is a form of deep, feedforward neural networks. CNN uses a system like a multilayer perceptron that has been designed for reduced processing requirements. [80] The input of CNN in NLP processes are sentences or documents that are denoted as a matrix. Each row of the matrix related to one token or a word and it could be a syllable or character. This means that, each word is represented by a vector. Typically, these vectors are word embedding. The layers of a CNN compose of an input layer, an output layer and a hidden layer that includes multiple convolutional layers, pooling layers, fully connected layers and normalization layers.

- Input layer

  Input layer holds images or text document as a vector representation.

- Convolution Layer

  This layer determines the output volume by computing the dot products between set of weights in the input layer. The main goal of a convolutional layer is to observe features [89].

- Activation Function Layer

  This layer accepts the feature map developed by the convolutional layer and apply activation function to the output of the

convolutional layer. The activation function is an element-wise operation that commonly uses RELU over the input data so the dimensions of the input and the output are equivalent [89].

- Pool Layer

  The pooling or downsampling layer is responsible for decreasing the spatial dimensions of the activation maps. [10] In general, they are used as subsequent of convolutional layer in order to cut down the computational supplies progressively through the following layer and that help to minimize the likelihood of overfitting. The key concept of the pooling layer is to provide translational invariance. Therefore, the pooling operation aims to preserve the detected features in a smaller representation and discards the less significant data at the cost of spatial resolution.

- Fully-Connected Layer

  Fully connected layers are similar to the output layer of multilayer perceptron (MLP). This layer aggregates information from the final feature maps and generates final classification. [10] In fully connected layer, all neurons are fully connected to all neurons in the previous layer.

Figure 4.4 shows the typical convolutional neural network.

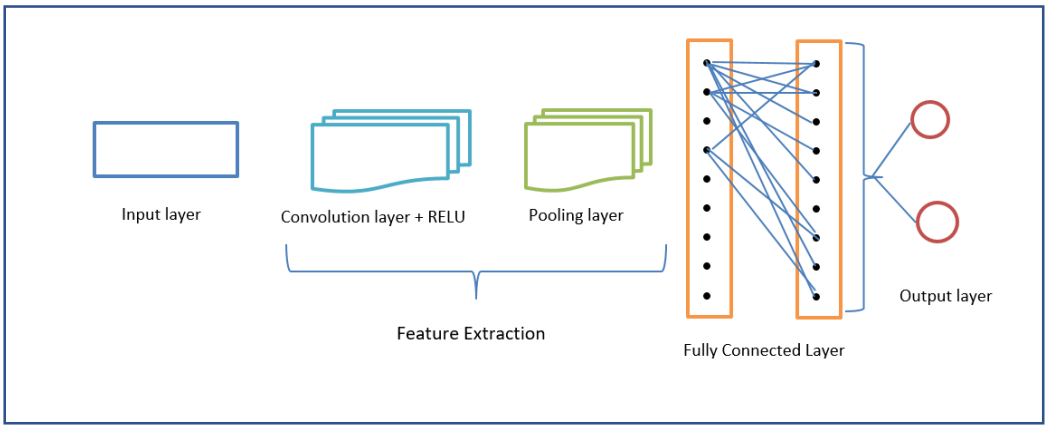

Figure 4.4 Convolutional Neural Network

4.2.3 Recurrent Neural Network (RNN)

In traditional neural networks, all the inputs and outputs do not depend on each other, but in some cases, it is required to predict the next word of a sentence, the previous words are required and therefore it is a need to learn the past words. [81] RNNs are called recurrent because they use the sequential information to perform the same task and output depends on the previous information.

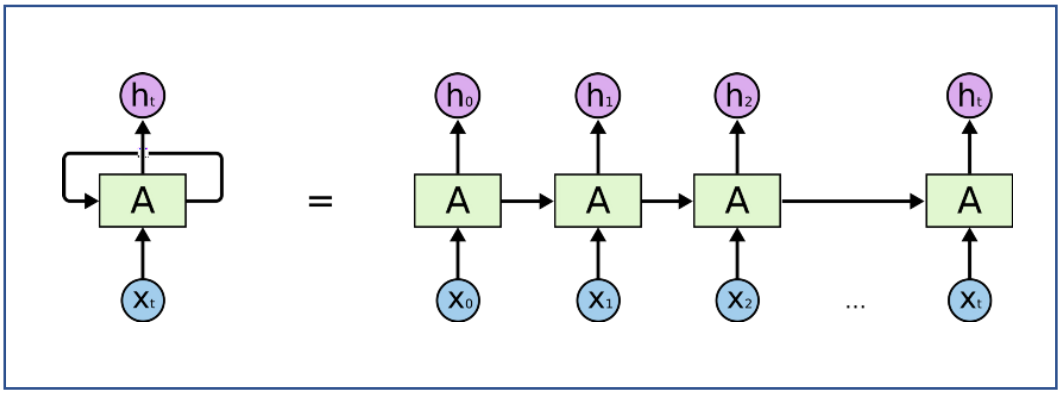

Figure 4.5 Recurrent Neural Network

Figure 4.5 shows the recurrent neural network operates on a sequence that contains vectors x(t) with time step t ranging from 1 to t. [82] It also has a hidden state vector h(t) for each time sequence. Commonly hidden state h(t) is a function f of the previous hidden state h(t-1) and the current input x(t).

Theoretically, RNNs can learn long distance dependencies, still in practice they fail due vanishing/exploding gradients. To solve this problem, they introduced the LSTM RNN.

4.2.3.1 LSTM

Long Short Term Memory networks (LSTM) are a special type of RNN that able to learn long-term dependencies. [5] LSTMs network intend to avoid the long-term dependency problem. It can recognize information for long periods of time. LSTM weights are determined by three operation gates such as input gate, forget gate, and output gate.

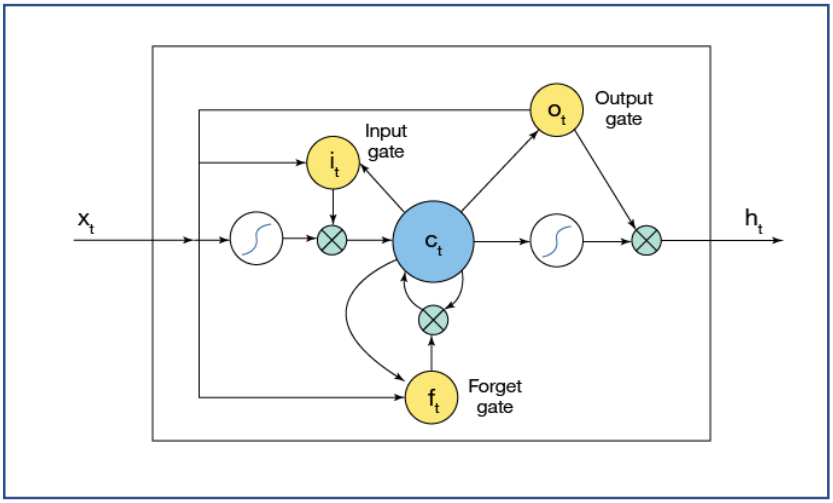

Figure 4.6 Long Short Term Memory (LSTM) Network

Figure 4.6 shows the long short term memory network. Input gate jointly takes the previous output with the new input and passes through another sigmoid layer. This gate determines whether or not to let new input. Forget gate is responsible to delete the information that aren't important. Output gate let impact the output at the current time step.

4.2.3.2 BI-LSTM

Bidirectional LSTM operates on two ways, one from past to future and another from future to past. So, information from past and future contexts are merged.

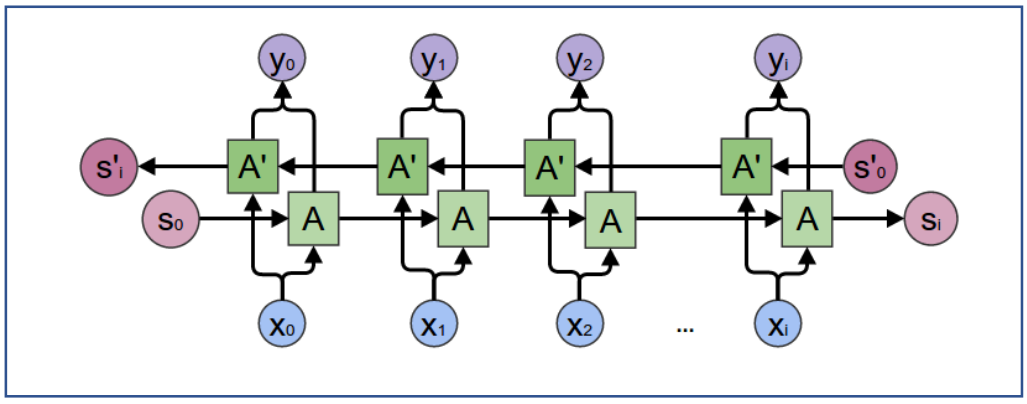

Figure 4.7 Bidirectional Long Short Term Memory Network

Figure 4.7 shows the bidirectional recurrent neural network operate on a sequence that contains input vectors x(i). [32] In a hidden state vector h(t), current input sequence passes through forward and backward and concatenate the left-to-right final state S_0 and the right-to-left final state S'_0.

4.2.3.3 GRU

Gated Recurrent Unit (GRU) is the newer generation of Recurrent Neural Network and similar to an LSTM. [83] GRU can memorize long distance dependencies and it has only two gates; reset gate and update gate.

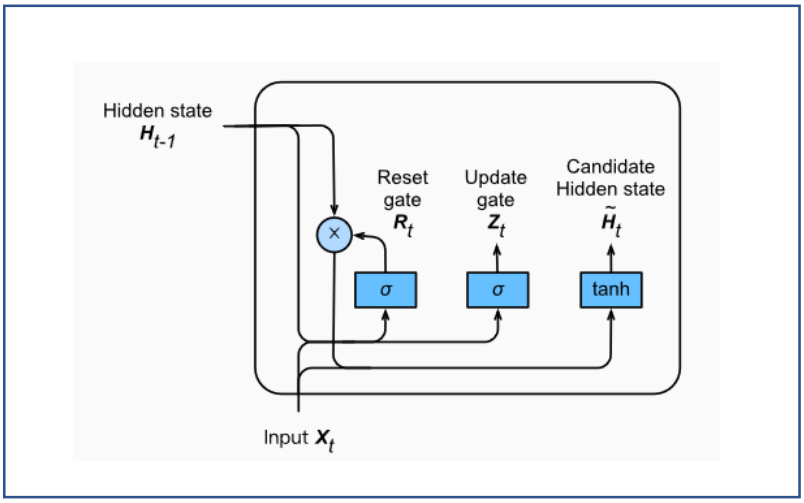

Figure 4.8 Gated Recurrent Unit (GRU) Network

Figure 4.8 shows gate recurrent unit (GRU) network. The update gate acts as similar to the forget and input gate in LSTM. It decides what information from the past discard and what new information to add. The reset gate is used to decide how much pass information to forget.

4.3 Neural Sequence Labeling Model

Most of the NLP processes such as word segmentation, part-of-speech tagging and named entity detection have been improved significantly from the earliest dictionary based approach to CRF approaches with handcrafted features and task-specific resources. With advances in deep learning, neural models have given state-of-the-art results on many sequence labeling tasks. In general, many existing neural

sequence labeling models utilize word-level structure to represent global sequence information inference layer to capture dependencies between neighboring labels.

In this architecture, there are six layers: input layer, character embedding layer, character representation layer, word embedding layer, word representation layer, and inference layer. Figure 4.9 shows the main architecture of neural sequence labeling model.

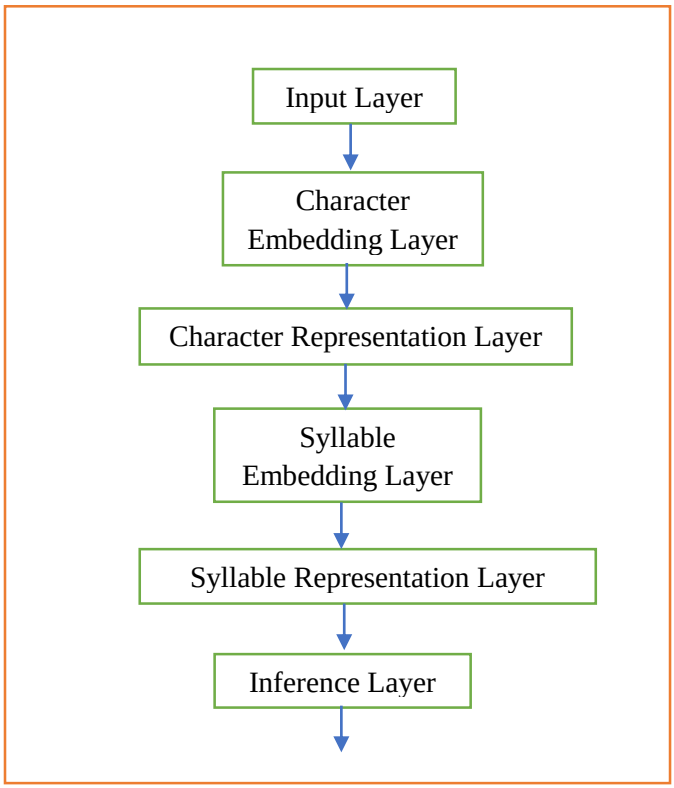

Figure 4.9 The Main Architecture of Neural Sequence Labeling Model

They take as input a sequence of character (c_1, c_2, ..., c_n). The first step to process a sentence by neural architecture is to transform characters into embeddings. This transformation is done by lookup embedding table. A character lookup table M_{char} € $R^{|Vchar| \times d}$ where |Vchar| denotes the size of the character vocabulary and d denotes the dimension of embeddings is associated with all characters. Given a sentence S = (c_1; c_2; ...; c_L), after the lookup table operation, we obtain a matrix X € $R^{L \times d}$ where the i^{th} row is the character embedding of c_i. And character level embedding integrates neural encoder LSTM to encode the character level representation. Character-level information combines with syllable embedding and

feed them into CNN network to model context information of each syllable. The inference layer takes the extracted syllable sequence representations as features and assigns labels to the syllable sequence.

Neural sequence labeling framework contains three layers; a character sequence representation layer, a word sequence representation layer and an inference layer, as shown in Figure 4.10.

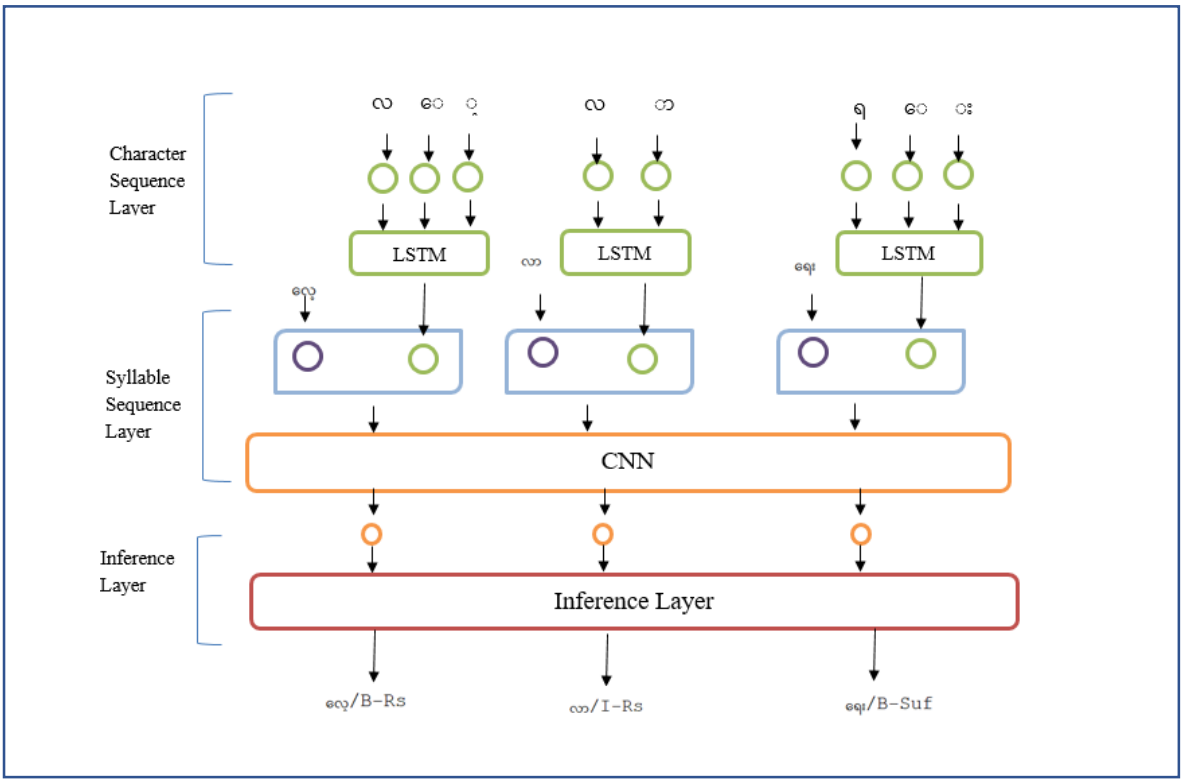

Figure 4.10 Neural Sequence Labeling Architecture for Word "လ့လာရေး". Green, purple, blue and orange represent character embeddings, syllable embeddings, character sequence representations and syllable sequence representations, respectively.

4.3.1 Character Sequence Layer

Character features can be automatically extracted by encoding the character sequence within the syllable. Character sequence layer integrates neural encoders LSTM to encode character-level information of a syllable into its character-level representation. If a character sequence representation layer is used, syllable embeddings and character sequence representations are concatenated for syllable representations.

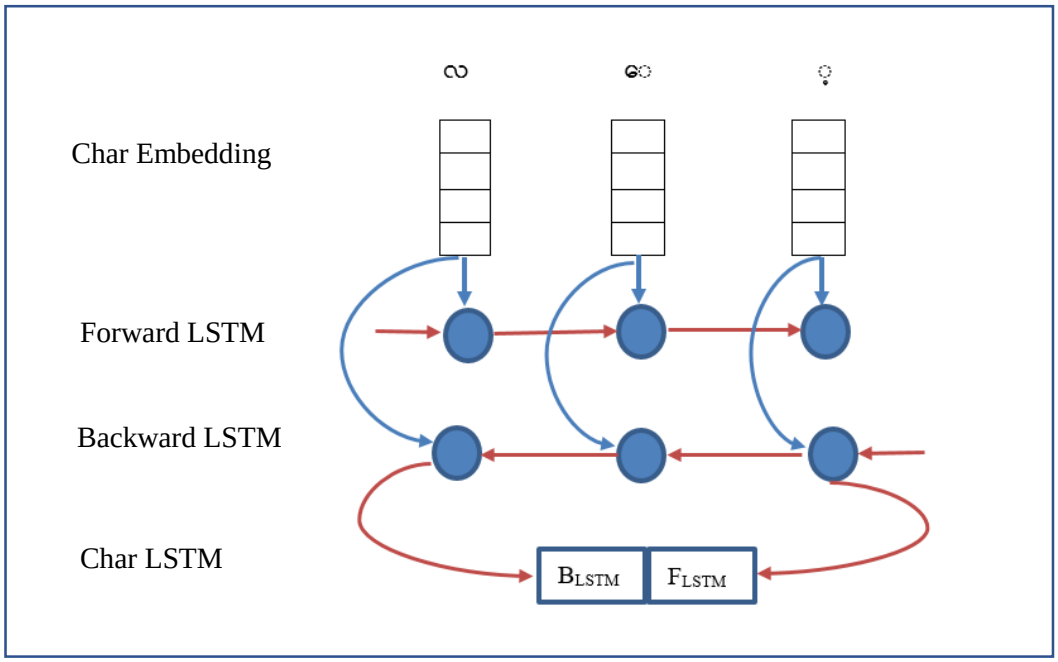

Figure 4.11 Character Long Short Term Memory

The idea consists of augmenting an RNN with memory cells to overcome difficulties with training and efficiently cope with long distance dependencies. Bi-LSTM networks are extensions to single LSTM networks [56]. In order to model the character sequence information of a syllable, Bi-LSTM encodes the character sequence of each syllable and concatenates the left-to-right and right-to-left as character sequence representations.

4.3.2 Syllable Sequence Layer

Character-level information combines with syllable embedding and feed them into CNN networks to model context information of each word. Similar to character sequences, syllable sequence layer can model syllable sequence information through CNN structures.

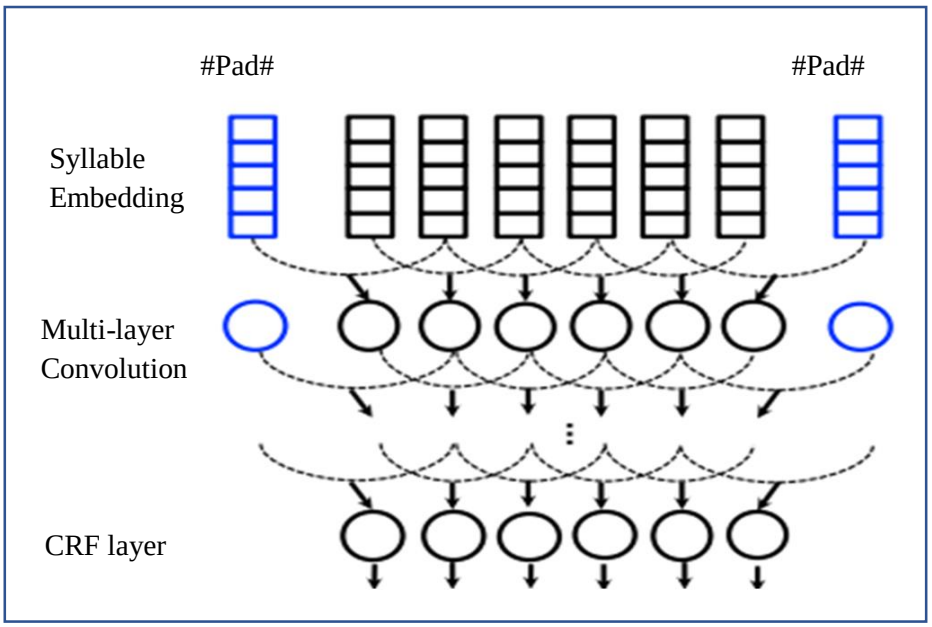

Figure 4.12 Word Convolutional Neural Network

Convolutional Neural Network (CNN) is popular in dealing with images and it has excellent results. In the syllable representation layer, character sequence representation and syllable embedding are concentrated. Convolutional layers are used to model the contextual information of syllable level. Convolutional neural networks (CNNs) have shown its great effectiveness to extract morphological information such as prefix and suffix of a word. Our CNN network is quite simple, multi convolutional layers is used (no pooling layer). For each CNN layer, a window of size 3 slides, extracting local features on the syllable input and rectified linear unit (ReLU) is used in our convolutional layer.

4.3.3 Inference Layer

The inference layer takes the extracted word sequence representations as features and assigns labels to the word sequence. CRF considers the correlations between labels in neighborhoods. For sequence labeling tasks, the interaction between labels in surroundings is considered and jointly decode the best chain of labels for a given input sentence. For example, in this approach of stemming suffix words are absolutely followed by a root word and standard BIO annotation I-R cannot follow I-Suf. Therefore, an inference layer with a linear-chain Conditional Random Field (CRF) is used. This classifier is beneficial for tasks with strong dependencies between token tags.

4.4 Overview of the Proposed System

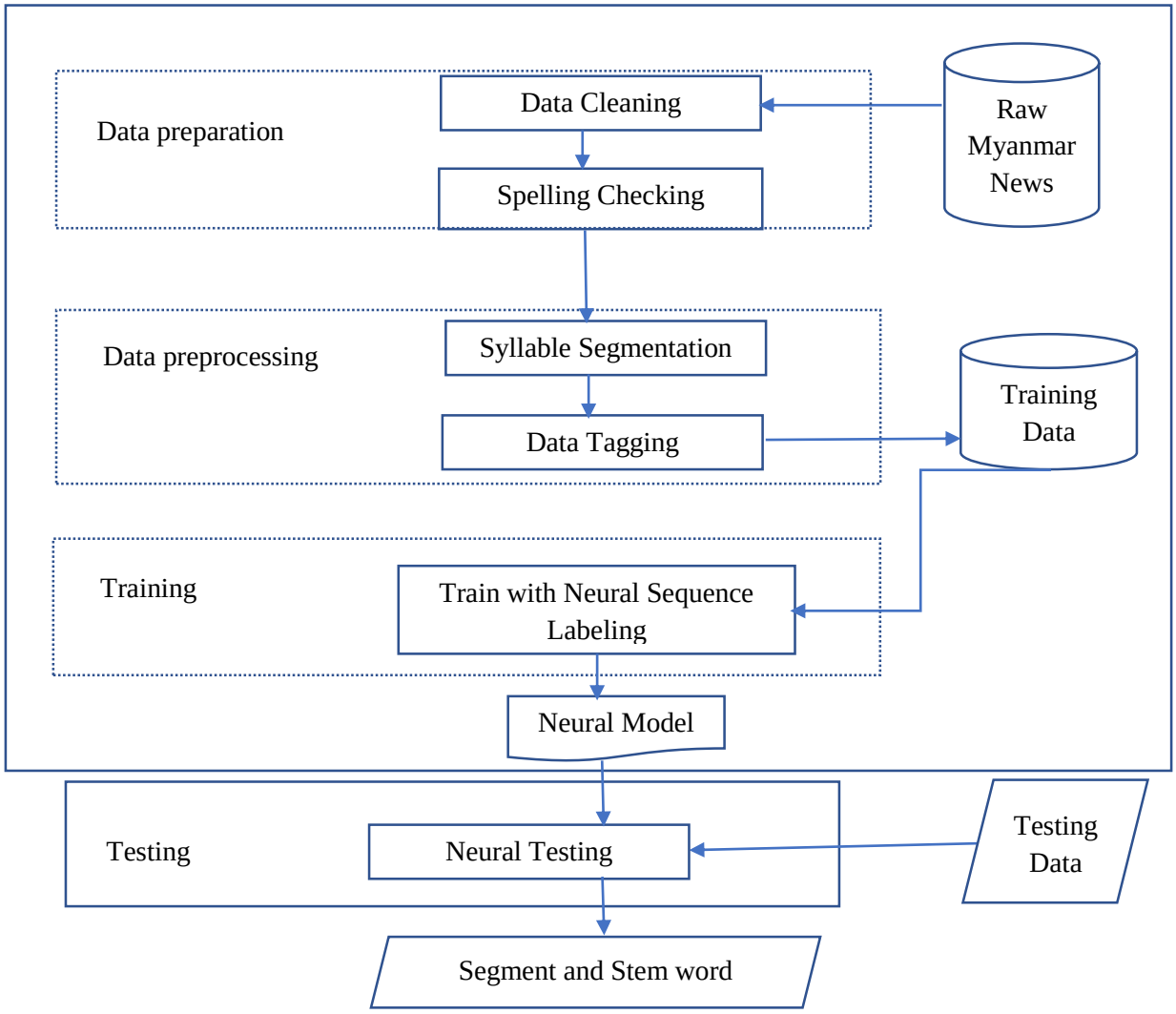

Figure 4.13 Overview of the proposed system

In the system, there are four phases. Data preparation, data preprocessing, training, and testing Phase. Data collection and data cleaning is performed in the Dataset preparation phase. And then, as a preprocessing, sentences are segmented as syllable and manually tag the syllable. Then, neural sequence labeling is trained with tagged data set. Finally, testing data are tested on the neural model. The results are segment and stem word.

4.5 Summary

This chapter presents simple description of the basic architecture of neural network and different kinds of neural networks. Moreover, this chapter explains the details of the state-of-the-art neural sequence labeling structure which is used in proposed system. It also illustrates the main architecture of neural sequence labeling model and explanation of each layer. The implementation of the proposed system is described in chapter 5 with detailed system process and implementation. The performance evaluations are discussed in chapter 6.

CHAPTER 5
IMPLEMENTATION OF THE PROPOSED SYSTEM

This research introduces a Myanmar lexical analysis model that jointly accomplishes three tasks: word segmentation, stemming, and named entity detection. Most East Asian languages including Myanmar are written without explicit word delimiters. Therefore, word segmentation is a preliminary step for processing those languages. Stemming refers to the development of indicating each word in the word segmentation result with an appropriate morphological analysis, e.g. root word, suffix, prefix, etc. Named entity detection, refers to detecting entities that have specific meanings in the identified text, including persons, locations, organization, etc. Normally, segmentation is considered as a separate process from stemming and named entity recognition. In this approach, word segmentation, stemming, and named entity detection are implemented as a joint process. This chapter describes the architecture and detailed description of the proposed joint word segmentation, stemming and named entity detection.

In the system, there are four phases:

i. Data preparation
ii. Data preprocessing
iii. Training
iv. Testing

The framework of the proposed system is shown in Figure 5.1. In the system, there are four phases. Data collection and data cleaning is performed in the data preprocessing phase. And then, input sentence is segmented into syllable. After syllable segmentation, each syllable is tagged manually as a preparation of data. Because there is no standard corpus for joint word segmentation and stemming in our language, Myanmar. In order to train the joint word segmentation, stemming and named entity detection model, each syllable is need to tagged manually. And then, in the training phase, joint model is trained on Neural architecture. Training data firstly through the character representation layer. And then, character-level representation and syllable-level embedding are combined at syllable sequence layer. Subsequently,

the inference layer assigns labels to each word using the hidden states of words sequence representations. In the final stage, untagged data are tested.

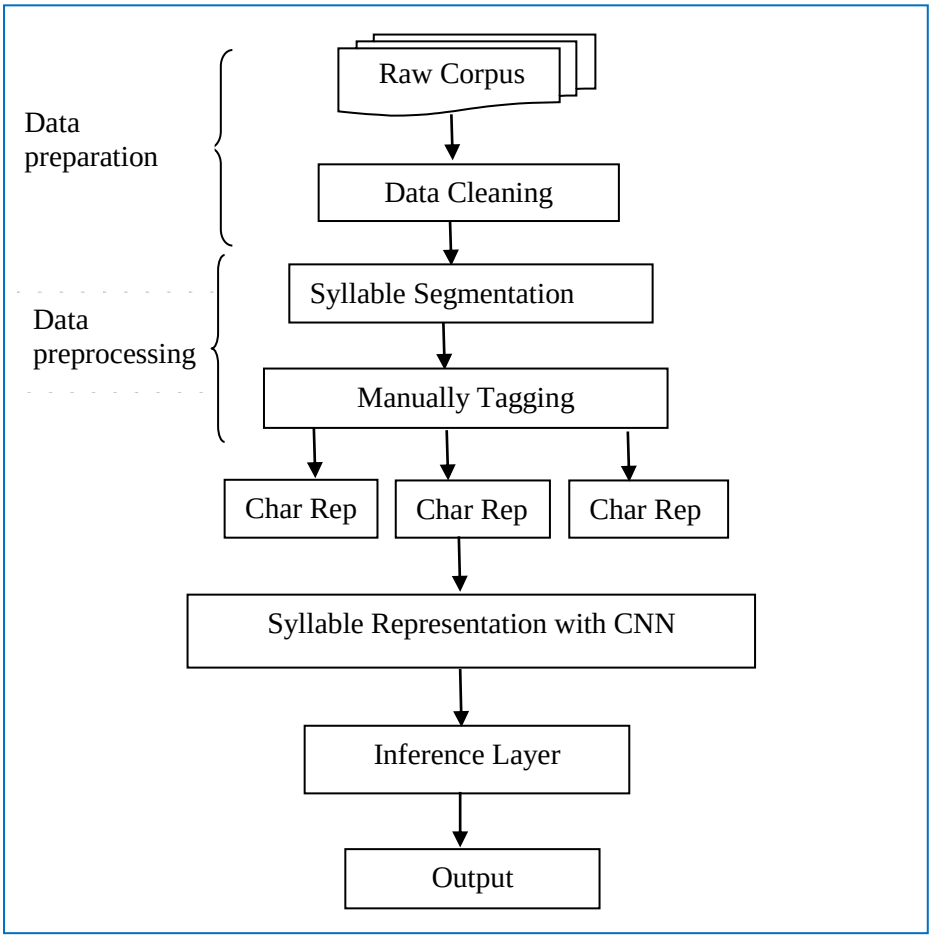

Figure 5.1 Framework of the Proposed System

5.1 Proposed System Specification

In this approach, stemming and named entity detection are considered as a typical sequence tagging problem over segmented words, while segmentation also can be modeled as syllable-level tagging problem via prediction the labels that identify the word boundaries. Stemming is performed as a learning process which begins with the collection of a sizeable set of examples $\{(x_1; y_1); : : : ; (x_n; y_n)\}$, where for each i € {1; : : : ; n} the input vector x_i represents the features of a given sentence and the scalar y_i is a label indicating whether the word belongs ($y_i = 1$) or not ($y_i = -1$) to a particular word class (i.e., root word, suffix). However, word segmentation for Myanmar Language, like for most Asian Languages, is a vital task and widely-studied sequence labeling problem. Normally, stemming is considered as a separate process from

segmentation. This new approach indicates segmentation boundaries when it performs stemming.

5.1.1 Sentence Segmentation

In Myanmar language, there is no white space between words, but the sentences are delimited by sentence end marker called "။" pote-ma. In order to process any NLP tasks, the sentences are firstly separated by using sentence end marker. The following paragraph is an example of input paragraph.

ရန်ကုန်မြို့ သုဝဏ္ဏအားကစားကွင်းတွင် ဒီဇင်ဘာလ ၁၅ရက်မှ ၂၅ရက်အထိ တိုင်းနှင့်ပြည်နယ် ဆောင်းရာသီ အားကစားပြိုင်ပွဲ ကျင်းပခဲ့သည်။ ရန်ကုန်မြို့တွင် ကျင်းပသော အားကစားပွဲတော် ဖွင့်ပွဲတွင် ဒေါ်အောင်ဆန်းစုကြည် မှ မိန့်ခွန်းပြောကြားခဲ့သည် ။ ထို့နောက် ဆောင်းရာသီ အားကစားပွဲတော် ဖွင့်ပွဲကို ဖဲကြိုးဖြတ် ဖွင့်လှစ်ခဲ့ကြသည်။ ပထမပွဲစဉ် အဖြစ် ရန်ကုန်တိုင်း အသင်းနှင့် ဧရာဝတီတိုင်း အသင်းတို့ ယှဉ်ပြိုင်ခဲ့ကြသည်။

The input paragraph has four end markers "။" pote-ma, the following four sentences are obtained after sentence segmentation.

1. **ရန်ကုန်မြို့ သုဝဏ္ဏအားကစားကွင်းတွင် ဒီဇင်ဘာလ ၁၅ရက်မှ ၂၅ရက်အထိ တိုင်းနှင့်ပြည်နယ် ဆောင်းရာသီ အားကစားပြိုင်ပွဲ ကျင်းပခဲ့သည်။**
2. **ရန်ကုန်မြို့တွင် ကျင်းပသော အားကစားပွဲတော် ဖွင့်ပွဲတွင် ဒေါ်အောင်ဆန်းစုကြည် မှ မိန့်ခွန်းပြောကြားခဲ့သည်။**
3. **ထို့နောက် ဆောင်းရာသီ အားကစားပွဲတော် ဖွင့်ပွဲကို ဖဲကြိုးဖြတ် ဖွင့်လှစ်ခဲ့ကြသည်။**
4. **ပထမပွဲစဉ် အဖြစ် ရန်ကုန်တိုင်း အသင်းနှင့် ဧရာဝတီတိုင်း အသင်းတို့ ယှဉ်ပြိုင်ခဲ့ကြသည်။**

5.1.2 Syllable Segmentation

Syllable is a basic sound unit. A word can be consisted of one or more syllables. Every syllable boundary can also be a word boundary. Some words can include other words; it is called a compound word. Syllable breaking is a necessary

step for Myanmar word segmentation. For syllable segmentation, this system uses the algorithm of [47]. Examples of syllable segmentation are shown below.

1. ရန်ကုန်မြို့ သုဝဏ္ဏအားကစားကွင်းတွင် ဒီဇင်ဘာလ ၁၅ရက်မှ ၂၅ရက်အထိ တိုင်းနှင့်ပြည်နယ် ဆောင်းရာသီ အားကစားပြိုင်ပွဲ ကျင်းပခဲ့သည်။

   After syllable segmentation,

   ရန် ကုန် မြို့ သု ဝဏ္ဏ အား က စား ကွင်း တွင် ဒီ ဇင် ဘာ လ ၁ ၅ ရက် မှ ၂ ၅ ရက် အ ထိ တိုင်း နှင့် ပြည် နယ် ဆောင်း ရာ သီ အား က စား ပြိုင် ပွဲ ကျင်း ပ ခဲ့ သည် ။

2. ရန်ကုန်မြို့တွင် ကျင်းပသော အားကစားပွဲတော် ဖွင့်ပွဲတွင် ဒေါ်အောင်ဆန်းစုကြည် မှ မိန့်ခွန်းပြောကြားခဲ့သည်။

   After syllable segmentation,

   ရန် ကုန် မြို့ တွင် ကျင်း ပ သော အား က စား ပွဲ တော် ဖွင့် ပွဲ တွင် ဒေါ်အောင် ဆန်း စု ကြည် မှ မိန့် ခွန်း ပြော ကြား ခဲ့ သည် ။

3. ထို့နောက် ဆောင်းရာသီ အားကစားပွဲတော် ဖွင့်ပွဲကို ဖဲကြိုးဖြတ် ဖွင့်လှစ်ခဲ့ကြသည်။

   After syllable segmentation,

   ထို့ နောက် ဆောင်း ရာ သီ အား က စား ပွဲ တော် ဖွင့် ပွဲ ကို ဖဲ ကြိုး ဖြတ် ဖွင့် လှစ် ခဲ့ ကြ သည် ။

4. ပထမပွဲစဉ် အဖြစ် ရန်ကုန်တိုင်း အသင်းနှင့် ဧရာဝတီတိုင်း အသင်းတို့ ယှဉ်ပြိုင်ခဲ့ကြသည်။

   After syllable segmentation,

   ပ ထ မ ပွဲ စဉ် အ ဖြစ် ရန် ကုန် တိုင်း အ သင်း နှင့် ဧ ရာ ဝ တီ တိုင်း အ သင်း တို့ ယှဉ် ပြိုင် ခဲ့ ကြ သည် ။

5.1.3 Train with Neural Network Architecture

In the system, there are four phases. Data collection and syllable segmentation is performed in the Dataset preparation phase. Then, as a preprocessing, syllables are tagged manually. In the training phase, joint process is trained on Neural architecture. In the final stage, untagged data are tested. Training and testing phase of neural sequence labeling is shown in Figure 5.2.

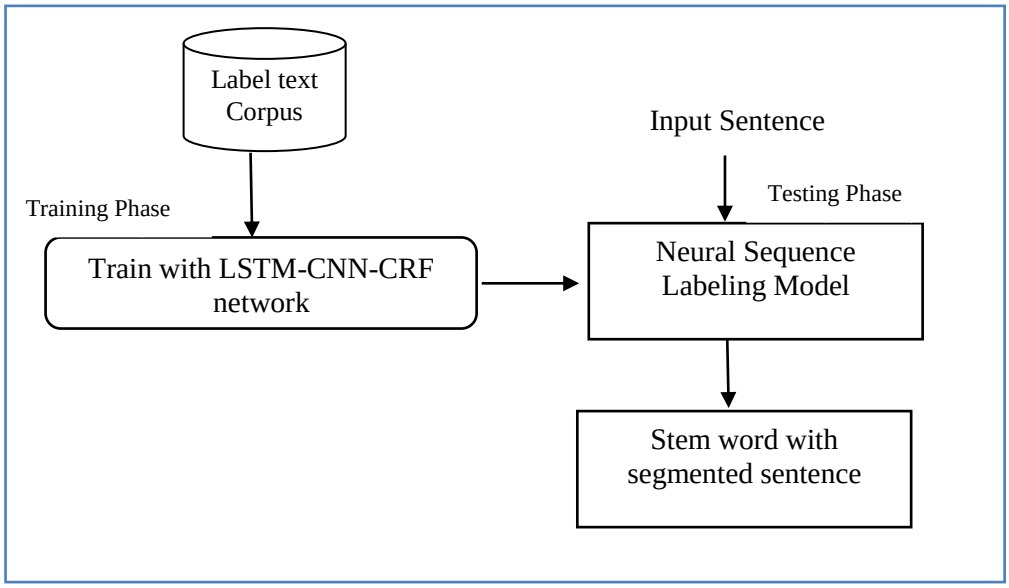

Figure 5.2 Training and Testing Phase of Neural Sequence Labeling

The training and testing phase of neural sequence labeling includes the following three steps:

(1) Training label text corpus with neural sequence labeling model

(2) Applying the model to test the new input sentence.

(3) From the tagging result word segmentation boundary, root word and named entity are detect.

The input of the training phase is the label text corpus which are acquired from the manually segmented and tagged the raw data. In the testing phase, input sentences are segmented into syllable. From the syllable segmented sentences, each syllable assigns the word type label to detect the boundary of the word and to extract the word type. So, the task of joint word segmentation, stemming and named entity detection is to assign word type label to every syllable in a sentence.

Example of syllable tagging for the first sentence is:

ရန်/B-NE ကုန်/I-NE မြို့/B-R သု/B-NE ဝဏ္ဏ/I-NE အား/B-R က/I-R စား/I-R ကွင်း/I-R တွင်/B-S ဒီ/B-NE ဧင်/I-NE ဘာ/I-NE လ/B-R ၁/B-S ၅/I-S ရက်/B-R မှ/B-S ၂/B-S ၅/I-S ရက်/B-R အ/B-S ထိ/I-S တိုင်း/B-R နှင့်/B-S ပြည်/B-R နယ်/I-R ဆောင်း/B-R ရာ/I-R သီ/I-R အား/B-R က/I-R စား/I-R ပြိုင်/B-R ပွဲ/I-R ကျင်း/B-R ပ/I-R ခဲ့/B-Suf သည်/I-Suf ။/O

In this example, "ရန်ကုန်" is the named entity. So, it is assigned as NE tag. In this named entity "ရန်" is beginning of the name "B-NE" and "ကုန်" is end of the named entity "I-NE" "ရန်/B-NE ကုန်/I-NE". "မြို့" is root word, it is tagged as "R" and it is also the only one word and beginning of the root word "မြို့/B-R". In the word "သုဝဏ္ဏ" also the named entity and "သု" is the beginning of the named entity "B-NE" and "ဝဏ္ဏ" is the end of the named entity "I-NE". "အားကစားကွင်း" is the root word and "အား" is the beginning of the root word "B-R", က is the intermediate word of the root "I-R", "စား" also the intermediate word of the root "I-R" and "ကွင်း" is the end of the root word "I-R" "အား/B-R က/I-R စား/I-R ကွင်း/I-R". The word "တွင်" is the Postpositional marker and is the assigned as a single word "တွင်/B-S". "ဒီဇင်ဘာ" is the name of the month, it is identified as named entity and "ဒီ" is the beginning of the named entity "B-NE", "ဇင်" is the middle word of the named entity "I-NE" and "ဘာ" is the last word of the named entity "I-NE" "ဒီ/B-NE ဇင်/I-NE ဘာ/I-NE". "လ" means month and it is only one root word "လ/B-R". ၁၅ is the numerical number so it is added as a single word "၁/B-S ၅/I-S". "ရက်" means day and it is root word "ရက်/B-R". "မှ" is the postpositional marker and label with single word "မှ/B-S". "၂၅" also means numerical number "၂/B-S ၅/I-S". "အထိ" is postpositional marker "အ/B-S ထိ/I-S". In the word "တိုင်းနှင့်ပြည်နယ်", it is separated into three words "တိုင်း", "နှင့်" and "ပြည်နယ်". "တိုင်း" is the root word "တိုင်း/B-R", "နှင့်" is postpositional marker "နှင့်/B-S" and "ပြည်နယ်" is the root word "ပြည်/B-R နယ်/I-R". "ဆောင်းရာသီ" is the root word "ဆောင်း/B-R ရာ/I-R သီ/I-R". "အားကစား" "ပြိုင်ပွဲ" also root word "အား/B-R က/I-R စား/I-R ပြိုင်/B-R ပွဲ/I-R". "ကျင်းပခဲ့သည်" is past tense verb and the word "ကျင်းပ" is root verb "ကျင်း/B-R ပ/I-R" and "ခဲ့/B-Suf သည်/I-Suf" is past tense suffix. "။" assign as other

word, it means that it is not root word, single word or suffix. It is just a symbol so it is assigned as other word "။/O".

Example of syllable tagging for the second sentence is:

ရန်/B-NE ကုန်/I-NE မြို့/B-R တွင်/B-S ကျင်း/B-R ပ/I-R သော/B-Suf အား/B-R က/I-R စား/I-R ပွဲ/B-R တော်/I-R ဖွင့်/B-R ပွဲ/I-R တွင်/B-S ဒေါ်/B-Pre အောင်/B-NE ဆန်း/I-NE စု/I-NE ကြည်/I-NE မှ/B-S မိန့်/B-R ခွန်း/I-R ပြော/B-R ကြား/I-R ခဲ့/B-Suf သည်/I-Suf ။/O

In this example, the word "**ဒေါ်အောင်ဆန်းစုကြည်**" is named entity and "**ဒေါ်**" is the prefix of the named and it refers to femal name so "**ဒေါ်**" is prefix "**ဒေါ်**/B-Pre" and "**အောင်ဆန်းစုကြည်**" is assigned as "**အောင်**/B-NE **ဆန်း**/I-NE **စု**/I-NE **ကြည်**/I-NE".

Example of syllable tagging for the third sentence is:

ထို့/B-S နောက်/I-S ဆောင်း/B-R ရာ/I-R သီ/I-R အား/B-R က/I-R စား/I-R ပွဲ/B-R တော်/I-R ဖွင့်/B-R ပွဲ/I-R ကို/B-S ဖဲ/B-R ကြိုး/I-R ဖြတ်/I-R ဖွင့်/B-R လှစ်/I-R ခဲ့/B-Suf ကြ/I-Suf သည်/I-Suf ။/O

Example of syllable tagging for the last sentence is:

ပ/B-R ထ/I-R မ/I-R ပွဲ/B-R စဉ်/I-R အ/B-S ဖြစ်/I-S ရန်/B-NE ကုန်/I-NE တိုင်း/B-R အ/B-R သင်း/I-R နှင့်/B-S ဧ/B-NE ရာ/I-NE ဝ/I-NE တီ/I-NE တိုင်း/B-R အ/B-R သင်း/I-R တို့/B-S ယှဉ်/B-R ပြိုင်/I-R ခဲ့/B-Suf ကြ/I-Suf သည်/I-Suf ။/O

Now, the tag data mentioned above are described in what way to denote in result data.

- The stem word is placed within the boundary marker []

  [တာဝန်]+များ , [လေ့လာ]+ရေး, [မြို့]+များ

  In this case, "**တာဝန်**", "**လေ့လာ**", "**မြို့**" are stem word. So, these words are placed in the boundary marker [].

- suffix words are marked by +

  [ဖွံ့ဖြိုး]+ဖို့, [ပေါ်ထွက်]+ခဲ့+သော

In these words, "ဖို့" and "ခဲ့သော" are suffix word. So, these words are marked by + marker.

- prefix is delimited by ^ marker

  အ ^ [ကြီးမြတ်]+ဆုံး

  In this word, "**အ**" is prefix, "**ကြီးမြတ်**" is stem word and "**ဆုံး**" is suffix. So, the prefix "**အ**" is delimited by ^ marker.

- spaces between words are separated by _ marker

  အ^[ပူ]+ဆုံး_ [ရာသီ]

  In this case, "**အပူဆုံး**" is one word and "**ရာသီ**" is one word. These words are separated by _ marker.

- {} is used to identify the named entity

  {တရုတ်}_[နိုင်ငံ], ဦး ^{မြင့်ဦး}

  In these words, "**တရုတ်**" and "**မြင့်ဦး**" are named entities. So, they are placed in the boundary marker {}.

In the first sentence,

ရန်/B-NE **ကုန်**/I-NE **မြို့**/B-R **သု**/B-NE **ဝဏ္ဏ**/I-NE **အား**/B-R **က**/I-R **စား**/I-R **ကွင်း**/I-R **တွင်**/B-S **ဒီ**/B-NE **ဧင်**/I-NE **ဘာ**/I-NE **လ**/B-R **၁**/B-S **၅**/I-S **ရက်**/B-R **မှ**/B-S **၂**/B-S **၅**/I-S **ရက်**/B-R **အ**/B-S **ထိ**/I-S **တိုင်း**/B-R **နှင့်**/B-S **ပြည်**/B-R **နယ်**/I-R **ဆောင်း**/B-R **ရာ**/I-R **သီ**/I-R **အား**/B-R **က**/I-R **စား**/I-R **ပြိုင်**/I-R **ပွဲ**/I-R **ကျင်း**/B-R **ပ**/I-R **ခဲ့**/B-Suf **သည်**/I-Suf **။**/O

The output text for joint segmentation and stemming is,

{ရန်ကုန်}_[မြို့]_{သုဝဏ္ဏ}_[အားကစားကွင်း]_တွင်_{ဒီဧင်ဘာ}_[လ]_၁၅_[ရက်]_မှ_၂၅_[ရက်]_အ ထိ_[တိုင်း]_နှင့်_[ပြည်နယ်]_[ဆောင်းရာသီ]_[အားကစား]_ [ပြိုင်ပွဲ]_[ကျင်းပ]+ခဲ့+သည်_။_

The output text for segmentation is,

ရန်ကုန်_မြို့_သုဝဏ္ဏ_အားကစားကွင်း_တွင်_ဒီဇင်ဘာ_လ_၁၅_ရက်_မှ_၂၅_ရက်_အထိ_တိုင်း_နှင့်_ပြည်နယ်_ဆောင်းရာသီ_အားကစား_ပြိုင်ပွဲ_ကျင်းပ+ခဲ့သည်_။_

In the second sentence,

ရန်/B-NE ကုန်/I-NE မြို့/B-R တွင်/B-S ကျင်း/B-R ပ/I-R သော/B-Suf အား/B-R က/I-R စား/I-R ပွဲ/B-R တော်/I-R ဖွင့်/B-R ပွဲ/I-R တွင်/B-S ဒေါ်/B-Pre အောင်/B-NE ဆန်း/I-NE စု/I-NE ကြည်/I-NE မှ/B-S မိန့်/B-R ခွန်း/I-R ပြော/I-R ကြား/I-R ခဲ့/B-Suf သည်/I-Suf ။/O

The output text for joint segmentation and stemming is,

{ရန်ကုန်}_[မြို့]_တွင်_[ကျင်းပ]+သော_[အားကစား]_[ပွဲတော်]_[ဖွင့်ပွဲ]_တွင်_ဒေါ်{အောင်ဆန်းစုကြည်}_မှ_ [မိန့်ခွန်း]_ [ပြောကြား]+ခဲ့+သည်_။_

The output text for segmentation is,

ရန်ကုန်_မြို့_တွင်_ကျင်းပ+သော_အားကစား_ပွဲတော်_ဖွင့်ပွဲ_တွင်_ဒေါ်အောင်ဆန်းစုကြည်_မှ_ မိန့်ခွန်း_ပြောကြား+ခဲ့သည်_။_

In the third sentence,

ထို/B-S နောက်/I-S ဆောင်း/B-R ရာ/I-R သီ/I-R အား/B-R က/I-R စား/I-R ပွဲ/I-R တော်/I-R ဖွင့်/B-R ပွဲ/I-R ကို/B-S ဖဲ/B-R ကြိုး/I-R ဖြတ်/I-R ဖွင့်/B-R လှစ်/I-R ခဲ့/B-Suf ကြ/I-Suf သည်/I-Suf ။/O

The output text is for joint segmentation and stemming is,

ထို့နောက်_[ဆောင်းရာသီ]_[အားကစား]_[ပွဲတော်]_[ဖွင့်ပွဲ]_ကို_[ဖဲကြိုးဖြတ်]_[ဖွင့်လှစ်]+ခဲ့+ကြ+သည်_။_

The output text for segmentation is,

ထို့နောက်_ဆောင်းရာသီ_အားကစားပွဲတော်_ဖွင့်ပွဲ_ကို_ဖဲကြိုးဖြတ်_ဖွင့်လှစ်ခဲ့ကြသည်_။_

In the last sentence,

ပ/B-R ထ/I-R မ/I-R ပွဲ/B-R စဉ်/I-R အ/B-S ဖြစ်/I-S ရန်/B-NE ကုန်/I-NE တိုင်း/B-R အ/B-R သင်း/I-R နှင့်/B-S ဧ/B-NE ရာ/I-NE ဝ/I-NE တီ/I-NE တိုင်း/B-R အ/B-R သင်း/I-R တို့/B-S ယှဉ်/B-R ပြိုင်/I-R ခဲ့/B-Suf ကြ/I-Suf သည်/I-Suf ။/O

The output text is for joint segmentation and stemming is:

[ပထမ]_[ပွဲစဉ်]_အဖြစ်_{ရန်ကုန်}_[တိုင်း]_အသင်း_နှင့်_{ဧရာဝတီ}_[တိုင်း]_[အသင်း]_တို့_

[ယှဉ်ပြိုင်]+ခဲ့+ကြ+သည်_။_

The output text for segmentation is,

ပထမ_ပွဲစဉ်_အဖြစ်_ရန်ကုန်_တိုင်း_အသင်း_နှင့်_ဧရာဝတီ_တိုင်း_အသင်း_တို့_ ယှဉ်ပြိုင်ခဲ့ကြသည်_။_

5.2 Summary

This chapter describes the design and implementation of the proposed system by displaying the output results. Step by step output result is added so that it can clearly understand the flow of this system and the proposed methods. The structure of the proposed neural sequence labeling and proposed tagging scheme incorporates in this sections. It displays a more understandable form of tagging scheme for root word extraction and named entity detection. The evaluation results of the proposed system are discussed in chapter 7.

CHAPTER 6
EXPERIMENTAL RESULTS

In this chapter, the experimental study is discussed on performance evaluations of the proposed joint word segmentation, stemming and Named Entity Detection process. The results are attained by the required data collection, data preprocessing and hyper-parameter tuning. The goal of this chapter is to evaluate the performance of neural network architecture for joint process. Moreover, error analysis is also presented with related examples.

There are four main parts in this chapter. The first part explains about the data and parameter setting. The second part is generating and evaluating different architecture of the neural network. The third part is the evaluation of performance based on different hyper-parameters such as applying proposed pre-trained embedding, tuning different optimizers, different learning rate and dropout rate. The last part is error analysis. The NCRF++ toolkit [66] was used to build neural sequence labelling architecture for joint process of Myanmar word. Experimental setting is trained and discussed on Nvidia Tesla K80 GPU sever, training takes about 18 hours while tagging the test set takes about 60 seconds for CoNLL 2003.

6.1 Setting

This part will explain about the dataset and the corpus building. And it will describe the parameter setting in neural sequence labeling model. Moreover, it will reveal about the evaluation.

6.1.1 Corpus Building

In Myanmar language, there is no standard corpus for joint word segmentation and stemming. Building a robust and accurate corpus is more complex and time consuming, compared to building such a corpus in other similar language by the facts that a "word" is difficult to define, as it does not exhibit explicit word boundaries and there is no agreement on word segmentation. [63] So, training corpus are not useful because results will be different across different researchers.

In this work, a corpus of 20K sentences are built from many sites such as Thit Htoo Lwin, Eleven News, 7Days News Journal and Newspapers. Moreover, 10K sentences are also collected from ALT (Asian Language Treebank). Although the

corpus is not large enough to cover a broad range of Myanmar words, it contains documents from many popular news articles such as politic, information technology, sport, education, economics etc. In addition to the corpus, 30K sentences are split into 80% of the data to training and the last 10% each to testing and development set. There are 5 different labels {B-R, I-R, B-Suf, I-Suf, B-Pre, I-Pre, B-S, I-S, O} (11 with BIO prefix included). The dataset statistics are shown in Table1.

Table 6.1: Statistics of datasets

Dataset	Train	Dev	Test
@Sent	25,000	2,418	2,286
@Syllable	1416k	133k	142k
@Root Word	240k	27k	28k
@Named Entity	40k	4k	4k
@Single Word	280k	32k	29k
@Suffix	118k	13k	12k
@Prefix	8k	932	883

6.1.2 Parameter Setting

The required parameters and their values for performance evaluations are described in table 6.2. Epoch 100 is used for training. In each epoch, the whole training data is divided into batches and process on one batch at a time. It is evaluated on batch size 20 in the experiments.

Table 6.2. Parameters and Values

Parameters	**Values**
CNN window	3
Character hidden	50
Syllable hidden	200
Embedding size	300
L2 regularization λ	1e-8
learning rate decay	0.05
Batch size	20
Epoch	100

6.1.3 Evaluation

Standard precision (P), recall (R) and F1-score (F) are used as the evaluation metrics for joint word segmentation, stemming and Named Entity Detection process [84].

$$Precision = \frac{Number\ of\ correctly\ extracted\ instances}{Number\ of\ extracted\ instences} \quad (6.1)$$

$$Recall = \frac{Number\ of\ Correctly\ extracted\ instances}{Number\ of\ all\ instances} \quad (6.2)$$

$$F - Mesaure = \frac{2 * Precision * Recall}{Precision + Recall} \quad (6.3)$$

6.2 Performance Evaluation on Different Network Architecture

Joint word segmentation and stemming neural sequence labelling framework contains three layers: character sequence representation, syllable sequence representation, and an inference layer. In character sequence representation, three different neural structures are modeled and the performance are compared through CNN, LSTM or GRU. Similarly, on the syllable level, CNN, LSTM and GRU models are investigated for joint sequence labelling tasks.

In this approach, each token in a sentence is considered independently and correlations between tags in a sentence cannot take into account. CRF classifier captures label dependencies by adding transition scores between neighboring labels. During the decoding process, the Viterbi algorithm is used to search the label sequence with the highest probability. To simplify the description, we use “CCNN”, “CLSTM”, “CGRU” represent character structure and “SCNN”, “SLSTM”, “SGRU” represent syllable structure, respectively. Table 6.3 shows the experimental results on different architecture of networks under same hyper-parameters with CRF inference layer.

Training is done by stochastic gradient descent (SGD) optimizer with fixed learning rate 0.005.

Table 6.3. Comparison and Analysis of Different Architecture of Network

Model	Precision	Recall	F1
NoChar+SCNN+CRF	86.66	87.50	87.08
NoChar+SLSTM+CRF	85.93	85.52	85.72
NoChar+SGRU+CRF	87.11	85.88	86.49
CCNN+SCNN+CRF	86.86	87.53	87.19
CCNN+SLSTM+CRF	85.81	87.00	86.40
CCNN+SGRU+CRF	86.21	87.03	86.62
CLSTM+SCNN+CRF	**87.64**	**87.97**	**87.80**
CLSTM+SLSTM+CRF	86.39	86.92	86.65
CLSTM+SGRU+CRF	85.44	85.34	85.39
CGRU+SCNN+CRF	87.17	87.67	87.42
CGRU+SLSTM+CRF	86.77	87.56	87.17
CGRU+SGRU+CRF	89.02	84.57	86.74

The performance difference between character representation and syllable representation with different model are also evaluated. In the table, most work focus on SCNN+CRF structure with different character representations. NoChar+SCNN+CRF model also gives the comparable performance even though there is no character representation. CLSTM+SCNN+CRF model gives the best result.

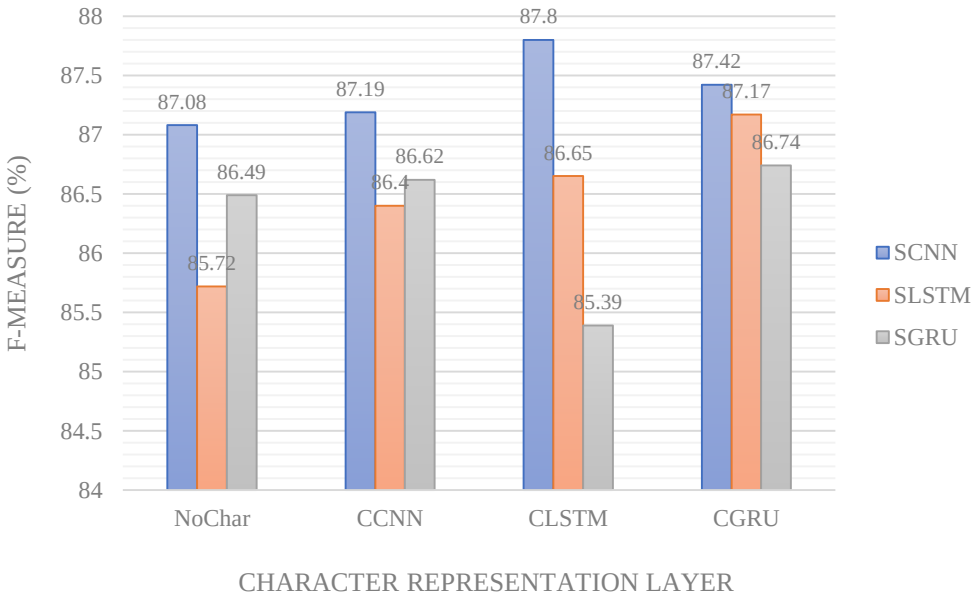

Figure 6.1 Comparison with different Architecture of Neural Network

The performance of difference between character representation and syllable representation with different models are evaluated. According to the experimental results, two things are found. The first one is, when character sequence layer is used, accuracy is better and the second one is, most work focuses on syllable CNN network, when the CNN network in syllable layer are used, accuracy is better than others. CLSTM+SCNN+CRF model gives best result.

6.3 Performance Evaluation on Different Hyper-parameter

In this part, joint model is evaluated on different setups like the importance learning rate, dropout rate, different kinds of optimizers and pre-trained word embedding that have a large impact on the performance.

6.3.1 Word Embedding

Nowadays, deep learning approaches have become more and more popular in NLP tasks and we need to do word embedding because many machine learning algorithms and most of the deep learning architectures cannot process the raw form of strings or plain texts. Therefore, pre-trained embedding layers have been applied to improve the performance of neural network architectures for NLP tasks. [84] The main target of word embedding model is to convert word to the form of numeric vectors. Most existing word embedding results are generally trained on data source such as news pages or Wikipedia articles.

Pre-trained embedding is a type of vector representation that admits words with same meaning to have a same vector representation. It is influenced on various NLP research fields including document classification, author identification, sentiment analysis, etc. Actually, it is a class of techniques where individual words are represented as real-valued vectors in a predefined vector space. Each word is mapped to one vector and the vector values are learned in a way that relates a neural network, and the technique is associated with deep learning approach. [7] They are a distributed representation for text that may be one of the key improvements for the effective performance of deep learning approaches on challenging natural language processing problems.

The main influence of having such a distributed representation over word classes is that it can capture many dimensions of both semantic and syntactic information in a vector where each dimension correlate to an inherent feature of the

word. To train word vector, we use Word2Vec [86] both Skip-gram and CBOW models between 100 to 700 dimensions. In English language, word embedding model can be applied for data preprocessing well but there is a very little amount of work done in Myanmar language. Information about word morphology and shape is normally overlooked when learning word representations.

However, for tasks like stemming, intra-word information is intensely useful, especially when dealing with morphologically rich languages. Text preprocessing as word embedding is important part to build neural network and it is a significantly effect on final results.

The first step to process a sentence by neural architecture is to transform characters into embedding. This transformation is done by lookup embedding table. A character lookup table $M_{char} \in R^{|Vchar| \times d}$ where $|V_{char}|$ denotes the size of the character vocabulary and d denotes the dimension of embeddings is associated with all characters. Given a sentence $S = (c_1; c_2; ...; c_L)$, after the lookup table operation, we obtain a matrix $X \in R^{L \times d}$ where the i^{th} row is the character embedding of c_i. When we apply pre-trained embedding with own training data, the performance improves. One of the key points of this architecture to take advantage of better pre-trained embedding. Experiment on several word embedding approaches that influence the accuracy of joint model on this task.

6.3.1.1 Evaluation with Different Dimension

The model of embedding word2vec is used which implements the CBOW and skip-gram with dimensionality size equal to 100-700. The CBOW model learns the embedding by predicting the current word based on its context. [87] The continuous skip-gram model learns by predicting the surrounding words given a current word. Both models are undertaken on learning about words given their local usage context, where the context is defined by a window of neighboring words. The key advantage of the approach is that high-quality word embedding can be learned effectively (low space and time complexity), allowing larger embedding to be learned (more dimensions) from much larger corpora of text (billions of words).

In both methods, same parameters are used when running word2vec with iteration and a context window of size five. Pre-trained embedding is used on raw segmented data with 27K vocabulary size. All of the pre-trained data are included in

the training corpus. Mostly use the huge size of the crawl data. In this work, own data is used to retrained data the model which are included in training corpus.

By using this idea, it can reduce oov% (out-of-vocabulary) when the embedding table is used in CNN model. As a result, the number of perfect-match words raise in embedding lookup tables, oov% also decrease. The table shows the comparison of oov% between common crawl data and own pre-trained data.

Table 6.4. Comparison of OOV% in CNN Model

Type of Data	Pre-trained word	Perfect-match	OOV%
Crawl Data	341158	2144	44.67
Raw Segmented Data	26792	2739	29.30

Firstly, the performance is evaluated with different dimensions between 100-700 on raw segmented data. We named the Skipgram1 for dimension-100 in skipgram approach and CBOW1 for dimension-100 in CBOW approach.

Table 6.5. Results with different dimensions of pre-trained embedding

Dimension	Name	Precision	Recall	F-Measure
D-100	Skipgram1	87.33	86.08	86.69
	CBOW1	87.23	86.04	86.63
D-200	Skipgram2	88.43	87.45	87.94
	CBOW2	88.34	87.26	87.80
D-300	Skipgram3	88.50	87.57	88.03
	CBOW3	88.94	88.30	88.62
D-400	Skipgram4	88.89	88.41	88.65
	CBOW4	89.09	88.56	88.82
D-500	Skipgram5	88.55	87.58	88.06
	CBOW5	88.73	88.35	88.54
D-600	**Skipgram6**	**89.59**	**88.49**	**89.40**
	CBOW6	89.11	88.38	88.74
D-700	Skipgram7	89.18	88.56	88.87
	CBOW7	88.52	88.31	88.41

The results in Table 6.5 are trained using Word2Vec embedding both CBOW and skip-gram dimension vary from 100-700 with raw segmented data. In this work, we have tried to provide very high-quality set of pre-trained word vector representations. Our findings indicate that the performance in CNN based model is improved when dimension of embedding vector is expanded. The best result is 600-d with skipgram approach in CNN based model. In CBOW approach, the best result achieves in 400-d.

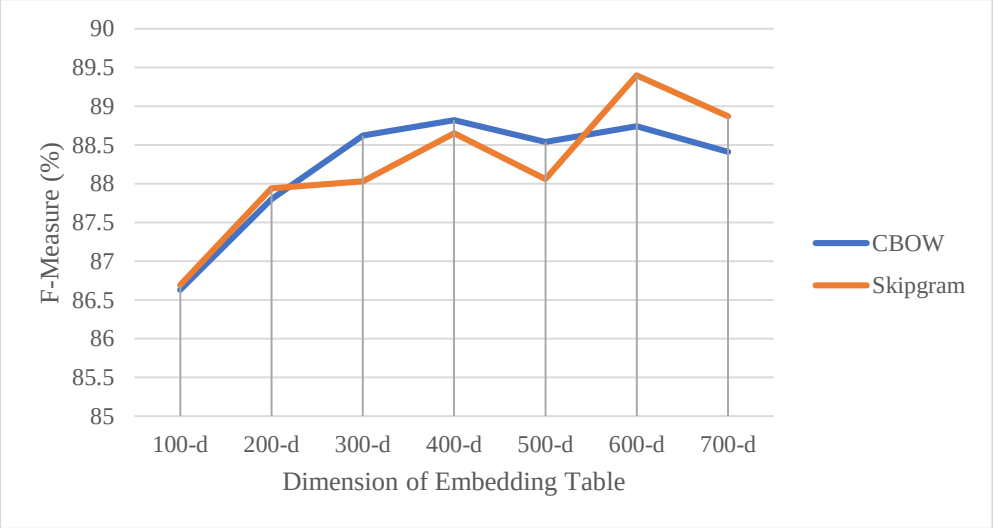

Figure 6.2 Comparison of Word Embedding with Skipgram and CBOW

Word embedding is just mapping from word to vectors. Dimensionality in word embedding refers to the length of these vector. According to the chart, the higher dimension of embedding table, the better performance we get. The larger vector can store more information. So, the higher the dimension, the more information can capture. The best performance is 600-d in Skipgram approach.

6.3.1.2 Evaluation with Baseline Embedding

Moreover, the performance with baseline is compared with own embedding. Baseline CNN based model using embedding vector from Learning Word Vectors for 157 Languages that trained on 3 billion vocabulary size from [17] by E. Grave, P. Bojanowski, P. Gupta, A. Joulin, T. Mikolov for both word and character embedding as a baseline an approach.

In the experiment, it compares the performance of different pre-trained embedding choices and baseline embedding impact on the CNN based model for joint process. CNN based model with no embedding is also compared.

Table 6.6. Comparison of Different Pre-Trained Models with Baseline Word Embedding

Name	Precision	Recall	F-Measure
Baseline	89.10	87.86	87.08
No Embedding	86.08	83.43	84.62
Skipgram 300-d	88.50	87.57	88.03
CBOW 300-d	88.94	88.30	88.62

Table 6.6 presents the result compared to different pre-trained embedding table when incorporate into CNN based model and with baseline embedding table. With no embedding layer in neural sequence labeling model, F-Measure is 84%. With baseline embedding, F-Measure is 87%. Baseline embedding is using 300-d. Although own data that trained in embedding file is not large enough but it is cleaner than crawl data. So, the results are better than the baseline embedding.

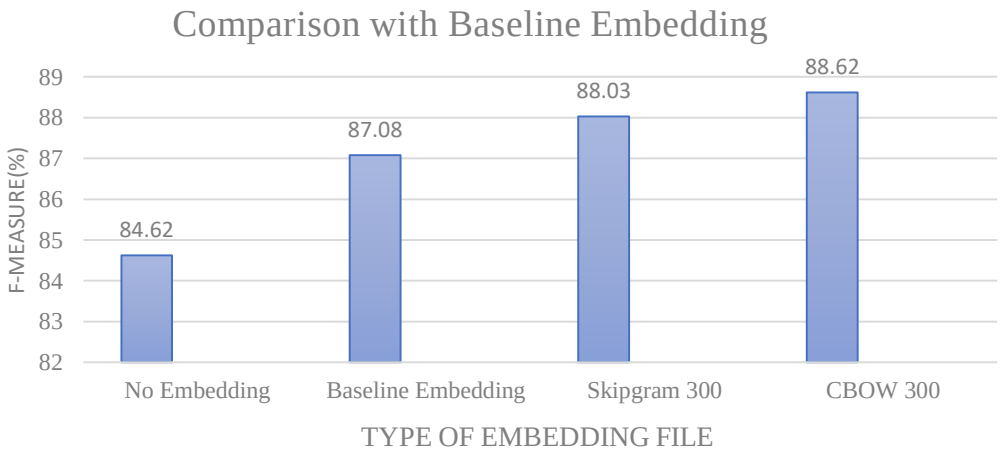

Figure 6.3 Comparison with baseline embedding

6.3.2 Optimizers

Optimizers, combined with Loss Function, are the key pieces that enable Machine Learning to work for training data. During the training process, optimizers

adjust and change the parameters of model to minimize the loss function and make predictions as possible as it is.

It is generally agreed that the choice of optimizers acts a vital role. The optimizer is responsible for minimizing the loss functions of neural network. A frequently selected optimizer is stochastic gradient descent (SGD). This section proposes the optimization process in Neural Architecture, how loss functions fit into the equation and finding the best optimizer on Adagrad, Adadelta, Adam, RMSProp and SGD.

Generally, neural network suffers from major problem, overfitting. Overfitting means the model which performs very well in training data, but could not work well in test data. To minimize overfitting, regularization process is applied on the model. Regularization is a term added into the optimization process that help to avoid overfitting. Dropout is a technique to address overfitting problem in neural networks which helps reducing interdependent learning between the neurons. [61] Dropout forces a neural network to learn more effective features that are useful in conjunction with many different random subsets of the other neurons. So, to mitigate overfitting, dropout is applied to regularize model.

Dropout is more effective on those problems where there is a limited amount of training data and the model is likely to overfit the training data. Myanmar is low resource language. Thus, for the task of joint word segmentation and stemming in Myanmar Language, different dropout rates are used to tune for each optimizer. Dropout is applied on character embedding before inputting to CNN.

Figure 6.4 shows the example of dropout, to understand, figure 6.4(a) represents a particular neural network model, then figure 6.4(b) represents the neural network model with drop out where few nodes were dropped with 'X' symbol. In this section, some effective regularization approaches such as dropout and optimization function are evaluated.

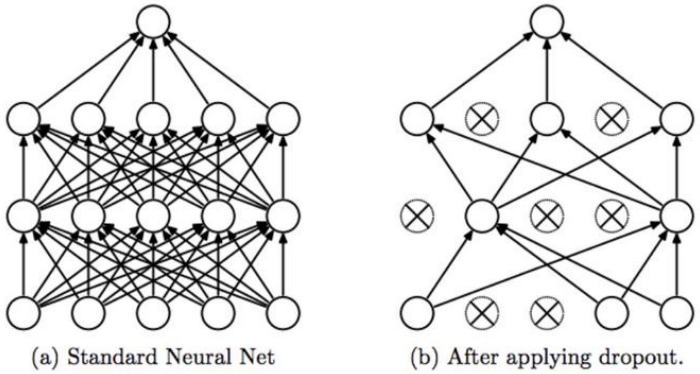

Figure 6.4 The Neural Networks Model (a) and the Model after Applying Dropout(b).

During the training process, with Dropout, individual nodes will be deleted with probability p; so incoming and outgoing edges also will be removed. For example, n be the number of node in neural network, then the possibility of the number of node to be active at each dropout p*n. If there are 1024 nodes in neural network and set p=0.5, then only the half of the node (512) will be active. It has applied a randomly selected dropout set {0.0, 0.2, 0.5, 0.7, 0.9} for joint model. Different value of dropout is added for each optimizer and applied to the sequential model of neural network architecture. All other hyper-parameters remain the same as mentioned above. It has obtained significant improvements on model performance after using dropout.

6.3.2.1 Stochastic Gradient Descent (SGD)

The first optimizer is SGD. It is an efficient and effective optimization method for a large number of published machine learning systems. However, SGD can be quite sensitive towards the selection of the learning rate [53]. Choosing a too large rate can cause the system to deviate the objective function, and choosing a too low rate results in a slow learning process. To exclude the short comings of SGD, other gradient-based optimization algorithms have been intended.

Table 6.7. Different Dropout rate with SGD optimizer

Dropout rate	***Precision***	***Recall***	***F-Measure***
0.0	89.18	89.58	89.38
0.1	89.84	89.64	89.74
0.2	**89.75**	**91.40**	**90.56**
0.3	89.04	90.12	89.57

Dropout rate	***Precision***	***Recall***	***F-Measure***
0.4	89.35	89.69	89.52
0.5	89.59	88.49	89.40
0.6	87.21	88.57	87.89
0.7	75.35	77.45	76.38
0.8	68.16	63.92	65.98
0.9	59.42	48.07	53.15

As stated in the chart while implementing neural network architecture with SGD optimizer on different dropout rate between 0.0 to 0.9, the best F-Measure is dropout 0.9 (90.56%). But dropout rate between 0.0 to 0.5, F-Measure is not very different. All are around 90% F-Measure. But F-Measure is decreasing from dropout rate 0.6 to 0.9. To be conclude, dropout rate 0.0 to 0.5 can give the best results. The larger the dropout rate, the lower the performance.

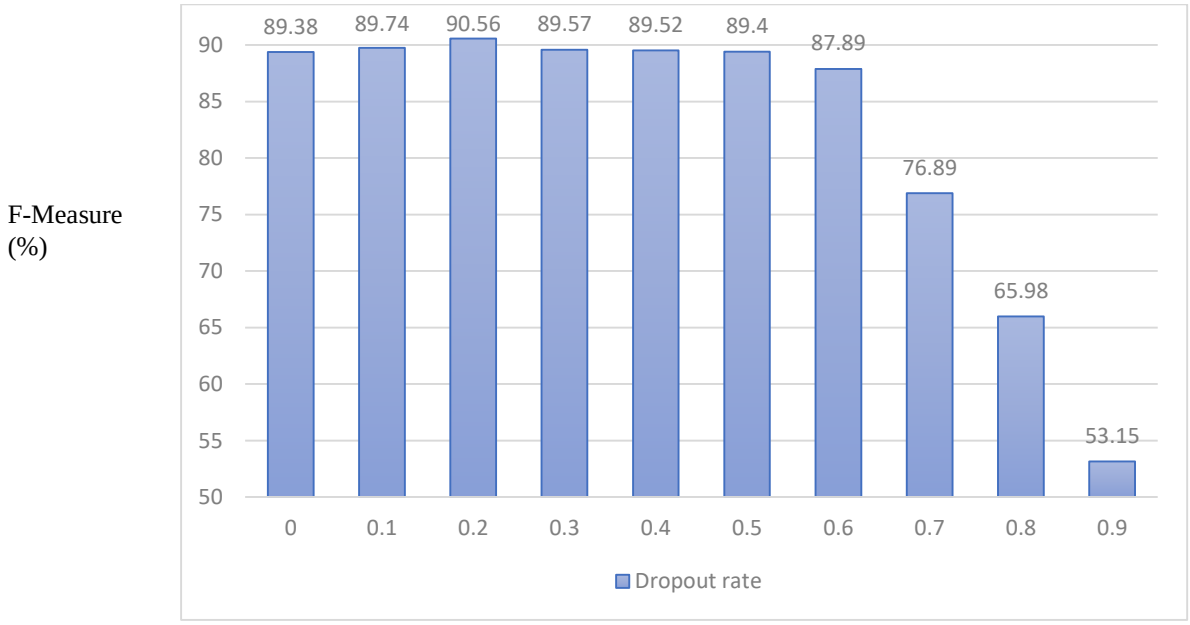

Figure 6.5 Comparison with Different Dropout rate with SGD Optimizer

6.3.2.2 Adagrad

The Adaptive Gradient Algorithm (Adagrad) [13] is an alternative of SGD and it is an adaptive learning rate method. Adagrad adapts the learning rate modified for each the parameters. It makes the larger update for infrequent and smaller update for frequent parameters. For this reason, it is well-suite for dealing with sparse data.

Table 6.8. Different Dropout rate with Adagrad Optimizer

Dropout rate	*Precision*	*Recall*	*F-Measure*
0.0	87.96	86.90	87.43
0.1	90.26	89.61	89.93
0.2	90.50	89.52	**90.01**
0.3	89.74	90.21	89.97
0.4	88.44	88.81	88.63
0.5	89.08	87.80	88.44
0.6	86.43	87.01	86.72
0.7	83.08	80.31	81.67
0.8	71.40	66.84	69.05
0.9	59.39	44.53	50.90

Table 6.8. shows the comparison between different dropout rate with Adagrad optimizer. The optimizer Adagrad gets maximum accuracy with dropout rate 0.2. Dropout rate 0.3 gets the second best result. It is observed that the result from dropout rate 0.0 to 0.7 gets over 80% F-Measure. Moreover, the result of dropout rate 0.1, 0.2 and 0.3 are not very different. All are around 90% F-Measure. F-Measure is dramatically decrease from 0.7 to 0.8. Dropout rate 0.9 is the worst performance.

6.3.2.3 Adadelta

Adadelta [67] is another extension of Adagrad optimizer that inquires to reduce its aggressive, monotonically decreasing learning rate. Instead of gathering all past gradients, it restricts the window of accumulated past gradient to some fixed size w.

Table 6.9. Different Dropout rate with Adadelta Optimizer

Dropout rate	*Precision*	*Recall*	*F-Measure*
0.0	86.32	85.87	86.09
0.1	87.96	88.40	88.18
0.2	90.77	89.82	90.30
0.3	90.08	90.87	**90.47**
0.4	89.31	90.10	89.70
0.5	87.39	88.22	87.80
0.6	68.77	67.13	67.94
0.7	59.54	55.77	57.59

Dropout rate	*Precision*	*Recall*	*F-Measure*
0.8	50.51	47.02	48.70
0.9	41.02	27.83	33.16

Table 6.9. shows the comparison between different dropout rates with Adadelta optimizer. The optimizer Adadelta gets maximum accuracy with dropout rate 0.3. Dropout rate 0.2 gets the second best result. It is observed that the performance increase 2% in dropout rate 0.0 to 0.1. The results form dropout rate between 0.0 to 0.5 are not very different. All are above 86% F-Measure. But the performance decrease nearly 20% in dropout rates 0.5 to 0.6. According to the experiment, the selection of dropout rate in Adadelta optimizer has a large impact on the performance of the system. The worst result is in dropout rate 0.9.

6.3.2.4 Adam

Adaptive Moment Estimation (Adam) [27] is another optimization algorithm that estimates adaptive learning rates for each parameter. It is computationally efficient because it requires less memory and it is well suited for problems that are large in terms of data or parameters. It is similar to Adadelta and RMSProp and it also keeps an exponentially decaying average of past gradients.

Table 6.10. Different Dropout rate with Adam optimizer

Dropout rate	*Precision*	*Recall*	*F-Measure*
0.0	89.04	87.89	88.46
0.1	88.41	89.73	89.06
0.2	90.14	89.18	**89.66**
0.3	89.63	89.05	89.34
0.4	89.27	89.01	89.14
0.5	88.80	89.15	88.98
0.6	87.14	89.18	88.15
0.7	85.22	86.82	86.01
0.8	75.61	70.48	72.95
0.9	70.32	71.04	70.68

Table 6.10. shows the comparison between different dropout rates with Adam optimizer. Adam optimizer gets maximum accuracy with dropout rate 0.2. The

performance of the model between dropout rates 0.0 to 0.5 are not very different. All are around 90% F-Measure. Dropout rate 0.9 gets the worst result. But in Adam optimizer, the result in dropout rate 0.9 is better than SGD. Moreover, Adam optimizer is more stable than SGD optimizer.

6.3.2.5 Root Mean Square Propagation (RMSProp)

Another optimizer is Root Mean Square Propagation (RMSProp) [64] and extension of Adagrad (adapting the learning rate). In RMSProp, learning rate gets adjusted automatically and it chooses a different learning rate for each parameter. Subsequent parameter values are based on previous gradient values calculated for particular parameter.

Table 6.11. Different Dropout rate with RMSProp optimizer

Dropout rate	*Precision*	*Recall*	*F-Measure*
0.0	88.67	88.09	88.38
0.1	89.43	90.51	89.96
0.2	90.01	90.64	90.32
0.3	90.57	90.22	**90.40**
0.4	90.02	90.20	90.11
0.5	88.70	90.48	89.58
0.6	87.30	88.94	88.11
0.7	85.75	85.45	85.60
0.8	81.29	80.81	81.05
0.9	76.24	75.31	75.78

Table 6.11. shows the comparison between different dropout rates with RMSProp optimizer. The optimizer RMSProp gets maximum accuracy with dropout rate 0.3. Dropout rate 0.2 gets the second best result. The performance of the model between dropout rates 0.1 to 0.5 are not very different. All are around 90% F-Measure. Dropout rate 0.9 gets the worst result. But in RMSProp optimizer, the result of dropout rate 0.9 is better than Adam.

6.3.3 Learning Rate

The learning rate is a hyper-parameter that controls how much to change the model in response to the estimated error each time the model weights are updated. [Link18] Choosing the learning rate is a challenging task because too small value can cause the system in a long training process and too large value can result in learning sub-optimal set of weight too fast or unstable training process. So, learning rate is the important parameter when configuring neural network model.

In this approach, LSTM-CNN-CRF model is trained for word segmentation, stemming and check the name entity across the different learning rate in the joint model. Due to the time constraints we did not perform every point of learning rate tuning. In this section, learning rate from 0.001 to 0.009 is tuned for LSTM-CNN-CRF model.

Table 6.12. The performance of joint model on different learning rate

Learning rate	Precision	Recall	F-Measure
0.001	88.16	87.48	87.82
0.002	89.44	89.36	89.40
0.003	91.36	91.14	**91.25**
0.004	90.42	89.18	89.80
0.005	89.75	91.40	90.56
0.006	90.40	89.39	89.89
0.007	90.98	90.08	90.53
0.008	87.55	87.58	87.56
0.009	90.59	90.14	90.37

In the experiment, learning rate hyper-parameter settings are evaluated. Then, the same setting is taken and tuned the learning rate. According to the experimental result, the selection of the learning rate has a large impact on the performance of the system. On most tasks, F1-score is around 89 and 90% by learning rate from 0.001 to 0.009. Learning rate 0.003 gives the best performance compare to the others. The worst learning rate is 0.008 that has F1-score 87.56. But the performance increase to 3% in learning rate 0.009. Learning rate 0.007 gives the second best performance.

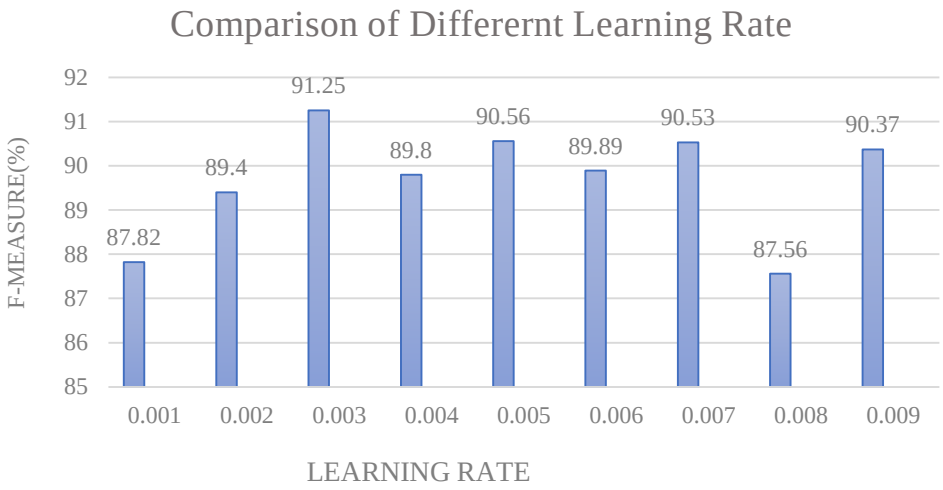

Figure 6.6 Comparison of Different Learning Rate

6.4 Error Analysis

Error analysis is also presented with related examples.

6.4.1 Named Entity Errors

In the phrase "**ဖျာပုံ, ဘိုကလေး, လပွတ္တာ သုံးမြို့နယ်**". The township names "**ဖျာပုံ**" and "**ဘိုကလေး**" is tagged correctly as named entity but in the last township name "**လပွတ္တာ**", it wrongly tagged as "**လပွတ္တာသုံး**" as named entity because of the named entity appears in front of the word "**မြို့**" "**မြို့နယ်**".

In the sentence "**စက်တင်ဘာ ၂၀ ရက် ၌ ပင် ဘူးသီးတောင် မြို့နယ်**", "**စက်တင်ဘာ**" is correctly tagged as named entity, "**၂၀**" is tagged as single word, "**ရက်**" is tagged as "root word", "**၌**" is correctly tagged as single word but the word "**ပင်**" is combined with "**ဘူးသီးတောင်**" and it is tagged "**ပင်ဘူးသီးတောင်**" as named entity.

But the township name "**ကျိုက်ထို မြို့နယ်**" is tagged as "{**ကျိုက်**}_**ထို**_[**မြို့နယ်**]", the word "**ကျိုက်**" is named entity but the word "ထို" is tagged as single word. In the training sentences, "**ထို**, **ယင်း**, **၎င်း**, **ဒီ**" are tagged as single word. So in this word "**ကျိုက်ထို**" "**ထို**" is tagged as single word even though it is in front of the word "**မြို့နယ်**".

The output result of the sentence "**ကွတ်ခိုင် မြို့ နှင့် နမ့်ဖတ်ကာ မြို့ တို့တွင်**" is "{**ကွတ်ခိုင်**}_[**မြို့**]_**နှင့်**_[**နမ့်ဖတ်**]+**ကာ**_[**မြို့**]_**တို့**_**တွင်**". Although the named entity "**ကွတ်ခိုင်**" is correctly tagged, the town name "**နမ့်ဖတ်ကာ**" is wrongly tagged as "**နမ့်ဖတ်**" is named entity and "**ကာ**" is suffix. It is because, the word "ကာ" is tagged as suffix in most word for example, "**ရည်ရွယ်ကာ, စတင်ကာ, ရှင်းလင်းကာ**".

The word "**ဧဝကလမ်း**" is the street name "**ဧဝက**" and it is wrongly tagged as "[**ဧဝ**]_**က**_[**လမ်း**]". In this case, the word "က" is wrongly tagged as postpositional marker. The correct result is "{**ဧဝက**}_[**လမ်း**]".

6.4.2 Root Word Errors

The result of the sentence "**ရင်ဆိုင် ကစားလိုခြင်း မရှိကြောင်း ပြောကြားသည် ။**" is "[**ရင်ဆိုင်**]_**က**_[**စားလို**]+**ခြင်း**_**မ**^[**ရှိ**]+**ကြောင်း**_[**ပြောကြား**]+**သည်**_။_". In this sentence, "က" is wrongly tagged as single word. The correct word is "**ကစား**". Moreover, the word "**လို**" is suffix but it combines with "**စား**" and wrongly produce "**စားလို**" as stem word. But the sentence "**ယှဉ်ပြိုင် ကစားလိုခြင်း မရှိကြောင်း ပြောကြားသည် ။**" is correctly produce "[**ယှဉ်ပြိုင်**]_[**ကစား**]+**လို**ခြင်း_**မ**^[**ရှိ**]+**ကြောင်း**_[**ပြောကြား**]+**သည်**_။_".

In the sentence "**ကလေး ကို အားဆေးတိုက်ကျွေးသည်။**", the output result is "[**ကလေး**]_**ကို**_**အား**_[**ဆေးတိုက်ကျွေး**]+**သည်**_။_". The word "**အား**" is wrongly tagged as single word and like postpositional marker but the actual word is "**အားဆေး**". However, the sentence "**ဘောလုံးပွဲ ကို အားပေးသည်။**" correctly tagged as "[**ဘောလုံးပွဲ**]_**ကို**_[**အားပေး**]+**သည်**_။_".

The word "**ချိုးရေတော်သုံးပွဲ**" is the stem word but it is wrongly tagged as "[**ချိုးရေတော်**]_**သုံး**+**ပွဲ**_။_". The word "**သုံး**" is ambiguous as number three and "**ပွဲ**" is ambiguous as type classifier particle.

And then, the result of the phrase "**အရှေ့ဘက် ဖို့မြေနေရာ**" is "[**အရှေ့**]_[**ဘက်**]+**ဖို့**_[**မြေနေရာ**]". In this case, "**ဖို့**" is wrongly tagged as particle. The right answer is "[**အရှေ့**]_[**ဘက်**]_[**ဖို့မြေ**]_[**နေရာ**]". Likewise, the word "**ခရီးသွားကောက်ကြောင်း**" is wrongly tagged as "[**ခရီးသွားကောက်**]+**ကြောင်း**" and the word "ကြောင်း" is wrongly tagged as particle. The correct output is "[**ခရီးသွားကောက်ကြောင်း**]". Furthermore, the output result of "**ဒီမိုကရေစီ ကျင့်စဉ်**" is "{**ဒီမိုကရေစီ**}_[**ကျင့်**]+**စဉ်**". In this case, the stem word "**ကျင့်စဉ်**" is wrong tagged as "[**ကျင့်**]+**စဉ်**".

The word "**သီလရှင် ဆရာလေးများ**" is wrongly tagged as "[**သီလရှင်**]_[**ဆရာ**]+**လေးများ**". The right answer is "[**သီလရှင်**]_[**ဆရာလေး**]+**များ**". The phrase "**ခုန်ပြီး နောက်ပြန် ကန်ချက်**" is wrongly tagged as "[**ခုန်**]+**ပြီး**_**နောက်**_[**ပြန်ကန်**]+**ချက်**". The correct output is "[**ခုန်**]+**ပြီး**_[**နောက်ပြန်**]_[**ကန်ချက်**]".

6.4.3 Simple Word Errors

The word "**ယင်းပြင်**" is wrongly tagged as "**ယင်း**_[**ပြင်**]". "**ထို့ပြင်**", "**ယင်းပြင်**" are conjunction and it must be tagged as single word and the word "**ပြင်**" is not stem word.

The phrase "**လက်ရှိ နိုင်တဲ့ ပါတီ**" is wrongly tagged as "**လက်ရှိ**+**နိုင်တဲ့**_[**ပါတီ**]". In this case, "**နိုင်**" is tagged as suffix. The right answer is "[**လက်ရှိ**]_[**နိုင်**]+**တဲ့**_[**ပါတီ**]". For example, the correct output of "**လက်ရှိ အခြေအနေ**" is "[**လက်ရှိ**]_[**အခြေအနေ**]".

The phrase "**နှမ်းနီ တစ်တင်းကျပ် ၄၃၀၀၀ ခန့်**" is wrongly tagged as "[**နှမ်းနီ**]_**တစ်**_[**တင်းကျပ်**]_**၄၃၀၀၀**_**ခန့်**". In this case, "**တစ်**" is numerical number and "**တင်း**" is type classifier. The correct output is "[**နှမ်းနီ**]_**တစ်**+**တင်း**_[**ကျပ်**]_**၄၃၀၀၀**_**ခန့်**_".

6.5 Summary

This chapter focuses on the experimental results of different network design and different hyper-parameter tuning on neural sequence labeling for joint word segmentation, stemming and Named Entity Detection. Normally, segmentation is considered as a separate process from stemming and Named Entity Detection. But, in this process segmentation is not standalone process and it is an integral part of stemming and Named Entity Detection. Thus, the proposed joint word segmentation, stemming and Named Entity Detection structure provides a powerful segmenter and detect stem words and named entity.

According to the experiment, LSTM-CNN-CRF model outperforms than others, and hyper parameters help to improve the model performance. In LSTM-CNN-CRF model, F-Measure is 87.80. When using our own embedding, performance increases to 89.40. And then, we tuned the different dropout rate and different optimizers. As reported by the experimental result, Optimizer SGD is the best optimizer and performance increases to 90.56 with dropout rate 0.2. Furthermore, dropout rate 0.2 gets the stable F-Measure in all optimizer. In Adadelta and RMSProp, dropout rate 0.3 gets the best result. Moreover, the performance of the model is upgraded by tuning the different learning rate. In accordance with the experimental results, performance increase to 91.25 F-Measure in learning rate 0.003. The contribution of this research is a measurement of which hyper parameters are important to optimize and which are of less importance.

CHAPTER 7
CONCLUSION AND FUTURE WORKS

In this chapter, the main contents of the thesis are summarized, advantages and limitation of the proposed system are also described and future work is suggested.

7.1 Thesis Summary

The objective of the research is to select the root words and named entity from Myanmar sentences using morphological stemming. Very less work has been done in Myanmar because of absence of resources such as Corpus, Ontology and WordNet etc. Retrieval system requires exact match of query word with words in the input documents. So, we cannot achieve effective results without extracting the root words from the input sentences.

Word segmentation is the very first problem in Myanmar language processing. It is because there are no indicators such as blank spaces to show the word boundaries in Myanmar text. The same phenomenon does not happen to only Myanmar language but also many other Asian languages such as Japanese, Chinese, and Thai. Therefore, in order to understand the Myanmar text, the first thing needed to do is to cut the sentences into word segments.

During the process of Myanmar word segmentation, two main problems are encountered: segmentation ambiguities are dealt with known words, i.e. words found in the dictionary. An unknown word is defined as a word that is not found in the system dictionary. In other words, it is an out-of-vocabulary word. For any languages, even the largest dictionary will not cover all geographical names, organization names, person names, technical terms and some duplication words. Name entity detection is one of the issues in Asian Language that has traditionally required large amount of feature engineering to achieve high performance.

This research is the initial effort to deal with Myanmar lexical analysis model that jointly accomplishes word segmentation, morphological stemming and named entity detection based on the concept of neural network architecture. This research considers morphological stemming as a typical sequence tagging problem over segmented word, while segmentation can also be modelled as a syllable-level tagging problem via predicting the labels that identify the word boundaries. This research

proposed a simple and effective neural sequence labeling model for joint Myanmar word segmentation, stemming and named entity detection. Moreover, it performs embedding as a preprocessing step in CNN-based model which learns character and syllable-level representation of syllables for Myanmar stemmer.

7.2 Advantages and Limitations of the Proposed System

The advantages of this system are as follows: Firstly, in Myanmar writing system, words are explicitly delimited with non-whitespace character. Since errors at the word segmentation stage directly affect all later processing stage, it is essential to completely address the issues. This research provides an effective word segmentation that do not require large dictionary or lexicon. Secondly, the very first morphological stemmer is proposed without using lexicon or rule. This stemmer can refer different forms of word into the common base form. Thirdly, it can identify Myanmar names such as Person name, Location and Organization without using name dictionary or lexicon. Fourthly, the customized tagset for Myanmar lexical analysis is identified and the first manually annotated lexical analysis corpus for Myanmar language is also constructed and proposed. It can be used in later NLP research.

This system is cheap, less amount of memory needed, saves time to accomplish three basic requirements for Myanmar NLP application. The cost is reduced since this system uses most of the open source softwares like Python and Pytorch.

The main limitation of the system is; sequence labeling model is used for joint word segmentation and stemming. It means that this research has not used the rule patterns. So, the stem word for some reduplicated words cannot be produced, for example, "**အ-တ**", "**မ-မ**", "**တ-တ**". This kind of reduplicated words are produced as a stem word, "**အရောတဝင်**", "**မပြောမဆို**", "**တရင်းတနှီး**".

7.3 Results and Discussion

In East Asian language including Myanmar, word segmentation is an initial step for NLP processes because there is no explicit word delimiter in those language. Stemming refers to the process of marking each word in word segmentation result with a correct word type, e.g. root word, prefix, suffix, etc. Named entity detection refers to detecting entities that have specific meanings in the identified text including

persons, locations, organization, etc. These lexical analysis approach is believed to be a crucial step towards natural language understanding. When building our lexical analysis system, the model works in a full end-to-end manner and turns out to be effective and efficient. Its input is character embedding, without any hand-crafted features. The model output tags according to a unified tag scheme with BIO style decoration, thus jointly accomplishes all three analysis tasks.

The model achieves 91.25% accuracy of both word boundaries and tags. To get the sufficient amount of data to train the LSTM-CNN-CRF model, a corpus from online news are constructed and this corpus is pre-labeled manually. Parameter optimization is performed using stochastic gradient descent (SGD) optimizer, Dropout rate is 0.2, embedding dimension is 600-d and learning rate 0.003. The batch size is 20 and iteration is 100.

According to the experiment, LSTM-CNN-CRF model outperforms than others and hyper parameters help to improve the model performance. In LSTM-CNN-CRF model F-Measure is 87.80%. When using our own embedding, performance increase to 89.40%. And then, the different dropout rate and different optimizers are tuned. As reported by the experimental result, Optimizer SGD is the best optimizer and performance increases to 90.56 with dropout rate 0.2. Furthermore, dropout rate 0.2 gets the stable F-Measure in all optimizer. Moreover, we upgrade the performance of the model by tuning the different learning rate. In accordance with the experimental results, performance increase to 91.25% F-Measure in learning rate 0.003.

Moreover, two running environment CPU and GPU are also compared. In neural sequence labeling approach, GPU is more efficient than CPU because it reduces the training time. We run on both Nvida Tesia K80 GPU sever and CPU intel Core i7. Firstly, the running speed is totally different. When using GPU, it takes 18hours for training. With CPU, it takes 80hours. So, running speed is highly affected by hardware environment. Secondly, the performance of the model between CPU and GPU is not very different. When using the CPU, performance decrease to nearly 1%. So, it can be concluded that hardware environment has a small impact on the performance of the system.

This research explores the effectiveness of neural network on Myanmar lexical analysis and conducted a systematic comparison between different architectures of neural network and different optimizers. This exploration of using neural networks for

Myanmar lexical analysis is the first work to apply neural network and joint lexical analysis approach.

7.4 Future Works

This research is the first attempt for joint word segmentation, stemming and named entity detection processes by using neural network architecture. It is necessary to increase the corpus size for training in order to get the better performance because the word also has mis-segmentation and disambiguation for stemming process. Moreover, it is needed to add the rules for some reduplication words.

For the future work, joint word segmentation process would like to use in further processing such as parsing, chunking and machine translation. Moreover, stemmer also uses in text summarization, information retrieval and text categorization processes.

AUTHOR'S PUBLICATIONS

[P1] Yadanar Oo, Khin Mar Soe "Joint Word Segmentation and Stemming for Myanmar Language Based on Conditional Random Fields", Proceeding of the 16th INTERNATIONAL CONFERENCE ON COMPUTER APPLICATIONS (**ICCA 2018**), Yangon, MYANMAR, February, 2018.

[P2] Yadanar Oo, Khin Mar Soe "Joint Word Segmentation and Stemming with Neural Sequence Labeling for Myanmar Language", 2019 The 11th International Conference on Future Computer and Communication (**ICFCC 2019**), Yangon, Myanmar, February 2019. Science and Engineering Institute, USA, co-organized by University of Computer Studies, Yangon. **(EI Compendex and Scopus)**

[P3] Yadanar Oo, Khin Mar Soe "Optimizer Comparison with Dropout for Neural Sequence Labeling in Myanmar Stemmer", The 2019 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (**IAICT, 2019**), Bali, Indonesia, June 2019.

[P4] Yadanar Oo, Khin Mar Soe "Better Pretrained Embedding with Convolutional Neural Networks for Morphological Stemming", 2019 3rd International Conference on Artificial Intelligence and Virtual Reality (**AIVR, 2019**), Singapore, Singapore July 2019. **ACM ICPS Program (ISBN: 978-1-4503-7161-2)**. (**Ei Compendex and Scopus**).

[P5] Yadanar Oo, Khin Mar Soe "Applying RNNs Architecture by Jointly Learning Segmentation and Stemming for Myanmar Language", 2019 IEEE 8th Global Conference on Consumer Electronics (**GCCE 2019**), OSAKA, Japan, October 2019.

[P6] Yadanar Oo, Khin Mar Soe, "Optimize The Learning Rate of Neural Architecture in Myanmar Stemmer" International Journal on Natural Language Computing (IJNLC), Vol 8, No 5, October, 2019.

BIBLIOGRAPHY

[1] A. Abu-Errub, A. Odeh, Q. Shambour, O.A. Hassan, "Arabic roots extraction using morphological analysis". International Journal of Computer Science Issues (IJCSI). Mar 2014.

[2] M.P. Aung, O. Aung, N.Y. Hlaing, "Proposed Framework for Stochastic Parsing of Myanmar Language". In International Conference on Big Data Analysis and Deep Learning Applications (pp. 179-187). Springer, Singapore. May 2018.

[3] M. Bacchin, N. Ferro, M. Melucci, "A probabilistic model for stemmer generation". Information Processing & Management. Jan 2005.

[4] K.S. Bajwa, A. Kaur, "Hybrid Approach for Named Entity Recognition". International Journal of Computer Applications. Jan 2015.

[5] A. Berlati, P.D. Tombari, "Ambiguity in Recurrent Models: Predicting Multiple Hypotheses with Recurrent Neural Networks".

[6] D. Bijal, S. Sanket, "Overview of stemming algorithms for Indian and Non-Indian languages". arXiv preprint arXiv:1404.2878. Apr 2014.

[7] J. Brownlee, "Deep Learning for Natural Language Processing". Machine Learning Mystery, Vermont, Australia. 2017.

[8] V. Chea, Y.K. Thu, C. Ding, M. Utiyama, A. Finch, E. Sumita. "Khmer word segmentation using conditional random fields". Khmer Natural Language Processing. Dec 2015.

[9] J.P. Chiu, E. Nichols, "Named entity recognition with bidirectional LSTM-CNNs". Transactions of the Association for Computational Linguistics. Dec 2016.

[10] G. Choy, O. Khalilzadeh, M. Michalski, S. Do, A.E.Samir, O.S. Pianykh, J.R. Geis, P.V. Pandharipande, J.A.Brink, K.J. Dreyer, "Current applications and future impact of machine learning in radiology". Radiology. Jun 2018.

[11] C. Ding, Y.K. Thu, M. Utiyama, E. Sumita. "Word segmentation for burmese (Myanmar)". ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP). Jun 2016.

[12] C. Ding, Y.K. Thu, M. Utiyama, E. Sumita. "Word segmentation for

burmese (Myanmar)". ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP). Jun 2016.

[13] J. Duchi, E. Hazan E, Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization". Journal of Machine Learning Research. Jul 2011.

[14] A. Ekbal, R. Haque, S.Bandyopadhyay, "Named entity recognition in Bengali: A conditional random field approach". In Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II 2008.

[15] Z. Ghahramani, "An introduction to hidden Markov models and Bayesian networks". In Hidden Markov models: applications in computer vision. (pp. 9-41). 2001.

[16] S.S. Govilkar, J.W. Bakal, S.R. Kulkarni, "Extraction of root words using morphological analyzer for devanagari script". International Journal of Information Technology and Computer Science (IJITCS). 2016.

[17] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, T. Mikolov, "Learning word vectors for 157 languages". arXiv preprint arXiv:1802.06893. Feb 2018.

[18] A.Z. Gregoric, Y. Bachrach, S. Coope, "Named entity recognition with parallel recurrent neural networks". In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (pp. 69-74). Jul 2018.

[19] C. Haruechaiyasak, S. Kongyoung, M. Dailey. "A comparative study on thai word segmentation approaches". In 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology. May 2008 (Vol. 1, pp. 125-128). IEEE.

[20] C. Haruechaiyasak, S. Kongyoung, M. Dailey, "A comparative study on Thai word segmentation approaches". In 2008 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology. (Vol. 1, pp. 125-128) IEEE. May 2008.

[21] P. Hemanth, Adarsh, C.B. Aswani, P. Ajith and A. Veeena Kumar "EMO PLAYER: Emotion Based Music Player". International Research Journal of Engineering and Tedchnology (IRJET). Volume:5 Issue:4. Apr 2018.

[22] H.H. Htay, K.N. Murthy, "Myanmar word segmentation using syllable level

longest matching". In Proceedings of the 6th Workshop on Asian Language Resources 2008.

[23] H.H. Htay, K.N. Murthy, "Myanmar word segmentation". In Proc. 4th International Conf. on Computer Application (ICCA). (pp. 353-357). 2006.

[24] Z. Huang, W. Xu, K. Yu. "Bidirectional LSTM-CRF models for sequence tagging." arXiv preprint arXiv:1508.01991. Aug 2015.

[25] J.P. Jayan, R.R. Rajeev, S. Rajendran. "Morphological analyser and morphological generator for Malayalam-Tamil machine translation". International Journal of Computer Applications. Jan 2011.

[26] Z. Jiao, S. Sun, K. Sun, "Chinese Lexical Analysis with Deep Bi-GRU-CRF Network". arXiv preprint arXiv:1807.01882. Jul 2018.

[27] D.P. Kingma, J. Ba, "Adam: A method for stochastic optimization". arXiv preprint arXiv:1412.6980. Dec 2014.

[28] J. Lafferty, A. McCallum, F.C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data". In Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[29] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C, Dyer, "Neural architectures for named entity recognition". arXiv preprint arXiv:1603.01360. Mar 2016.

[30] T.M. Latt, A. Thida. "An Analysis of Myanmar Inflectional Morphology Using Finite-state Method". In IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS), (pp. 297-302). IEEE. Jun 2018.

[31] S. Li, C.R. Huang. "Word boundary decision with CRF for Chinese word segmentation." In Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation, Volume 2, (pp. 726-732). Dec 2009.

[32] C.Y. Lin, N. Xue, D. Zhao, X. Huang, Y. Feng, editors. "Natural Language Understanding and Intelligent Applications": 5th CCF Conference on Natural Language Processing and Chinese Computing, NLPCC 2016, and 24th International Conference on Computer Processing of Oriental

Languages, ICCPOL 2016, Kunming, China, December 2–6, 2016, Proceedings. Springer; Nov 2016.

[33] J.B. Lovins, “Development of a stemming algorithm”. Mech. Translat. & Comp. Linguistics. Mar 1968.

[34] T. Luong, R. Socher, C. Manning, “Better word representations with recursive neural networks for morphology”. In Proceedings of the Seventeenth Conference on Computational Natural Language Learning. (pp. 104-113). Aug 2013.

[35] X. Ma, E. Hovy. “End-to-end sequence labeling via bi-directional lstm-cnns-crf.” arXiv preprint arXiv:1603.01354. Mar 2016.

[36] P. Majumder, M. Mitra, S.K. Parui, G. Kole, P. Mitra, K. Datta, “YASS: Yet another suffix stripper”. ACM transactions on information systems (TOIS). Oct 2007.

[37] A. Mansouri, L.S. Affendey, A. Mamat, “Named entity recognition approaches”. International Journal of Computer Science and Network Security. Feb 2008.

[38] X. Mao, Y. Dong, S. He, S. Bao, H. Wang, “Chinese word segmentation and named entity recognition based on conditional random fields.” In Proceedings of the Sixth SIGHAN Workshop on Chinese Language Processing 2008.

[39] U. Mishra, C. Prakash, “MAULIK: an effective stemmer for Hindi language”. International Journal on Computer Science and Engineering. May 2012.

[40] J. Mu, S. Bhat, P. Viswanath. “All-but-the-top: Simple and effective postprocessing for word representations.” arXiv preprint arXiv:1702.01417. Feb 2017.

[41] A.M. Mustafa, T.A. Rashid. “Kurdish stemmer pre-processing steps for improving information retrieval.” Journal of Information Science. Feb 2018.

[42] P.H. Myint, T.M Htwe, N.L. Thein, “Bigram part-of-speech tagger for Myanmar language.” In Proceedings of 2011 International Conference on Information Communication and Management, Singapore (pp. 147-152), Oct 2011.

[43] T. Myint, A. Thida, “Name entity recognition and transliteration in

Myanmar text". PhD Research, University of Computer Studies, Mandalay. May 2014.

[44] C.T. Nguyen, T.K. Nguyen, X.H. Phan, L.M. Nguyen, Q.T. Ha. "Vietnamese word segmentation with CRFs and SVMs: An investigation." In Proceedings of the 20th Pacific Asia Conference on Language, Information and Computation. (pp. 215-222). Nov 2006.

[45] C. Özgen, "Evaluation of Performance and Optimum Valve Settings for Pressure Management Using Forecasted Daily Demand Curves by Artificial Neural Networks". (Doctoral dissertation, MIDDLE EAST TECHNICAL UNIVERSITY).

[46] W.P Pa, N.L Thein, "Myanmar word segmentation using hybrid approach". In Proceedings of 6th International Conference on Computer Applications, Yangon, Myanmar (pp. 166-170), 2008.

[47] W.P. Pa, Y.K. Thu, A. Finch, E. Sumita. "Word boundary identification for Myanmar text using conditional random fields." In International Conference on Genetic and Evolutionary Computing (pp. 447-456). Springer, Cham. Aug 2015.

[48] S. Parvez, "NAMED ENTITY RECOGNITION FROM BENGALI NEWSPAPER DATA".

[49] M.F. Porter, "An algorithm for suffix stripping". Program. Jul 2006.

[50] E. Rahimtoroghi, H. Faili, A. Shakery. "A structural rule-based stemmer for Persian". In2010 5th International Symposium on Telecommunications (pp. 574-578). IEEE. Dec 2010.

[51] N. Reimers, I. Gurevych. "Optimal hyperparameters for deep lstm-networks for sequence labeling tasks." arXiv preprint arXiv:1707.06799. Jul 2017.

[52] Y. Roh, G.Heo, S.E. Whang, "A survey on data collection for machine learning: a big data: AI integration perspective". arXiv preprint arXiv:1811.03402. Nov 2018.

[53] S. Ruder, "An Overview of Gradient Descent Optimization Algorithms". arXiv preprint arXiv:1609.04747. Sep 2016.

[54] A. Rugchatjaroen, S. Saychum, S. Kongyoung, P. Chootrakool, S. Kasuriya, C. Wutiwiwatchai. "Efficient two-stage processing for joint sequence model-based Thai grapheme-to-phoneme conversion. Speech

Communication". Jan 2019.

[55] M. Sato, H. Shindo, I. Yamada, Y. Matsumoto, "Segment-level neural conditional random fields for named entity recognition". In Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers) (pp. 97-102). Nov 2017.

[56] M. Schuster, K.K. Paliwal, "Bidirectional recurrent neural networks". IEEE Transactions on Signal Processing. Nov 1997.

[57] Y. Shao, C. Hardmeier, J. Tiedemann, J. Nivre. "Character-based joint segmentation and POS tagging for Chinese using bidirectional RNN-CRF." arXiv preprint arXiv:1704.01314. Apr 2017.

[58] D. Sharma, M. Cse. "Stemming algorithms: a comparative study and their analysis". International Journal of Applied Information Systems. (pp. 7-12) Sept 2012.

[59] N. Sobhana, P. Mitra, S.K. Ghosh, "Conditional random field based named entity recognition in geological text". International Journal of Computer Applications. Feb 2010.

[60] S. Song, N. Zhang, H. Huang. "Named entity recognition based on conditional random fields". Cluster Computing. Sep 2017.

[61] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting". The journal of machine learning research. Jan 2014.

[62] T.T. Swe, H.H. Htay, "A Hybrid Methods for Myanmar Named Entity Identification and Transliteration into English", 2010.

[63] T.T. Thet, J.C. Na, W.K Ko, "Word segmentation for the Myanmar language." Journal of information science. (pp. 688-704) Oct 2008.

[64] T. Tieleman, G. Hinton, Lecture 6.5—RmsProp: Divide the Gradient by a Running Average of its Recent Magnitude. COURSERA: Neural Networks for Machine Learning, 2012. TRITA-MAT-E 2017: 81 ISRN-KTH/MAT/E--17/81--SE; 2012.

[65] J. Yang, S. Liang, Y. Zhang. "Design challenges and misconceptions in neural sequence labeling." arXiv preprint arXiv:1806.04470. Jun 2018.

[66] J. Yang, Y. Zhang, "NCRF++: An Open-source Neural Sequence Labeling Toolkit". arXiv preprint arXiv:1806.05626. Jun 2018.

[67] M. D. Zeiler, “ADADELTA: An Adaptive Learning Rate Method”. arXiv preprint arXiv:1212.5701. Dec 2012.

[68] M. Zhang, N. Yu, G. Fu. “A simple and effective neural model for joint word segmentation and POS tagging”. IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP). 26(9):1528-38, Sep 2018.

[69] H. Zhao, C.N. Huang, M. Li. “An improved Chinese word segmentation system with conditional random field.” In Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing. (pp. 162-165) Jul 2006.

[70] H. Zhao, C.N. Huang, M. Li, B.L. Lu. “Effective tag set selection in Chinese word segmentation via conditional random field modeling.” In Proceedings of the 20th Pacific Asia Conference on Language, Information and Computation. (pp. 87-94). 2006.

[71] “The 7 Steps of Machine Learning”, https://towardsdatascience.com/the-7-steps-of-machine-learning-2877d7e5548e

[72] “Datasets and Machine Learning”, https://skymind.ai/wiki/datasets-ml

[73] “Machine Learning Algorithms: Which One to Choose for Your Problem” , https://blog.statsbot.co/machine-learning-algorithms-183cc73197c

[74] "What is Semi-Supervised Learning?", https://www.datascience.com/blog/what-is-semi-supervised-learning

[75] “Machine Learning vs. Deep Learning” , https://dzone.com/articles/comparison-between-deep-learning-vs-machine-learni

[76] “Laphet-waing”, http://myanmar-teacircle.blogspot.com/2010/03/grammar-terms.html

[77] “Artificial Neural Networks (ANN) and Different Types” , https://www.elprocus.com/artificial-neural-networks-ann-and-their-types/

[78] “Neural Networks: What does the input layer consist of?”, https://stackoverflow.com/questions/32514502/neural-networks-what-does-the-input-layer-consist-of

[79] “Hidden Layer”, https://www.techopedia.com/definition/33264/hidden-layer-neural-networks

[80] “Convolutional Neural Network , https://searchenterpriseai.techtarget.com/definition/convolutional-neural-

network

[81] "Introduction to Recurrent Neural Network", https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/

[82] "All of Recurrent Neural Networks", https://medium.com/@jianqiangma/all-about-recurrent-neural-networks-9e5ae2936f6e

[83] "Illustrated Guide to LSTM's and GRU's: A step by step explanation", https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

[84] "How to evaluate model performance in Azure Machine Learning Studio", https://docs.microsoft.com/en-us/azure/machine-learning/studio/evaluate-model-performance

[85] "What Are Word Embeddings for Text?", https://machinelearningmastery.com/what-are-word-embeddings/

[86] "word2vec", https://code.google.com/archive/p/word2vec/

[87] "What Are Word Embeddings for Text?" , http://computer-trading.com/what-are-word-embeddings-for-text/

[88] "Understand the Impact of Learning Rate on Neural Network Performance", https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/

[89] "A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way", https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

LIST OF ACRONYMS

Adam	Adaptive Moment Estimation
AI	Artificial Intelligent
ALT	Asian Language Treebank
ANN	Artificial Neural Network
BLSTM	Bi-directional LSTM
CBOW	Continuous Bag of Words
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CRF	Conditional Random Fields
CW	Compound Word
DCB	Dictionary Based
FF	Feed Forward
GPU	Graphics Processing Unit
GRU	Gated Recurrent Unit
HMM	Hidden Markov Model
IR	Information Retrieval
LSTM	Long Short Term Memory
MEMM	Maximum Entropy Markov Model
MLB	Machine Learning Based
MLP	Multilayer Perceptron
NB	Naive Bayes
NE	Named Entity
NER	Named Entity Recognition
NLM	Neural Language Model

NLP	Natural Language Processing
NN	Neural Network
OOV	Out of Vocabulary
POS	Part of Speech
ReLU	Rectified Linear Unit
RMSProp	Root Mean Square Propagation
RNN	Recurrent Neural Network
SGD	Stochastic Gradient Descent
SLP	Single Layer Perceptron
SVM	Support Vector Machine
SW	Single Word
WBD	Word Boundary Decision