SYLLABLE-BASED NEURAL NAMED ENTITY RECOGNITION FOR MYANMAR LANGUAGE

HSU MYAT MO

UNIVERSITY OF COMPUTER STUDIES, YANGON

AUGUST, 2019

Syllable-based Neural Named Entity Recognition for Myanmar Language

Hsu Myat Mo

University of Computer Studies, Yangon

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

Doctor of Philosophy

August, 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 Hsu Myat Mo

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. Nwe Nwe Win, Vice-President, Myanmar Computer Federation (MCF), 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 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 them.

ABSTRACT

More and more information is being created at online every day, and a lot of it is the natural language. Until recently, businesses have been unable to analyze this data. But advances in Natural Language Processing (NLP) make it possible to analyze and learn from a greater range of data sources. Additionally, NLP has many central implications on the ways that computers and humans network on our daily life. By promising a bridge between human and machine, and accessing stored information, NLP plays a vital role in the multilingual society. Technologies constructed on NLP are becoming increasingly widespread.

Named Entity Recognition (NER), the task of recognizing names in text and assigning those recognized Named Entities (NEs) to particular NE types such as person name, location or organization, is a key component in many sophisticated systems, especially in information retrieval (IR) systems. NER for Myanmar language is essential for the development of Myanmar NLP and it is not an easy task for many reasons.

This dissertation aims to develop Named Entity Recognition (NER) for Myanmar language as well as to promote Myanmar NLP research. Myanmar NLP is said to be still developing and has now been struggling to be developed. In the same situation, there are no publicly available resources that can be accessed freely or commercially for language computation so that Myanmar is being regarded as low-resourced language. For this reason, named entity (NE) tagged corpus for Myanmar NER research is manually annotated and constructed as part of this dissertation. The annotated NE corpus is essential for the development of Myanmar NER research. This NE tagged corpus is applied during all the conducted experiments for Myanmar NER and it will also be provided for future NER research.

In written style of Myanmar language, there is no regular space between words or phrases. In Myanmar language, syllables are the basic units. Thus, all the experiments are conducted on syllable-level data instead of characters or words in this work.

In this study, NER for Myanmar language is built by applying deep neural network architecture which can be said that Long Short-Term Memory (LSTM) - based network. The performance of neural model is also compared with baseline statistical Conditional Random Field (CRF) model. This statistical model totally

depends on feature engineering. As Myanmar language is low-resourced language, named dictionary or gazetteers are not available. If these external feature resources are available and feature engineering is carefully done based on knowledge to cover all situations, statistical methods provide a superior result. In this work, it has been proved that unless using additional features, deep neural networks work well on Myanmar NER and outperform baseline statistical CRF model. The best accuracy is achieved with bidirectional LSTM based network architecture. Therefore, this work eliminates the feature-engineering process and does not need to have language or domain knowledge.

The proposed syllable-based neural architecture for Myanmar NER model has three main layers: a character sequence layer, a syllable sequence layer, and inference layer. For each input syllable sequence, syllables are represented with their syllable embeddings. The character sequence layer is used to automatically extract syllable level features by encoding the character sequence within the syllable. Convolutional Neural Network (CNN) is applied to learn character sequence feature within each input syllable at character sequence representation layer. The syllable sequence layer takes the syllable representations as input and extracts the sentence level features, which are fed into the inference layer. For the syllable sequence representation, bidirectional LSTM is utilized to learn sentence level feature, and then CRF inference layer is jointly added above the network to tag the name labels. This proposed CNN_BiLSTM_CRF neural model gives the best performance out of the conducted experiments for the Myanmar NER.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

CHAPTER 1
INTRODUCTION

Named Entity Recognition (NER) is the task of identifying and classifying the Named Entities (NEs) from the plain text into pre-defined named entity categories. In other words, it aims to recognize words which are being used as NE in a given context and assign those recognized NEs to particular types of NE categories. NER is a key component in NLP systems for automatic questions and answering systems, information retrieval, relation extraction, summarization, anaphora, document organization or classification, automatic indexing, machine translation, etc. Therefore, robust handling of proper names and NER is essential for many applications.

In reality, NER is not an easy task for many reasons. Variations of NEs and ambiguity of NE types as well as ambiguity with common words are common issues that are usually encountered when addressing the NER problem. Not only these problems, but also the language style, structure, formatting, domain, and genre, etc., all have impacted on performance of NER.

Being a fundamental task and an essential part in information extraction, NER has got constant research attention over recent years. The NER for Myanmar language is absolutely necessary to textual language processing for Myanmar language. The issue of recognizing proper names in Myanmar text automatically is more complicated than other languages such as English; and it has been a challenging issue for many reasons. Additionally, well-prepared linguistic resources required for Myanmar NLP research have not been available sufficiently until now. Annotated corpora are a vital resource for NLP and information extraction approaches which employ machine learning techniques. Annotated corpora for Myanmar language are limited and scattered. This is one of the main reasons why Myanmar NLP lagged behind compared to others.

As part of this research, manually prepared and annotated NE tagged corpus for Myanmar language is proposed to address resource limitation problem. Currently, there are totally over 60,000 sentences and over 174,000 NEs in this manually annotated NE tagged corpus. This NE tagged corpus is developed with the intension of providing resource for future NER research. This NE corpus can also be modified or updated with more data and entity types.

NER problem has been addressed by three commonly known approaches: rule-based of knowledge-based approach, statistical or machine learning approach, and hybrid approach. Knowledge-based NER approaches do not require annotated training data corpus as they rely on lexicon resources and domain specific knowledge. These approaches work better when the lexicon is large enough to cover all possible occurrences of names. When NER models are based on rules defined by experts or linguistics, it may suffer from small coverage of rules and defining rules is expensive.

On the other hand, NER task has been solved as a sequence labeling problem, where entity boundary and category labels are jointly predicted. Sequence labeling is the task that involves the assignment of a categorical label to each member of a sequence of observed values by making use of algorithmic calculation.

Even though statistical sequence models rely on no complex rules but on human knowledge and feature engineering, and offer better performance than rules. The dependence upon hand-crafted features and task specific resources makes the model difficult to adapt to new tasks or to shift to new domain. Besides, designing effective features is still a labor-intensive and skill-dependent task. However, neural sequence models do not rely on rules or handcrafted features but need only large training data and can automatically learn features.

Neural network models have the ability to remove the burden of statistical models which needs to work with effective feature selection, because deep layers in neural networks can automatically learn relevant features to tasks. The benefits of neural networks on sequence labeling had been explored by Collobert et al. [16]. Subsequent to this effort, various recurrent neural networks (RNN) modifications have been used in sequence data modeling; and these networks have been revealed to be quite beneficial.

Long short-term memory (LSTM) neural network, a special kind of recurrent neural network (RNN), has been verified to be robust and quite effective in sequential data modeling. Furthermore, bidirectional LSTM network has made great enhancement in linguistic computation because of its ability which can retain information for long periods of sequence in both directions. The reason is that bidirectional LSTM neural network is the establishment of two independent LSTM layers so that it can accumulate contextual information from both the left and the right data.

With the speedy evolution of deep learning, several recent research has deploy a Conditional Random Field (CRF) layer by jointly adding above a bidirectional LSTM network to form a combined bidirectional LSTM network and statistical Conditional Random Field (CRF). Such a kind of network architecture can make use of the input features in the previously, tag information at sentence level as well as the upcoming input features. With such kind of deep neural networks, it has been found out that deep neural networks significantly outperform statistical algorithms.

Recent neural architectures for NER can be roughly classified into categories in accordance with their representation of the tokens (words) in a sentence. For instance, representations may be learnt via on characters, sub-word units, words, or any other combination of these. In word level architecture, words in a sentence are given as input to the networks, and then each word is represented by its word embedding, whereas a sentence is taken to be a sequence of characters and this character sequence is passed through the networks in character level architecture.

In this proposed neural architecture for Myanmar NER, syllables are applied as basic input unit to the network, thus a sequence of syllables in a sentence is taken as input and passed into the network. As a consequence, it eliminates the need of word segmentation which can lead to the wrong segmentation. When NE boundary does not match with the word boundary, a boundary conflict problem may probably be happened because of word segmentation.

In this dissertation, a syllable-based deep neural network architecture for Myanmar NER is proposed. Different combinations of deep neural network architectures are tested and investigated the power of deep neural networks on Myanmar NER. According to the experimental results, the proposed neural model without any additional features provides better performance than baseline statistical CRF model. Although the manually annotated NE tagged corpus is not as big as corpora of other languages, the proposed neural architecture works well on these data and provide superior results.

1.1 Problem Statement

Named entity recognition is a challenging and elaborated task that has typically needed large amounts of linguistic knowledge and resources in the form of

features, lexicons and gazetteers to achieve high performance. NER task for Myanmar language is complicated for many reasons.

The deficiency of linguistic resources such as annotated corpora, prepared name lists and also name dictionaries or gazetteers, is the main issue in resolving NER for Myanmar language. At the present time, Myanmar NLP is struggling to be developed but lexical resources available are very insufficient.

As one of the distinct characteristics of Myanmar language, its morphology is extremely rich and complex and even ambiguous. Besides, Myanmar language has no capitalizing feature that indicates proper names in some other European languages like English. Further, its writing structure has no definite order and it also makes the NER a complicated process.

Because of the fact that some proper names are loanwords or transliterated words, there are wide alternations in some Myanmar spelling for these words. Names in Myanmar texts also take all morphological inflections so that it can be ambiguous in classifying NEs into predefined categories.

Due to these facts, when Myanmar NER is approached with rule-based approach, it would be a difficult problem and there may probably need huge numbers of rules that cover all these facts. Likewise, feature engineering needs to be carefully prepared for statistical approaches and these approaches will be rely on human knowledge.

Since Myanmar is a morphologically rich language, it is necessary to deal with out-of-vocabulary (OOV). Moreover, when words are considered as basic training unit for distributed representation in model training, OOV problem may probably be happened. Additionally, word segmentation process is necessary to detect words boundary in written Myanmar text. Therefore, segmentation result will affect the NER performance. For these points, syllable, the smallest linguistic unit which can bring information about word, is considered as the basic input unit for distributed representation for NE label tagging in all our NER experiments.

Due to the lack of resource and its language nature, it can be said that how to accomplish the task of recognizing names in Myanmar scripts automatically is still difficult to handle. According to the mentioned problems, Myanmar NER is still necessary to develop in the area of Myanmar NLP research and it should be performed by using state-of-the-art approach.

1.2 Motivation of the Research

One of the reasons why we try to solve this NER problem is to provide NER model to integrate to other NLP research and applications. Because the result from NER can be applied into other sophisticated NLP works such as Myanmar-English machine translation system, summarization and recommendation system, Information Retrieval (IR) system, etc. Furthermore, the main motive for this work is that, currently, there is no available NER tool which can extract NEs in written Myanmar texts.

Moreover, even though there are many benchmark data resources for other languages, as far as it has been concerned, there is no publicity available NE tagged corpus. The resource corpus is vital while conducting experiments to address this NER problem. For this reason, a very first Myanmar NE tagged corpus was manually annotated with the defined NE tags and constructed and proposed.

Another point is that the previous studies on Myanmar NER had mainly focused on dictionary-based and statistical approaches which require careful feature engineering and linguistic knowledge. To eliminate these requirements, the effectiveness of neural sequence label modeling on Myanmar NER has been investigated. No other work has been published for applying deep neural networks architectures on Myanmar NER. Therefore, this research is only for the purpose of NER research development in Myanmar language.

1.3 The Objectives of the Research

Myanmar NLP is at developing state when compared to that of other countries. Every nation has been trying to develop their language technology. It is hoped that this work will be helpful in development of Myanmar NLP research work. The main objective of this research is to provide a good quality NER model for Myanmar language. Moreover, it is intended to address resource limitation problem in language computation because resource deficiency is one of the main barriers to develop NLP research.

The other objectives of this research area are as follows:

(i) To make available NE tagged corpus for future research

(ii) To meet the need to resource deficiency in Myanmar NER

(iii) To provide a good quality NER model for Myanmar language
(iv) To reduce the need of expensive additional feature engineering
(v) To introduce a way to automatically induce NEs
(vi) To discover the effectiveness of deep learning on Myanmar NER
(vii) To get the highly advanced neural models for Myanmar NER
(viii) To develop NER tool for Myanmar language

1.4 Focus of the Research

This research focuses on developing a new NE tagged corpus for Myanmar language and its use in neural modeling for Myanmar NER. These focused works include the following:

(i) Examining the previous approaches to Myanmar NER and their natures
(ii) Studying the existing approaches to NER for different languages
(iii) Learning the syllable structure of Myanmar language
(iv) Developing manually annotated NE tagged corpus for Myanmar language
(v) Learning NER problem as sequence learning problem
(vi) Developing a baseline CRF-based statistical NER model for Myanmar language
(vii) Investigating the effectiveness of different neural architectures for Myanmar NER
(viii) Introducing a deep neural architecture for Myanmar NER modeling
(ix) Evaluating the generated neural NER model for Myanmar language and comparing the results with the baseline statistical CRF model
(x) Proving syllable-based neural model for Myanmar language

1.5 Contributions of the Research

The very first manually annotated NE tagged corpus for Myanmar language was also contributed to make use of it during experiments, and to provide resource for future research in Myanmar NER. There is no other available corpus that has as much data as this manually prepared NE tagged corpus for Myanmar language. Developing NE tagged corpus is essential and it is very important for Myanmar NER modeling. Data in this corpus contains news data from various online official news websites.

Most recent approaches to NER have been characterized by use of traditional statistical CRF, support vector machine (SVM), and perceptron models, where performance is heavily dependent on the design of effective feature engineering. To my best knowledge, the previous Myanmar NER works generate NE recognized outputs which are based on rule-based and statistical approaches.

In this work, Myanmar NER problem is solved by means of deep neural network modeling and it is considered as sequence labeling problem.

The proposed neural model is implemented by representing syllables as a combination of syllable embedding with CNN over the characters of the syllables, following this with a bidirectional LSTM layer over the syllable representations of a sentence, and finally using a CRF layer above the bidirectional LSTM to generate labels.

In this neural model architecture, a sentence is taken to be a sequence of syllables and syllables are treated as basic input unit to the networks rather than characters or words.

The proposed deep neural architecture for Myanmar NER does not use any other additional feature engineering rather than training corpus. The main contribution lies in building NER model with deep neural network architectures for the Myanmar NER task. Therefore, it can be said that this effort contributes the very first evaluation of neural network models on NER task for Myanmar language.

To sum up, there are three primary contributions of this dissertation:

(i) NE tagged corpus for Myanmar language is manually annotated and developed.

(ii) Syllables are considered as basic input tokens.

(iii) Deep neural network architecture is constructed for Myanmar NER modeling.

1.6 Organization of the Research

This dissertation is organized with seven chapters, including introduction of NER, problem statements, objectives, focuses and contributions of the research work. Chapter 2 discusses the background of NER, the important factors in NER and the different approaches to NER problem in literature that are dealing with the dissertation. The theory background of deep neural networks and conditional random

fields, the differences in natures and components between them are described in Chapter 3. The state of Myanmar language and Myanmar NER is presented in Chapter 4. In Chapter 5, the overall work flow of entire proposed neural architecture is described, including corpus building, preprocessing and also experimental setup. The proposed syllable-based neural architecture for Myanmar NER is also discussed in detail. Chapter 6 describes the evaluation of the experimental results by comparing with baseline CRF models, and analysis of the evaluation among different experiments. Finally, Chapter 7 concludes with this research work and depicts the future research lines to continue it.

CHAPTER 2
LITERATURE REVIEW

Named Entity Recognition (NER) is the hub for many tasks related to Information Extraction. As the massive amount of unstructured text data are available from different sources today, a rich source of information is easier to get when the unstructured data can be structured. Nowadays, news, online media, and publishing houses are generating large amounts of data content on a daily basis. Therefore, managing those data correctly is very important to get the most beneficial information. When managing those huge amount of text data, being able to explore and browse them by the people, places and time mentioned in those texts becomes an essential feature. NER can provide meaningful relations between different documents, news and articles by establishing references to the same entity found in those different documents, news and articles.

NER becomes a core subtask to build large knowledge bases from unstructured and semi-structured texts from various data sources. Data scientists and developers have worked and researched on big data that contain millions of entities and hundreds of millions of facts about them and build useful large knowledge bases. Those knowledge bases are key contributors to several popular technologies such smart assistant, search engines and deep interpretation of natural language.

Despite search engines and main contributor of most AI technologies, NER is the main supporter behind recommendation systems, navigation systems and customer support systems. Moreover, NER plays as vital pre-processing for many NLP research works such as summarization, information retrieval, anaphora, document classification, content categorization, automatic indexing of books, machine translation and information filtering and so on. Not surprisingly, robust handling of proper names is essential for many applications. In research area of NLP, NER is an important foundation in order to extract relevant information.

To address NER problem, many researchers have tried with multiple approaches. While early NER works were making use of linguistic technique, today works mostly apply machine learning algorithms. In fact, there are three major approaches towards to NER: linguistic approach, machine learning or statistical approach, and hybrid approach. Moreover, over the past few years, numerous deep

neural architectures were applied to NER problem with the popularity in eliminating of feature engineering.

This chapter describes the literature review on NER research, and also on different approaches to NER. Firstly, dictionary look-up approach, the simplest approach, used in earlier days is explained and its usage in NER is described. And then ruble-based approach to NER and related researches are presented. After that statistical approach followed by hybrid approach is briefly explained. Finally, deep learning methods to NER are initiated and NER is discussed with related neural learning research.

2.1 Named Entity Recognition

Named Entity Recognition (NER) refers to data extraction task that is capable of finding names from content and can classify the category in which the name belongs. The term “Named Entity” was evolved for the Sixth Message Understanding Conference (MUC-6) [98]. At that time, MUC was focusing on Information Extraction (IE) tasks and people became aware of the essential to identify units of information like names, including person, organization and location names, and numeric expressions which represent time, date and percent expressions. Identifying these entities became a core task of IE and NER was acknowledged.

With a large focus on information extraction, it was found essential to do much research on NER. Since then, named entity recognition and classification research had been carried out using knowledge engineering as well as machine learning approaches and it had been accelerated with steady researches and numerous scientific events such as the Conference on Natural Language Learning (CONLL) until now. The Language Resource and Evaluation Conference (LREC) has also been staging workshops and main conference tracks on the NER topic since 2000 [55].

2.2 Related Factors

Along with the task of NER was given attention by the community research, some related factors came up that need to be considered while working in this field. Among these factors, the very important factors are the language, domain or textual genre and entity types that are being looked for.

2.2.1 Language Factor

One of the most dominant factors while doing research on NER is the language. Early attempts on NER were built based on rules for specific language so that it was impossible to alter these systems to different languages easily. On the contrary, with machine learning, it was possible to choose features independent on the language and can switch from language to language. The performance of NER system is significantly affected by the language that are working with.

The majority of NER research has been conducted for English. Language independence and multilingual become major challenge in this field. Japanese had been given attention since MUC-6 conference and Garman was well studied in CONLL-2003 and in prior studies. Likewise, languages like Dutch and Spanish are emphasized by a major devoted conference CONLL-2002. French, Italian and Greek have been widely studied and also Chinese has been explored in an abundant literature. Many other languages such as Korean, Hindi, Bulgarian, Polish, Romanian, Russian, Swedish, Turkish, and Portuguese got attention as well. Interest on Asian languages: Thai, Indonesia, Malaysia, Vietnamese, Laos and Indian, has been gaining in last decades and survey on these languages are in progress. A lot research has been done on even Arabic and Mongolian. Research on Myanmar NER is started to receive attention and just beyond the initial stage.

2.2.2 Domain Factor

Domain adaptation is another influential problem in NER. Besides, the domain (business, tourism, health, sports, religion and education, etc.,) and type of the text genre (news, informal, spoken style text, short message and scientific text) of corpora that are trained on can highly affect the performance of NER. One fundamental cause of performance degradation is when the domains and the text genres are slightly different. Few studies are specifically emphasized to diverse domains and text genres. Their experiments revealed that despite reasonable support to any domain, swapping the corpora of one domain with another still remains as a major challenge.

The increasing flood of user-generated text on social media has created a need for NER to adapt to this noisy text genre. Performance of state-of-the-art NER on social media still lags behind well edited text genres. The authors of [75] presented the development and evaluation of a shared task on NER in Twitter, which was held

at the 2nd Workshop on Noisy User-generated Text (W-NUT 2016). This paper described the results of the Twitter NER tasks from 10-participant teams.

2.2.3 Entity Factor

Another important aspect is the types of entities that are being searched. At MUC-6, three types of named entities were defined which were collectively so called the "Enamex" as the annotation scheme. These three types of named entities are names of "person", "location", and "organization". At CONLL-2 and CONLL-3, there were four types; one more named entity type "MISC" was defined for those names which were not in the enamex. Four different named entities types: Person, Geographical and Political Entities (GPE), Organization and Facility were defined as tags-set for data at Automatic Content Extraction (ACE-2003) whereas in ACE-2004 and ACE-2005 another two more named types Vehicles and Weapons were added. However, further studies have defined and divided these types into further many detail subtypes. The most studied named types in literature are "Person", "Organization", "Location", "Number", "Date", and "Miscellaneous". A hierarchy of 150 NE types was proposed in [71]. Generally, the more categories are emphasized to classify, the harder it is. NER in biomedical domain focuses the named categories like "Protein", "DNA" and "Disease", etc.

2.2.4 Tagging Scheme

When considering NER as sequence learning problem, it is needed to assign a NE label to every token (word) in sequence. There are many NEs that consist of multiple words (e.g., University of Computer Studies, Yangon). For those chunks of NE, it is needed to locate the boundaries of NE. There are different chunk representation formats also called tag encoding scheme. The tags are usually distinguished by a single letter prefix, for example, B-PERSON, I-PERSON. The prefix letters have a meaning of relative position in the NE. The tagging scheme is named after the format of using the following prefix letters:

- B (Beginning) - represents the first token of the NE.
- I (Inside) - represents a part of the NE.
- L/E (Last, sometimes also End) - represents last token or end of the NE.
- M (Middle) – represents middle token of the NE.

- U/S (Unit, sometimes also Single token) - represents a single word.
- (Outside or Other)-represents token that is not part of the NE.

There are commonly used tagging format for tagging or labeling tokens in sequence leaning task in computational linguistics such as IO, BIO, IOB1, IOB2, and BIOES. Some researchers have been focusing on impact of those tagging format on data and made comparisons among these different tagging format.

2.3 Evaluation Matric

In any research area, it is essential to evaluate and compare results of proposed methodologies. NER is usually evaluated with standard evaluation measurement, which used three metrics called precision, recall, and F-measure (also F-score or F1 score) to describe the performance of NER system. Precision is a measure that indicates the fraction of the extracted entities that are correct, whereas recall is the fraction of the correct entities that are extracted. In other words, precision can be calculated by dividing the number of entities that a model predicted correctly by the number of entities that the model predicted. Recall can be calculated as the number of entities that a model predicted correctly divided by the number of entities that are identified by the human annotators. The F-measure is a harmonic mean between precision and recall and can be said that its aspect is to compute an overall accuracy.

$$Precision = \frac{Number\ of\ correctly\ extracted\ entities}{Number\ of\ extracted\ entities} \quad \text{Equation (2.1)}$$

$$Recall = \frac{Number\ of\ correctly\ extracted\ entities}{Number\ of\ all\ entities} \quad \text{Equation (2.2)}$$

$$F - measure = \frac{2*Precision*Recall}{Precision+Recall} \quad \text{Equation (2.3)}$$

2.4 Approaches to NER

Approaches to NER can be either linguistic based or statistical (machine learning) techniques. Another approach, the hybrid approach is the combination of the above two approaches. The computational research aiming at automatically identifying named entities in texts had been started since over past decades. Researchers tried to improve the performance of NER with vast techniques and thus it forms a massive and heterogeneous pool of methods, strategies and representations. One of the earliest research papers in the field was presented by Lisa F. Rau [66]. This

paper described a detailed explanation of implemented algorithm that relies on heuristics and handcrafted rules to extract and recognize names. NER had been the center of attention after MUC-6 and many computational researches had been carried out with different learning methods in which most initial works were proposed with linguistic approach such as dictionary look-up approach and rule-based approach but lately statistical machine learning sequence modeling approach was used and combination of these two approaches was also applied. Most recent studies make use of different deep neural architecture in such a way to eliminate feature engineering.

2.4.1 Dictionary lookup based NER

Dictionary lookup based approaches utilize a provided list of names (gazetteers) to identify the occurrences of names in text, usually by means of various substring matching techniques. The idea is that once a comprehensive list of names is constructed, names in a given text are looked up in the corresponding name lists. Basically, lexical resources such as name dictionary or gazetteers and WordNet are required.

Compared with other approaches, dictionary based approach is simple, fast and more accurate, and it can give very high precision. However, collection and preparation of name dictionary is very expensive and tedious. Furthermore, it is not easy to cover all names variants in name dictionary because more and more named entities are appearing constantly and it cannot resolve ambiguity. NER is not just matching strings with carefully constructed lists of names but only recognizes named entities which are being used as named entities in a given context. Lately, dictionary based technique is not used separately, but is often used as part in other approaches. A drawback of this approach is the need of constructing and maintaining the resources.

2.4.2 Rule-based NER

In this approach, a set of rules is manually crafted by linguists and experts based on syntactic, linguistic and domain knowledge to recognize a particular NE type. These rules are then implemented and the output is given by matching the rules. While lexical and syntactic cues (e.g., capitalization and presence of prefix title, suffix, and special characters) are often used to create rules, these rules are not sufficient to identify all occurrence of names; there are many special cases and most

rules are exception. It is clear that rule-based approach depends on a set of rules and thus it requires experts with huge experience and grammatical knowledge to define rules.

Due to the lack of annotated corpus resources for Malay language that can be used as a training data, a rule-based approach that works through three-step process was proposed by the authors of [1]. As the first step, tokenization process splits the sentences into tokens. The second step is POS tagging process. The final processing step, deciding a tokenized word is whether or not a one of the three types of NE (person, location, and organization) was basically based on the rule-based POS tagging process and contextual rules. Their proposed Malay NER obtained a reasonable F-score value of 89.47%.

2.4.3 Statistical-based NER

Recent computational research trend on NER is moving from rule-based to statistical (machine learning) approaches. The statistical approach is quite different from rule based approach where detecting and classifying the named entity solely rely on the rules given by the linguists; it uses mathematical formulas and logic to find and classify named entity. Statistical models are automatically constructed from linguistically annotated resource. In statistical approach, a corpus is initially learnt where training module is run to identify named entity and then based on the occurrence of named entities in the corpus, a probability is calculated. When a text is given at any time, the result is provided, based on the probability value.

Statistical approach differs from early approaches in that instead of using handcrafted rules, it makes the system learn to identify named entities through their distinctive features and some mathematical implementations during training process. There are three main learning methodologies to learn the system: supervised learning, semi supervised learning and unsupervised learning. When working with statistical approach, feature selection becomes critical process and development of annotated corpus must be accounted. Choice of features affects the performance of NER and also errors in annotated training corpus affects badly to machine learning based models. The availability of large and sufficient annotated datasets becomes the major drawback when working through with a statistical way.

In developing statistical NER, there are different models ranging from Decision Tree to Conditional Random Fields (CRF) and all these models have their own mathematical approaches for training and determining the probabilistic values. Moreover, each model has own methodologies of working to get the desired result. Among most popular models, Hidden Markov Model (HMM), Maximum Entropy, support vector machine (SVM), and Conditional Random Fields (CRFs) are most common. Combination of different machine learning approaches is also used. In comparison among the statistical machine learning models, there is no particular winner found. Each model has its own advantages and disadvantages.

Florian et al. [24] presented the best system at the NER CoNLL 2003 challenge, with 88.76%F1 score. They used a combination of various machine-learning classifiers. Features they picked included words, POS tags, CHUNK tags, prefixes and suffixes, a large gazetteer (not provided by the challenge), as well as the output of two other NER classifiers trained on richer data sets. The second best performer of CoNLL 2003 (88.31% F1) [11], also used an external gazetteer (their performance goes down to 86.84% with no gazetteer) and several hand-chosen features.

2.4.3.1 Maximum Entropy Model

Maximum Entropy is a conditional probabilistic sequence model. This model is very flexible and it can handle the dependency between multiple features that are extracted from a single word. In Maximum Entropy model, every state holds an exponential model which takes the observation features as input and produces the conditional probability of the possible next state. Maximum Entropy model suffers from the label bias problem. Moreover, the future observations are not taken into account.

A named recognition system was built within the framework of Maximum Entropy in [9]. The system used knowledge sources in making its name tagging decision. These knowledge sources considered in their system were lexical features, capitalization features, and features indicating the current section of text (i.e. headline or main body). Besides, dictionaries of single or multi-word terms were also considered.

A named entity recognizer was developed by employing Maximum Entropy Markov Model and measured the performance on the Czech Named Entity Corpus and also on the English CoNLL-2003 shared task [74]. Maximum entropy model predicts the probability distribution and position with respect to an entity for each word in a sentence. In decoding, Viterbi algorithm decodes an optimal sequence labeling along with the probabilities estimated by the maximum entropy model. Morphological features, word information, and gazetteers were utilized as features. They achieved satisfactory results on both Czech and English.

2.4.3.2 Support Vector Machine (SVM)

Support Vector Machine is used for solving the two-class classification problems which means to identify whether an entity belongs to the target class or not. A hyperplane, which is generated during training, helps to categorize the observed data into targeted positive and negative class. This model computes margin value which is the distance of every vector from the hyperplane. Since SVM supports Kernel functions, it can learn various combinations of the given features with less computational complexity. But the dependencies such as state to state and the feature to feature dependencies are not considered by SVM.

As the continuation of [74] work, authors of [40] had prepared their own manually annotated Czech NER corpus in there a rich two-level annotation scheme was used and released for publicity. They tried to develop NER by utilizing SVM technique on this corpus. The authors did not use as many features as [74], but carefully selected features that were considered to be helpful such as using only part of speech, gender, case and number instead of using the entire Czech morphological features. Their classifier was capable of classifying NE into 62 categories and outperformed the previous state-of-the-art Czech NER by [74].

Asif et al. [22] reported SVM based NER system that was able to recognize four different NE classes, such as Person, Location, Organization and Miscellaneous name by working with language independent features that are helpful in predicting NEs. Although the authors developed NER system for Bengali and Hindi languages, they only considered language independent features with the intension of applying these features to NER in any language. Lexical patterns had also been generated from unlabeled news corpus through an unsupervised technique and used as the features of

SVM. As a result, the F-score improved 5.13% compared to without using those lexical patterns features.

The most challenging aspect of any machine learning approach is deciding on choosing the optimal features. The authors of [6] investigated the impact of using different language independent and also language specific features in a discriminative SVM machine learning framework for Arabic NER. The authors systematically investigated a large space of features and the impact of these different features in isolation and also combined. The best performance was achieved when they used all the features jointly and their system yielded F-score 82.71%.

A supervised machine learning based approach in which Fuzzy membership function was added and called Fuzzy Support Vector Machine (FSVM) for NER was presented by [50]. The authors applied fuzzy algorithm to improve classification in SVM method. By doing this way, it was possible to classify multi classes in SVM which means that it can resolve the SVM weakness in multi classification. Feature sets were selected based on lexical information, affix, NE information of previous word, possible NE class and token feature. They showed that Fuzzy membership function help recognizing named entities more semantically than existing method.

2.4.3.3 Hidden Markov Model (HMM)

Hidden Markov Model has been proved that it is such a very successful model in various sequence labeling tasks. It is a type of generative model and based on the Markov chain and each label is represented as states. HMM model defines the joint probability for each observation symbol and the state transition therefore there is a probability associated with each transition. Hence, this model can predict the NE of the next word when the NE class of the previous word is given. The future observations are taken into account. In HMM model no dependency among the words in the input sentence is considered. Due to the joint probability definition, a lot of parameters have to be evaluated. As a consequence, it requires a large dataset for training.

The researchers of [94] resolved the NER problem through HMM modeling along with the use of internal and external features. In order to resolve the data sparseness, two-level back-off modeling was applied. The influence of various internal and external features on NER performance was reported. Systematic

experiments were executed on MUC-6 and MUC-7 English NE tasks. It gained F-score of 96.6% and 94.1% respectively on those data. The authors also tried to examine the impact of training data size on performance.

The work of [20] was also the development of a statistical HMM based NER system. It was initially developed for Bengali using a POS tagged Bengali news corpus. Evaluation results of the 10-fold cross validation test revealed average F-score of 84.5%. This HMM based NER was then trained and tested on the Hindi data to show its effectiveness towards the language independent ability. During training, to avoid data sparse problem, the linear interpolation method had been applied. Additional context dependent features were also introduced to emission probability. Instead of using the emission probability directly, the smoothing technique was applied. Suffix features of the words and a prepared lexicon which has around 100,000 word entities were also used in Viterbi decoding to handle the unknown words in Bengali language.

In paper [57] described the HMM based language independent NER which can be applied to any domain or language. Although their HMM based NER system had been trained and tested with different Indian languages namely Hindi, Urdu, and Punjabi, etc. that methodology worked well on any natural language. The difference between their HMM based NER and other is that all the parameters used were dynamic which means that it can be used in accordance with the entity interest. Moreover, this dynamic nature of parameters can help the NER system to be used for other NLP classification tasks such as POS tagging, etc. Fine grained tagging was also allowed.

A statistical HMM based NER system for English and seven Indian languages which is of language independent nature was submitted for the ICON 2013 NLP tools contest [25]. It was designed to recognize various NE types like artifact, entertainment, facilities, location, locomotive, materials, organization, organisms, person, plants, count, distance, money, quality, date, day, period, time, and year. It did not use any gazetteers for the task because of the unavailability of gazetteers for all seven Indian languages. Information about word, POS, and also chunk tag were used during training and considered trigram. To handle unknown words, the researchers estimated the observation probability of an unknown word by analyzing POS information, chunk information and the suffix of the word associated with the

corresponding the trigram. All the experiments were performed on the ICON 2013 datasets.

2.4.3.4 Conditional Random Fields (CRF)

Conditional Random Field (CRF), a discriminative probability model, is similar to the Maximum Entropy Model but the main difference is that CRF overcomes the label bias problem. CRF is a kind of undirected graphical model and it identifies the relationship between the observed data and it generates structured predictions based on these observed data [39] [44] [76]. Unlike other discrete classifiers which predict a label for entities only without considering the context or the neighboring entities, a CRF model will take the context into account. Although CRF model is very adaptable and flexible with respect to feature selection, the computation time is very long.

Another submission for ICON-2013 was a statistical CRF based NER system which used different combination of language independent features such as context words, prefix and suffix information, POS and chunk Information, first and last words of a sentence, binary feature for digit, token id, associated verb, gazetteers, and capitalization for English [17]. They used gazetteer feature in every language except for Tamil and Telugu. F-score value of Tamil and Telugu were not pretty good as other languages which used gazetteers features during training. This means that among those features, gazetteers are very helpful to boost the performance of NER.

The effort [62] presented a CRF-based supervised approach towards NER task in clinical text and experiments were carried out on i2b2 shared task 2010 data, to recognize three types of NE (Problem, Treatment, and Test). With the help of unique feature set specifically customized for clinical NER, their approach worked much better than all supervised and hybrid models on the same data and gave almost as similar as semi-supervised models used in the shared task. This showed that feature selection is an influential factor in supervised learning.

The problem of NER in Manipuri language, a highly agglutinative Indian language, was also dealt with using CRF [60]. Experiments were carried out with different combination of features in order to identify best features which can give the maximum result. Likewise, the development for NER for Bengali language was reported using CRF in [23]. It deployed the contextual words along with the various

kinds of features that are beneficial in predicting the four major NE categories (person, location, organization, and miscellaneous name). The experimental results that come from the 10-fold cross validation test showed that their proposed CRF based NER was effective and achieved an overall average F-score value of 90.7% and average precision and recall values of 87.8% and 93.8%, respectively.

A comparative study between six ensemble learning approaches and six traditional classification approaches was made on two Arabic NER datasets (ANERcorp and AQMAR) [5]. Among six ensemble learning approaches (Ada Boost, Bagging, Ensembles of Nested Dichotomies, Multi Boost, Random Forest and Rotation Forest), the Random Forest achieved the highest F-score values on both datasets.

The performance of Naïve Bayes, SVM, and Simple Logistic algorithms on Indonesian NER for newspaper articles were compared to find out which machine learning algorithm gives the best accuracy in classifying into 15 types of NE [83]. Word-level features, sentence-level features, contextual feature plus list-lookup features were employed in combination through the experiments to determine the significance of the various features. The highest F-score was generated by the algorithm Simple Logistic along with the features combination of word-level, sentence-level and the list-lookup features.

2.4.4 Deep Learning Approach to NER

In dictionary lookup approach, the performance totally depends upon the coverage of the dictionary. Likewise, it is required to have linguistic knowledge to set rules for rule-based approach. On the other hand, statistical ways to NER are more robust than rule based methods. However, those statistical methods require handcrafted features and a large set of linguistic knowledge to identify NE effectively. Additionally, those approaches are highly dependent on the choice of features. Recently, deep learning has been extensively applied to sequence tagging in many languages, and there has been a shift of focus from feature engineering to designing and implementing effective deep neural network architectures.

Neural network models are capable of diminishing the burden of statistical models which is the need to work with prepared features, as neural networks' deep layers can learn task-relevant features automatically. Collobert et al. [16] developed a

general neural network architecture for sequence labelling tasks. Following this architecture, recurrent neural networks had also been designed to be used in sequential data modeling and had been proved that these networks are quite efficient in such sequential tagging task as well. Likewise, Long Shorten Memory (LSTM) neural network, a special kind of RNN, and bidirectional LSTM neural network where two LSTM networks work together, have also been widely applied in such sequential tagging tasks. Bidirectional LSTM network has an ability of retaining information about the sequence for long periods and also gathering it from both directions, thus making magnificent improvement in linguistic computation.

Huang et al. [35] integrated a bidirectional LSTM network and a statistical Conditional Random Field (CRF) to provide the past and the upcoming input feature, as well as sentence level tag information. Such kinds of network architectures become the primary choice in recent research and provide significant improvement than statistical algorithms.

A comprehensive survey was made by the authors of [86] on advances in NER form deep learning models. The authors also contrasted deep neural network architectures with previous approaches to NER, and highlighted the improvements achieved by neural models as well.

A deep learning architecture for Italian sequence labeling task that exploits both word-level and character-level representations through the combination of bidirectional LSTM, CNN, and CRF was proposed in [4]. In their work, word embeddings are built by exploiting the Italian Wikipedia. Word2vec was used for creating embeddings with a dimension of 300, and all words that have less than 40 occurrences in Wikipedia were removed. Their system was able to outperform the first three systems which all adopt statistical classification methods.

An implementation of Indonesian NER was managed to extract four classes of NE (Person, Organization, Location, and Event) by providing comprehensive comparison among all experiments with various deep learning approaches [28]. ‘BILOU’ scheme was used in label tagging. Dense layer with softmax was used as the activation. Through the whole conducted experiments, hybrid bidirectional LSTM and convolution neural network (CNN) architecture provide the highest score among all other architectures. CNN layer helps to extract morphological information from a given sentence and applied after the dropout layer. 4-fold cross validation was used to

measure the performance. Despite small size of dataset, their results reported that deep learning can achieve good performance without any feature engineering process.

The paper [8] introduced a deep neural network model to extract four types of NEs (Person, Location, Organization, and Geo-political) from Italian text. The authors proposed recurrent context window network architecture in which a sliding window of word contexts is used to predict. To avoid overfitting, early stopping [64], weight decay [41], and Dropout [73] were applied. The IOB tag encoding scheme was used in data annotation process. Through the training the Negative Log Likelihood was used as cost function, Stochastic Gradient descent (SGD) was applied to learn the parameters of the network. Furthermore, to compute the updates, the back propagation was utilized. At testing time, the tag associated with the highest class conditional probability was selected. 10-fold cross-validation evaluated on the Evalita 2009 benchmark dataset proved that their neural model was comparable with the state-of-the-art model.

Two neural architectures for NER as sequence labeling that use no language-specific resource was presented in [45]. One of the models was designed based on bidirectional LSTM that models output label dependencies via a simple CRF layer, and the other was constructed using a transition-based algorithm. The IOBES tag encoding scheme was used. Their models gained state-of-the-art performance on the CoNLL-2002 and CoNLL-2003 datasets which contain four languages (English, Spanish, German, and Dutch).

The success of applying a neural architecture for sequence labeling to Vietnamese NER was demonstrated in [58]. Their model relied on no task-specific resources, handcrafted features, or data pre-processing except two information: 1) character-based word representations learned from the annotated corpus and 2) pre-trained word embeddings on unannotated data. Firstly, this model used bidirectional LSTM (Long-Short Term Memory) units to learn character representations. Next, character representation was concatenated with pre-trained word embeddings and then applying dropout to encourage the model to avoid overfitting, then feed them into a bidirectional LSTM to capture context information of each word. They optimized parameters using (Adaptive Moment Estimation) Adam optimizer. On top of bidirectional LSTM, sequential CRF layer was added to decode labels for the whole sentence. Experiments on benchmark datasets showed that their work obtained state-of-the-art performance of 94.88% F-score.

The authors in [2] presented the bidirectional neural architectures which were built on LSTM and GRU for Arabic NER. In their networks, after obtaining the word and character embeddings, an embedding attention layer was adopted to combine the two features with the intention of getting the best word representation. Then, the embedded feature representation was fed into the encoding layer for processing. A SoftMax layer was applied to normalize the output. AdaGrad was used to optimize the network cost. Experiments were conducted on ANERcorp dataset and network architecture with LSTM gave better results than with GRU.

Recently, state-of-the-art performance on the CoNLL 2003 NER dataset has been achieved by employing multiple smaller independent bidirectional LSTM units rather than using a single LSTM component [27]. This new architecture has the benefit of reducing the total number of parameters.

In [12], the authors proved that their neural network model, which is an incorporation of a bidirectional LSTM and a character-level CNN, achieves state-of-the-art results in NER with little feature engineering. The benefits of robust training through dropout were also shown in their experiments.

A truly end-to-end NER system was introduced by [49]. The authors introduced a neural network architecture that benefits from both word-level and character-level representations automatically, by making use of a combination of bidirectional LSTM, CNN and CRF. Their proposed model required no task specific resource, no feature engineering or data pre-processing, thus making it applicable to a wide range of sequence labeling tasks and gained 91.21% F1 on CoNLL 2003 corpus for NER.

A Vietnamese Named Entity Recognition was proposed by incorporating automatic syntactic features with word embeddings as input for bidirectional LSTM in [63]. The authors showed that the effectiveness of automatic syntactic features and their proposed method achieved an overall F-score of 92.05%.

A model was implemented with a CNN over the characters of word in [70]. In their modeling, the word embeddings of the central word was concatenated with its neighbors, and then these were fed to a feed forward network, and followed by the Viterbi algorithm to predict labels for each word. The model achieved 82.21% F score on Spanish CoNLL 2002 data and 71.23% F score on Portuguese NER data.

In [18], the authors implemented a model by concatenating word embeddings with bidirectional LSTMs over the characters of a word, passing this representation

through another sentence-level bidirectional LSTM, and predicting the final tags using a final softmax layer in the NeuroNER toolkit with the main goal of providing easy usability and allowing easy plotting of real time performance and learning statistics of the model. The BRAT annotation tool is also integrated with NeuroNER to ease the development of NN NER models in new domains. NeuroNER achieved 90.50% F score on the English CoNLL 2003 data.

Deep learning has been shown that it performs domain independently so that many researchers tried to adapt deep learning methods form domain to domain. An attempt was made to analyze to what extent transfer learning with bidirectional LSTM-CRF on a noisy source sliver-standard corpus to a more reliable target gold-standard corpus reduces error rate and improves performance. Their study covered four different biomedical NE classes: chemicals, disease, species, and genes/proteins [26].

A deep neural network was developed to generate word embeddings form a large unlabeled corpus through unsupervised learning which are further fed into another deep neural network for the Chinese clinical NER in the paper [84]. Their experiment results showed that the deep neural network model with word embedding from large unlabeled corpus outperforms the state-of-the-art CRF model.

Similarly, a study that examines two popular deep learning architectures, CNN and RNN, was done to extract important concepts (clinical NE) from clinical texts [85]. The performance of their neural models was compared with baseline CRF model and two state-of-the-art clinical NER systems using the i2b2 2010 clinical concept extraction corpus. Their evaluation results showed that the RNN model trained with the word embeddings achieved a new state-of-the-art performance of 85.94% for the defined clinical NER task.

Bidirectional LSTM with inference CRF neural models was described in [92] to address chemical and disease NER tasks. The authors compared the use of LSTM-based and CNN-based character-level word embeddings in their modeling. Evaluation results on BioCreative V CDR Corpus showed that the use of both types of character-level word embeddings with bidirectional LSTM-CRF models were approaching to comparable state-of-the-art performance.

A NER system focusing on multi-task and multi-lingual joint learning was proposed by Yang et al. [89]. The network takes inputs which are word representation given by GRU (Gated Recurrent Unit) cell over characters plus word embeddings.

These embeddings were passed through another RNN layer and the output was given to CRF models trained for different tasks like POS, chunking and NER. Furthermore, they also proposed transfer learning for multi-task and multi-learning, and showed slight improvements on CoNLL 2002 and 2003 NER data, achieving 85.77%, 85.19%, 91.26% F scores on Spanish, Dutch and English, respectively.

Transfer learning for NER with neural networks was studied to address label scarcity issue [47]. The authors had studied transfer learning with ANNs for NER, specifically patient note de-identification, by transferring ANN parameters trained on a large labeled dataset to another dataset with limited human annotation. The authors also demonstrated that transfer learning improves the performance over the state-of-the-art results and may be especially beneficial for a target dataset with small number of labels.

2.4.5 Hybrid NER

Apart from those traditional approaches mentioned above, another approach is the hybrid approach, the integration of linguistic and statistical machine learning approaches, which leverages the advantages from these two approaches. Hence, it benefits from both approaches; this hybrid NER enables new methodologies by using the strongest points from each approach.

A hybrid approach which is composed of maximum entropy model, language specific rules and gazetteers was described by [69] to tackle the task of NER in Indian languages. The intention of which is to extract 12 types of NEs: person, title-Person, designation, organization, brand, abbreviation, title-object, location, time, number, measure, and term. In developing their maximum entropy model, orthography features, suffix and prefix information, morphology information, part-of-speech information plus information about surrounding words and their corresponding tags were considered as features. The authors also defined 36 rules in total for detecting time, measure and number types and also developed a module for semi automatically extraction of context patterns. The evaluation was conducted on 5 Indian languages: Hindi, Bengali, Oriya, Telugu and Urdu and Hindi led with better accuracy.

In the paper [72], the first hybrid approach in NER for Arabic that is capable of recognizing 11 types of Arabic named entities: Person, Location, Organization, Measurement, Date, Time, Percent, Price, Phone Number, ISBN and Filename, was

proposed. Decision tree, SVM and logistic regression classifiers were applied as machine learning techniques. This NER works through the two stages pipeline which is solved sequentially so that the output of the first rule-based stage is processed, afterwards fed it as input into the next machine learning stage in the sequence. Their proposed hybrid NER outperforms not only the rule-based but also machine learning based approach and even the state-of-the-art of Arabic NER.

Likewise, [56] attempted NER on Persian language, a rarely studied language; plus proposed a hybrid NER system that recognizes names of people, locations, and organizations. Their machine learning constituent used HMM and Viterbi algorithm, and a set of lexical resources and pattern bases were employed in its rule-based section. Although their experimental results showed that their approach had achieved satisfactory performance, lack of training data is still an obstacle and their lexical resources and rule patterns did not cover to solve all the language ambiguity in identifying names.

Another study adopting hybrid approach was developed by [3] in order to recognize names of Person, Location, Title-person, Date, Time, Organization, Event, Facility, Designation, Abbreviation, Artifact, Relation, Number, and Measure for Punjabi language. In their study, two versions of NER were presented; the first version was designed with HMM only whereas the second version had combined HMM with handcrafted rules. Their experimental result approved that the second version which is a hybrid approach gives better accuracy, nearly 26% was increased that results obtained from hybrid approach were pretty notable.

In the same way, NER task for Malayalam language, one of the understudied India languages, had been explored by applying a combination of statistical machine learning and rule-based approach [37]. A hierarchy of named entity which was composed of three major classes: Entity Name, Time, and Numerical expression as first level tags was defined for tagging. Furthermore, under Entity Name tag, there were 11 tags with 46 sub-tags; another 20 tags under those sub-tags. Time had 7 tags whereas under Numerical expression there were 7 tags, too. As their experiments, firstly, a comparison was made between two supervised methods namely TnT (a statistical Trigram n Tags) and SVM under the same domain and found that both methods performed well but did not give satisfactory result for embedded tags. However, their proposed hybrid supervised machine learning approach which is a

combination of TnT and rule-based approach showed better results for embedded tags and is suitable for Malayalam NER task.

Following the interest taken into NER in academic literature, [48] tried to detect mentions of chemical compounds and drugs in text by combining the statistical machine learning method CRF with the result from the dictionary lookup method. Experiments were conducted with various training strategies and features combination.

The effectiveness of a lattice-structured LSTM model was investigated for Chinese NER in the paper [93]. Their model encodes not only a sequence of input characters but also all potential words that match a lexicon. The authors showed that their model explicitly leverages word and word sequence information when compared with the character-based methods. Besides, when compared with word-based methods, their lattice LSTM did not suffer from segmentation errors.

2.5 Some Previous Research on Myanmar NER

There are only two previous attempts on Myanmar NER. Previous attempts on the task of NER for Myanmar language had been done by utilizing traditional approaches.

A hybrid method had been presented by the work [77] for Myanmar Named Entity Identification. This method is a combination of ruled based and statistical N-grams based method which use name database. They classified Myanmar NEs into three classes, namely person name (PER), organization name (ORG) and location name (LOC). They had examined a sample of 43 Myanmar text files. Their experiments gave 82.75% precision and 83.40% recall on the sample data.

Moreover, another effort on Myanmar NER proposed Myanmar Named Identification algorithm. In the algorithm, the system defines the names by using the POS information, Name entity identification rules and clue words in the left and/or the right contexts of NEs carry information for NE identification [79]. Therefore, input sentence must be specified with designated POS tags and the performance totally depends on the linguistic rules. Moreover, there is a weakness which is the ambiguity of sematic inference on proper names. Moreover, linguistic knowledge and external features are necessary for these approaches.

CRF-based approach had also been applied Myanmar NER as part of this

research. The influence of internal and external features had also been explored with statistical CRF approach for Myanmar NER.

2.6 Summary

This chapter is about the description of Named Entity Recognition, its related factors, approaches and evaluation factors. Moreover, related research and experimental results of each approach were also presented.

Along the work of NER, it is needed to consider factors such as language, domain, and entity types. Tagging scheme is another factor to take into account when NER is learned through sequence learning approach.

Among three main approaches, linguistics-based approach requires large amount of linguistic knowledge whereas traditional statistical machine learning approach works well with the help of effective features. By taking the advantages of linguistics approach and machine learning approach, the hybrid approach is also applied in many NER researches. In recent years, deep neural networks have got attention due to the ability of reduction of feature engineering and deep learning approach becomes the choice of many researchers.

Until now, no research has been done on NER for Myanmar language by using deep learning approach. The previous works for Myanmar NER were done by hybrid approach and statistical approach.

CHAPTER 3
DEEP LEARNING METHODOLOGIES

This chapter presents the concepts and nature of deep learning methodologies and widely used different neural networks in NLP, and recent trends of deep learning in NLP are described in detail.

3.1 Recent Trends in Deep Learning Based Natural Language Processing

Deep learning is a machine learning technique where learning the training parameters and feature extraction from input data are automatically performed with the help of deep neural networks without human intervention. The term “deep learning” turns out for “stacked neural networks”; that is, networks composed of several layers.

Deep learning algorithms and deep neural network architectures have already made magnificent advances in fields of computer vision and even pattern recognition. Deep learning methods make use of multiple processing layers to learn hierarchical representations of data; and have gained popularity by producing state-of-the-art results in several domains. Following this trend, recent NLP and speech research are now increasingly focusing on the use of new deep learning methods.

For decades, traditional statistical machine learning approaches especially targeting NLP problems have been stood upon shallow models trained on very high dimensional and sparse features. Over the past few years, deep neural network architectures based on dense vector representations have been providing better results on various kinds of NLP and speech processing research. This trend has been come out with the success of word embeddings and the ability of deep learning which enables automatic multi-level feature representation learning. In contrast, traditional machine learning based NLP systems work heavily together with hand-crafted features.

Collobert et al. [16] proposed a simple deep learning framework and demonstrated that it provides superior results in various NLP tasks. Since then, many complex deep learning based frameworks have been proposed to solve several problems in NLP tasks, ranging from POS tagging to machine translation. Young et

al. also made a survey on significant deep learning related models and methods which have been employed for numerous NLP tasks [90].

Despite deep neural sequence models being dominant in the recent research literature, the comparison between different deep neural models is challenging due to sensitivity on experimental settings. Yang et al. investigated the design challenges of constructing efficient and effective neural sequence models, and conducted a systematic model comparison on three benchmarks sequence labeling tasks (NER, POS tagging, and Chunking) [88].

A toolkit for neural sequence labeling, NCRF++, was designed for implementation of different neural sequence labeling models [87]. It is an open-source and provides users to design the custom neural models.

3.2 Deep Neural Networks

Generally speaking, a deep neural network is a neural network with a certain level of complexity, a neural network with more than two layers. Earlier versions of neural networks such as the first perceptrons were shallow, composed of one input and one output layer, and at most one hidden layer in between. More than three layers (including input and output) certifies as “deep” learning. Deep neural networks apply sophisticated mathematical modeling to process data in complex ways.

3.2.1 Neural Network Tuning

Deep neural networks can be difficult to tune. There are many aspects that can help to optimize the network. If the network hyperparameters are poorly chosen, the network may learn slowly and the result may not be satisfied, or perhaps not at all.

3.2.1.1 Parameters and Hyperparameters

The core difference between parameters and hyperparameters is that parameters are learned by the model during the training time, while hyperparameters can be changed before training the model. Parameters of a deep neural network are Weight and Bias, which the model updates during the backpropagation step. On the other hand, there are a lot of hyperparameters for a deep neural network, including:

- Data normalization
- Weight initialization

- Learning rate – α
- Number of iterations (Epoch)
- Number of hidden layers
- Units in each hidden layer
- Activation function
- Loss function
- Regularization (e.g. Dropout, Early stopping and weight decay)
- Minibatch size
- Choice of optimization algorithm, and so on.

3.3 Distributed Representation

For decades, statistical way to NLP has been the primary choice for modeling complex NLP tasks. However, this statistical NLP often suffers from the noted the curse of dimensionality problem while learning joint probability functions in modeling. This leads to the motive of learning distributed representation of words in low-dimensional space which main idea is to represent words as feature vectors. The major benefit of the dense representation is generalization power which can provide a representation that is able to capture similarities. A distributed representation is possibly one of the key breakthroughs for the impressive achievement in performance of deep learning methods on challenging NLP problems.

3.3.1 Word Embedding

Word embedding is a kind of word representation where individual words are represented as real-valued vectors in a predefined vector space. The distributed representation is learned through the usage of words so that words that are used in similar ways or words that have similar meaning can have similar representation. Word embeddings are often utilized as the first data preprocessing layer in a deep learning model. Word embeddings are typically trained by optimizing auxiliary objectives in a large unlabeled data; mainly learned through context where the learned word vectors can capture general syntactical and sematic information. Thus, these embeddings have revealed to be efficient in automatic capturing of features from text. Generally speaking, word embeddings provide the main support for the deep learning models to result in having state-of-the-art performance.

In literature, several neural language models that learned distributed representations for words have been developed. The work that shows the utility of pre-trained word embeddings has also been established. After the CBOW and skip-gram models were being proposed by Milolov et al. [52], word embeddings became revolutionized. CBOW computes the conditional probability of a target word given the context words surrounding it within a window size. In the case of the skip-gram model, the surrounding context words given the central target word are predicted (see Figure 3.1). The context words are assumed to be located systematically to the target words within a distance equal to the window size in both directions.

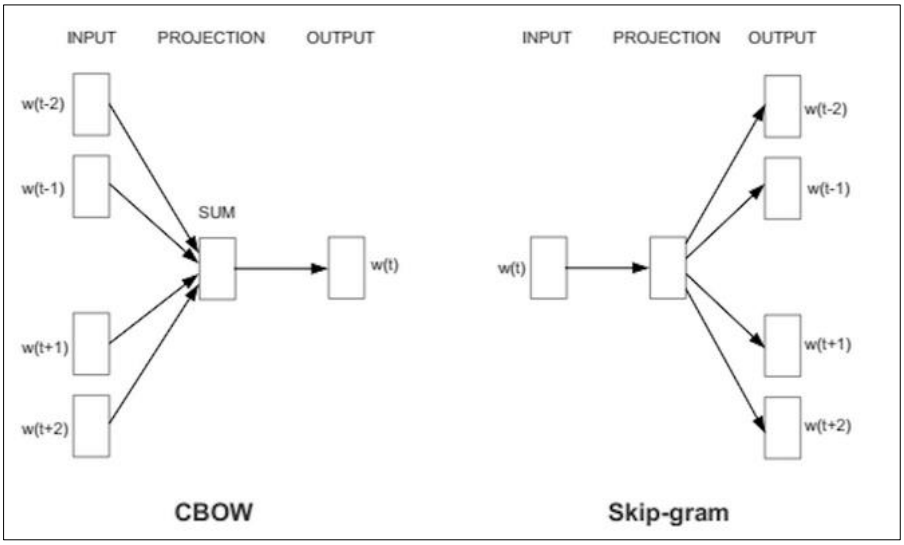


Figure 3.1 Difference between CBOW and Skip-gram (Figure Source: [38])

3.3.2 Character Embedding

Word embeddings are able to capture syntactic and sematic information. Besides these, intra-word morphological and shape information can also be very helpful as well as useful. The fact that character-level representations along with word embeddings offer better performance on morphologically rich languages has been reported in certain NLP tasks.

Furthermore, the unknown word issue or out-of-vocabulary word (OOV) issue is a common occurrence for languages with large vocabularies and with rich morphologies. This issue can be naturally dealt with by character embeddings for the reason that each word is considered as no more than a composition of individual characters. In languages where text is not composed of separated words but individual

characters, and the semantic meaning of words map to its compositional characters, processing on the character level may probably be a natural solution to avoid word segmentation. Hence, language processing employed deep learning methodologies on such languages tends to prefer character embeddings over word vectors.

In processing morphologically-rich languages, character-level information has also being used by many researchers in attempting to improve the representation of words. The skip-gram method by representing words as bag-of-characters n-grams is also applied. This kind of work had the effectiveness of the skip-gram model along with addressing some issues of word embeddings.

3.4 Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNNs) are deep neural networks that are primarily used in image processing. The efficiency of convolutional nets in image proceeding is one of the main reasons why the world has woken up to the efficiency of deep learning. Figure 3.2 gives the illustration of CNN. A basic convolutional network consists of three basic components:

- The convolutional layer
- The pooling layer
- The output layer

The convolutional layer is the core building block of a CNN. The convoluted output is obtained as an activated map. A set of learnable filters (kernels) is convolved across the width and height of the input volume to extract relevant features from the input and to pass further layer. Another important concept of CNNs is the pooling layer which serves to reduce the number of parameters while learning features. The output layer is a fully connected layer where a loss function is defined to compute the mean square loss. The gradient of error is then calculated and the error is then back propagated to update the weights and bias values.

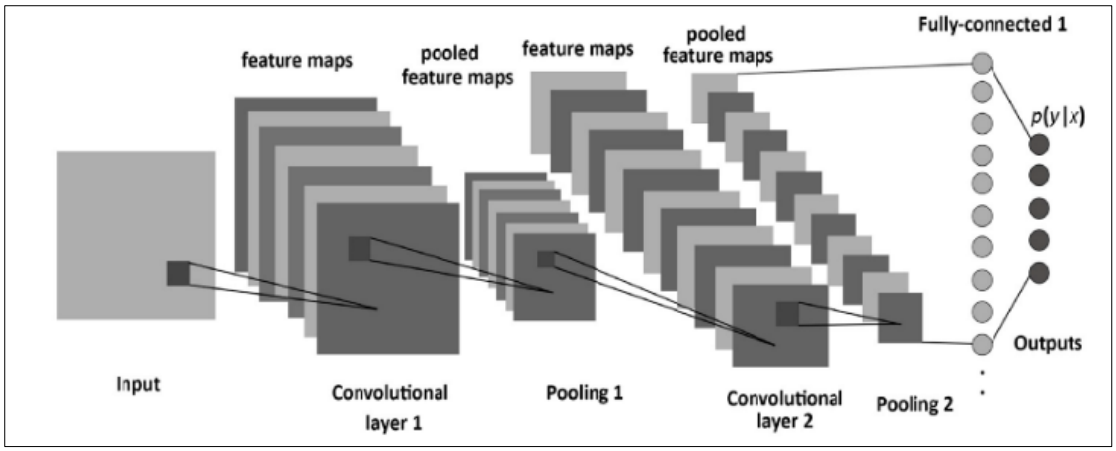


Figure 3.2 Convolutional Neural Network (Figure Source: [91])

Even though convolutional nets had emerged to be the natural choice given their effectiveness in image processing and advanced computer vision tasks, they have also been investigated for modeling character-level information, among NLP tasks. Following the popularization of word embeddings and its ability to represent words in distributed space, the necessity emerged for an effective feature function that extracts higher-level features from constituting words or n-grams. The use of CNN for sentence modeling was firstly pioneered by Collobert et al., [16]. A look-up table was utilized to transform each word into a vector of user-defined dimensions. CNNs are able to extract salient n-gram features from the input sentence to create an informative latent sematic representation of the sentence.

In a CNN, a number of convolutional filters (kernels) of different widths slide over the entire word embedding matrix. Each filter extracts a specific pattern of n-gram. The combination of convolutional layer followed by pooling is often stacked to generate deep CNN networks. These sequential convolutions help in improved mining of the sentence to seize a truly abstract representations comprising rich sematic information. The filters through deeper convolutions cover a large part of the sentence until finally covering it fully and creating a global summarization of the sentence features. To adapt CNNs for those NLP tasks such as NER, and POS tagging which requires word-based prediction, a window approach is used. For each word, a fixed-size window surrounding itself is assumed and CNN is applied.

Overall, CNNs are extremely beneficial in mining sematic clues in contextual windows but they are very heavy data model. Lately, CNN has been widely applied in

NLP research tasks after being found its ability of outperforming traditional models such as bag of words and n-grams, and so on.

3.5 Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN) allows us to operate over sequential information. The term recurrent means the output of the current time step becomes the input to the next time step. RNN does the same operation repeatedly on each instance of data stream such that the output is dependent on the previous computations and results. This means that RNNs have memory over previous computations and use this information in current processing. This nature is actually suited for many NLP tasks such as language modeling, machine translation, speech recognition, image caption. This made RNNs increasingly popular for NLP applications in recent years.

There are a few properties why RNNs gain popularity in numerous NLP tasks. Firstly, given that RNN operates sequential processing by modeling instances in sequence, it has the ability to capture the inherent sequential nature present in NLP, where instances are characters, words, phrases or even sentences. Semantic meaning between words can be developed based on the previous words in sentence. Thus, in modeling such context dependencies in language and sequence modeling tasks, RNN turns out the natural choice of researchers over other deep neural nets.

Another fact aiding RNN's suitability for sequence modeling tasks lies in its ability to model variable length of text, including vary long sentences, paragraphs and even documents. Unlike CNNs, RNNs have flexible computational steps that provide better modeling capability and create the possibility to capture unbounded context. This ability to handle input of arbitrary length becomes one of the selling points of major works using RNNs.

Moreover, many NLP tasks require sematic modeling over the whole sentence. This involves creating a gist of the sentence in a fixed dimensional hyperspace. In such a case where the whole sentence is needed to summarize to a fixed vector and then mapped back to the variable length target sequence, RNN's ability to summarize sentences led to their increased usage. Furthermore, RNN provides the network support to perform time distributed joint processing which is the main aid of most sequence labeling tasks like POS tagging. The above points are some of the focal reasons that motivated researchers to opt for RNNs.

RNNs have loops in them which mean that they have memory, allowing information to persist. A loop allows information to be passed from one step of the network to the next. Figure 3.3 illustrates a general RNN which is unfolded across time to accommodate a whole sequence. In the figure, x_t is taken as the input to the network at time step t and in the context of NLP, x_t typically comprises of embeddings. O_t illustrates the output of the network. U, V, and W account for weights that are shared across time and h_t represents the hidden state at the same time step. Thus, h_t is calculated based on the current input and the precious time step's hidden state. This chain-like nature reveals that RNNs are intimately related to sequence and lists. They are the natural architecture of neural networks to use for such data [46].

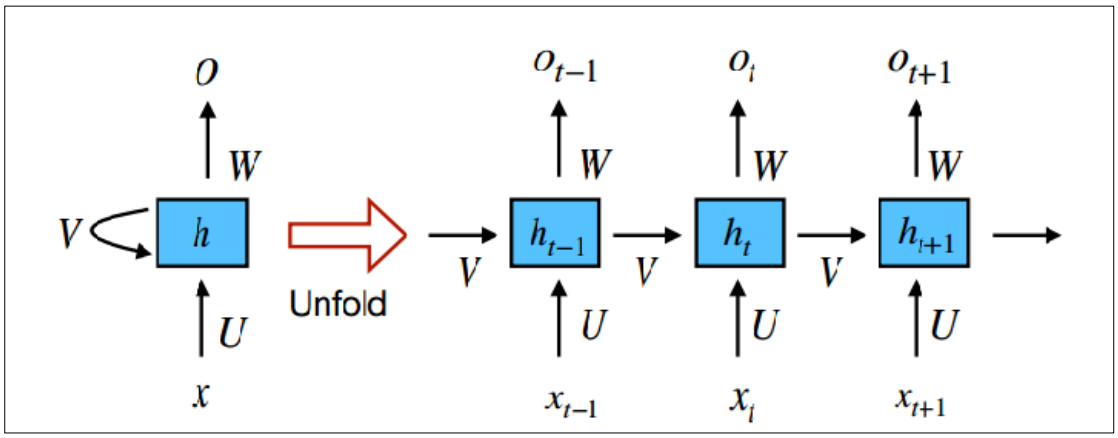


Figure 3.3 Simple Recurrent Neural Network (Figure Source: [46])

The hidden state of RNN is typically considered to be its most crucial element. As stated before, it can be considered as the network's memory element that accumulates information from one time step to the next and thus in theory it is stated that RNNs can make use of information in arbitrarily long sequence. In practice, however, it fails and RNNs suffers from the gradient vanishing/exploding problems, in which performance of the neural network suffers because it cannot be trained properly.

3.6 Long Short-Term Memory (LSTM)

Long short-term memory units (LSTMs), a special type of RNN, are designed to overcome the gradient vanishing/exploding problems of RNNs. Thus, they are capable of learning long-term dependencies. They were introduced by Hochreiter et

al., [32] in 1997 and are now widely used in solving a large variety of problems. The key to LSTM is the cell state which is controlled by three multiplicative gates namely input gate, forget gate, and output gate. These three gates control the flow of information whether to forget or pass on to the next time step. In other words, these gates enable the network to figure out what information is important and should be remembered and looped back into the network, and what information can be forgotten. This unique mechanism is the success of LSTM over RNNs to cope long-term dependencies problems. Figure 3.4 gives the basic structure of an LSTM unit.

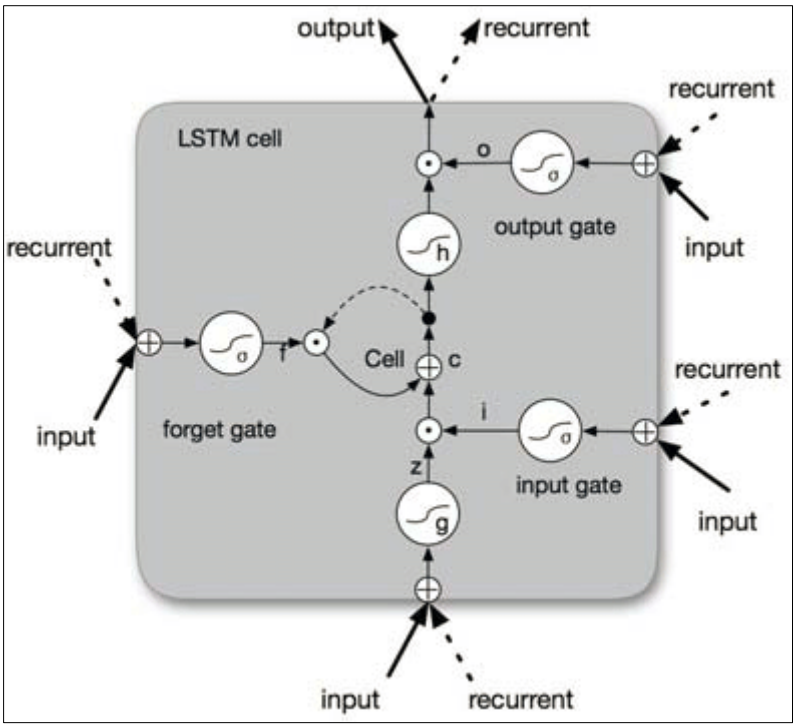


Figure 3.4 Schematic of LSTM Unit (Figure Source: [32])

The formulas to update each gate and cell state of input *x* are defined as follows:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \quad \text{Equation (3.1)}$$

$$f_{t=}\sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \quad \text{Equation (3.2)}$$

$$o_t = \sigma(W_{xo}x_o + W_{ho}h_{t-1} + b_o) \quad \text{Equation (3.3)}$$

$$c_n = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad \text{Equation (3.4)}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot c_n \quad \text{Equation (3.5)}$$

$$h_t = o_t \odot \tanh c_t \qquad \text{Equation (3.6)}$$

where σ denotes the sigmoid function; $\odot$ is the element-wise multiplication operator; *W* terms indicate weight metrics; *b* are bias vectors; and *i, f, o* represent input, forget and output gate respectively, and *c* are cell activation vectors. The hidden layer nodes are designated with the term *h.*

3.7 Bidirectional Long-Short Term Memory

For many sequence labeling tasks, it is beneficial to have access to both past (left) and future (right) contexts. However, the LSTM's hidden state h_t takes information only from the past knowing nothing about the future. Bidirectional LSTM network, a refinement with two LSTM layers, is designed to capture information of sequential data and have access to both previous and upcoming contexts. It is set up with two independent LSTMs, a forward and a backward propagating in two directions (see Figure 3.5). The forward LSTM acquires stream of input data and computes the forward hidden states. In the same way, the backward LSTM reads the sequence in reverse order and creates the backward hidden states. The intention behind is to create two separate hidden states for each sequence. As a consequence, the information of sequences from both directions is memorized. By concatenating the two hidden states, the final output is established. This advantage of maintaining information for long times in both directions can make significant improvement in linguistic computation.

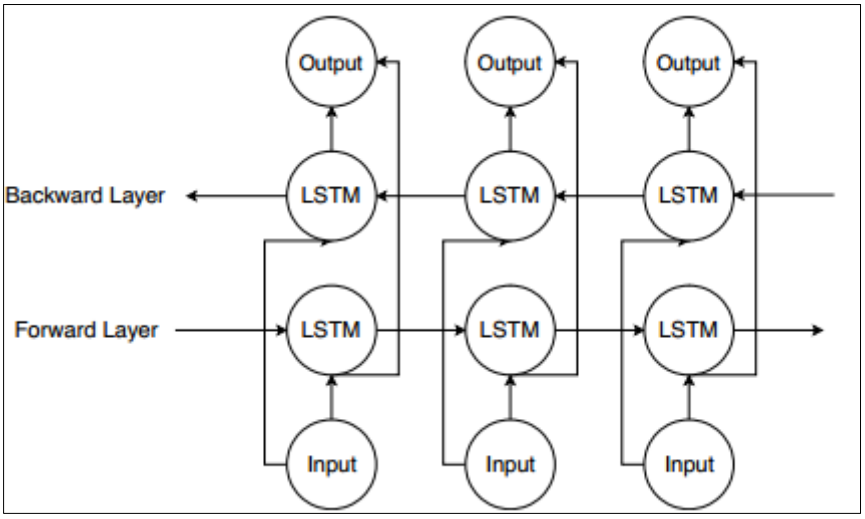


Figure 3.5 Bidirectional LSTM

3.8 Gated Recurrent Unit (GRU)

A Gated Recurrent Unit (GRU) is basically an LSTM variant without an output gate, which therefore fully writes the contents from its memory cell to the larger net at each time step. It was introduced by the work [15]. LSTM has three gates whereas GRU has only two gates: an update gate and a reset gate. The update gate decides how much of previous memory to keep around. The reset gate defines how to combine new input with previous value. In GRU, there is no persistent cell state distinct from the hidden state as in LSTM. GRU and LSTM have comparable performance and there is no simple way to recommend one or the other for a specific task. Generally, GRUs are faster to train and need fewer data to generalize. When there is enough data, an LSTM's grater expressive power may lead to better results.

3.9 Conditional Random Field (CRF)

Conditional Random Field (CRF) [39] [44] [76] [81] [95] is a statistical probabilistic model for structured prediction. It provides a probabilistic framework for labelling and segmenting sequential data, based on the conditional approach. There have been a lot of interests in CRFs by many researchers and CRF has been widely applied in NLP, computer vision, and many more.

For sequence labelling of general structured prediction tasks, it is beneficial to consider the correlations between labels in neighbourhoods and jointly decode the best chain of labels for a given sentence. Let *y* be a label tag sequence and *x* be an input sequence of words. Conditional models are used to label the observation input sequence *x* by picking the label sequence *y* that maximizes the conditional probability$p(y|x)$. To do so, a conditional probability is computed:

$$p(y|x) = \frac{exp^{Score(x,y)}}{\sum_{y'} exp^{Score(x,y')}} \qquad \text{Equation (3.7)}$$

where *Score* is determined by defining some log potentials $\log \psi_i(x, y)$ such that:

$$Score(x, y) = \sum_i \log \psi_i(x, y) \qquad \text{Equation (3.8)}$$

In here, there are two kinds of potentials: emission and transition.

$$Score(x, y) = \sum_i \log \psi_{EMIT}(y_i \rightarrow x_i) + \log \psi_{TRANS}(y_{i-1} \rightarrow y_i) \quad \text{Equation (3.9)}$$

In training, log probability of a correct tag sequence is maximized.

3.10 Summary

To sum up, deep learning is a brunch of machine learning that makes use of deep neural networks for powerful computation. The term deep usually refers to the number of hidden layers in the neural network. Although deep learning was first theorized in the 1980s, in its beginning, it was rarely applied because deep learning requires large amounts of labeled data and substantial computing power. CNN and RNN are examples of most popular types of deep neural networks.

There are a lot of aspects to consider while tuning the networks hyperparameters. Distribution of data is very important in neural training. Deep learning has given superior performance in most NLP research areas along with the power of distributed representation. Embedding is the key in extracting features in given input sentence.

By learning different portions of a feature space, CNN allows for easily scalable and robust feature engineering. RNN is powerful for learning sequential information. RNN encounters long term dependency problem while making use of information in arbitrarily long sequence due to its main shortcoming of the gradient vanishing/ exploding problems. However, this problem could be handled by LSTM and GRU, special kinds of RNN.

CRFs are undirected graphical models used to calculate the conditional probability of values on designated output nodes given values on other designated input nodes. It is beneficial to apply jointly CRF in model decoding in neural training for sequence learning.

CHAPTER 4
THE STATE OF MYANMAR LANGUAGE AND MYANMMAR NAMED ENTITY RECOGNINTION

This chapter has described the nature of Myanmar language, its characteristics, and the introduction of Myanmar Named Entity Recognition (NER). Furthermore, challenges encountered in processing named entity recognition for Myanmar language have also been discussed.

4.1 Outline of Myanmar Language

Myanmar language, formally also known as Burmese, is the official language of the Republic of the Union of Myanmar and has a long history of more than one thousand years. Myanmar language which is used by more than 50 million people is a kind of tonal language. In Myanmar language, syllable is the smallest linguistic unit and one word consists of one or more syllables.

Morphologically, Myanmar language is highly analytic with no inflection of morphemes which means that morphemes can be combined freely with no changes. Syntactically, Myanmar is typically head-final where the functional dependent morphemes succeeding content independent morphemes and the verb constituents working as the root of a sentence is always at the end of a sentence. Subordinate clauses are also placed before their modifying parts and before the main clause of a sentence. It is agglutinative and subject-object-verb (SOV) language. Orthographically, there is no specific rule or convention on the use of spaces to separate words in Myanmar, and spaces are actually used in Myanmar texts inconsistently to segment meaningful constituents [19] [31] [51] [78].

4.2 Outline of Myanmar Script

The Burmese or Myanmar script developed from the Mon script, which was adapted from a southern Indian script during the 8th century. The earliest known inscriptions in the Burmese script date from the 11th century. According to the documentations, the Myanmar scripts were descended from the Brahmi script of ancient South India and belong to the Sino-Tibetan language family.

Myanmar scripts are rounded in shape and are written in sequence from left to right. Between phrases, white space may occasionally be inserted but regular white space is not usually put between words (see Figure 4.1 for the example). However, sentences can be easily determined with sentence boundary maker "။". According to the historic facts, it can be said that the rounded appearance of scripts is a result of the use of palm leaves as the traditional writing material. Some historic fact reveals that the Myanmar script was originally adapted from the Mon script which was derived from Pali, the ancient Indian language of the text of Theravada Buddhism.

Word in Myanmar language is formed by a composition of multiple syllables or sometimes single syllable can stand as a word. Besides, a Myanmar syllable is composed of one or multiple characters. There are totally 75 characters in Myanmar scripts and these characters can also be further categorized as 12 groups.

Myanmar Sentence: ရန်ကုန်တွင်မိုးမရွာပါ။
English Sentence: It doesn't rain in Yangon.

Figure 4.1 Example of Myanmar Writing

4.2.1 Myanmar Word Formation

As described in previous section, in Myanmar written text, words are composed of one or more syllables. Likewise, a syllable may also include one character or more. For instance, a word 'နိုင်ငံ' is comprised of two syllables 'နိုင်' and 'ငံ'. In the same way, a syllable may be made up of one or several characters. For example, the syllable 'နိုင်' contains five characters, i.e., 'န, ိ, ု, င' and 'ိ' and similarly as the syllable 'ငံ' has two characters 'င' and 'ံ' (see Figure 4.2).

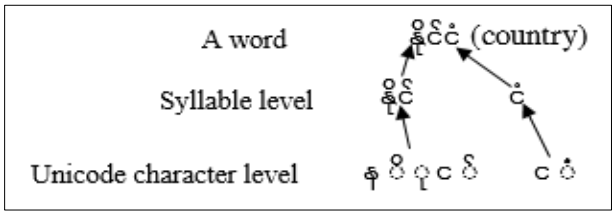


Figure 4.2 Example of Myanmar Word Formation

Generally, Myanmar words can either be standard words or special words such as stacked words. Example of these two types of word is expressed in Table 4.1. Besides, many Pali words, English loan words and transliterated words can also be found in Myanmar text.

Table 4.1 Examples of Standard Word and Special Word

Word Type	**Example Word**	**Meaning**
Standard Word	စာအုပ်	Book
Stacked Word	ဆန္ဒ	Desire

4.2.2 Myanmar Unicode

There are some complex encoding problems for Myanmar language. There are some other encodings that are created based on character image, and most Myanmar people are not only using these kinds of encodings that do not follow standard Unicode encoding but also more familiar with them. Myanmar syllable structure is quite easy to define when with the use of Unicode encoding. Hence, in order to represent Myanmar syllable structure in a definite way, the Unicode encoding is used in this work. According to the Unicode standard version 12.0, Myanmar characters range from U+1000 to U+109F [33] [99]. Unicode Code Point for all Myanmar characters can be seen in [33] and [99]. Some of Myanmar font scripts which follow Unicode Code Point are Myanmar2, Myanmar3, Pyidaungsu, Parabaik, Padauk, Thanlwin, WinuniInnwa, Win Myanmar, MyMyanmar, Yunghkio, Panglong, and Tharlon.

4.2.3 Myanmar Characters

Basically, as mentioned in section 4.2, there are 75 characters in total in Myanmar writing script. These characters can be further divided into 12 groups (see Table 4.2). They are 34 consonants, 4 medial letters, 8 dependent vowels, 1 Sign Virama and 1 Sign Asat [51]. Three independent vowels and 3 independent various signs can be considered as one group and the characters in this group can act as stand-alone syllables. Another group is formed with 4 independent vowels and 1 Myanmar Symbol Aforementioned. Myanmar Letter Great Sa is always preceded by a consonant and is never written alone. Moreover, there are 10 Myanmar digits and 2

punctuation marks. In addition to these characters, white space is used between phrases but there is no definite rule to use it.

Table 4.2 Classification of Myanmar Characters and their Associated Unicode Code Point

Category	Myanmar Characters	Unicode Code Point
Consonants	ကခဂဃငစဆဇဈညဋဌဍဎဏတထဒဓနပဖဗဘမယရလသဝဟဠအ	U+1000…U+1021
Medials	ျ ြ ွ ှ	U+103B…U+103E
Dependent Vowel Signs	ာ ါ ိ ီ ု ူ ေ ဲ	U+102B…U+1032
Myanmar Sign Virama	်	U+1039
Myanmar Sign Asat	်	U+103A
Dependent Various Signs	ံ ့ း	U+1036…U+1038
Independent Vowels, Independent Various Signs	ဤ ဧ ဪ ဿ ၌ ၍ ၏	U+1024; U+1027 U+102A; U+104C; U+104D; U+104F;
Independent Vowels, Myanmar Symbol Aforementioned	ဣ ဥ ဦ ဩ ၎	U+1023; U+1025; U+1026; U+1029; U+104E;
Myanmar Letter Great Sa	ဿ	U+103F
Myanmar Digits	၀ ၁ ၂ ၃ ၄ ၅ ၆ ၇ ၈ ၉	U+1040…U+1049
Punctuation Marks	၊ ။	U+104A…U+104B
White space		U+0020

Consonants are also known as ‘Byee’ in the Myanmar language. Unicode encodes the consonants between (U+1000) and (U+1021). Note that the consonants ‘ည’ and ‘ဉ’ are stored as different codes, although they can be considered the same consonant. Consonants serve as the base characters of Myanmar words, and are similar in pronunciation to those of other Southeast Asian scripts, such as Thai, Lao and Khmer. Medials are known as ‘Byee Twe’ and vowels are known as ‘Thara’ in Myanmar. Vowels are the basic building blocks of syllable formation in the Myanmar language, although a syllable or a word can be formed without a vowel.

Two noteworthy special characters are Myanmar Sign Virama (U+1039) which is also called ‘Htutsint’ and Myanmar Sign Asat (U+103A). Virama (U+1039) is also known as stacking, and Asat (U+103A) is commonly called ‘Killer’. Myanmar Sign Virama is invisible when it is typed, but changes the rendering of characters by stacking one consonant above the following one, example ‘မ္ဘ’ in the word ‘ကမ္ဘာ’[78].

Myanmar numerals are decimal-based, and Table 4.2 shows zero to nine in sequence. No thousand separators are used; instead, spaces are sometimes used between digits for easy reading. The two punctuation marks function in a similar manner to the comma and the period in English, respectively.

4.2.4 Syllable Structure of Myanmar Language

A syllable is a basic sound unit or a sound. A word can be made up of one or more syllables. Further, Myanmar syllable structure can be represented in two ways namely phonetic and orthographic representations [19] [31] [51]. Every syllable boundary can be a potential word boundary. In some cases, a word can include other words, in which case it is called a compound word.

In Myanmar text, a syllable is formed based on rules that are quite definite and unambiguous. A Myanmar syllable has a base character, and may also have (or not) a pre-base character, a post-base character, an above-base character and a below-base character [78].

Figures 4.3 and 4.4 illustrate how a syllable is formed. Regardless of the appearance of the characters on the screen, the characters are to be stored consistently in a sequence specified by the Unicode standard. For example, according to the Unicode standard, in computers, vowels are stored after the consonant. Further, the order in which the characters are stored may not be the same as their keyboarding sequence.

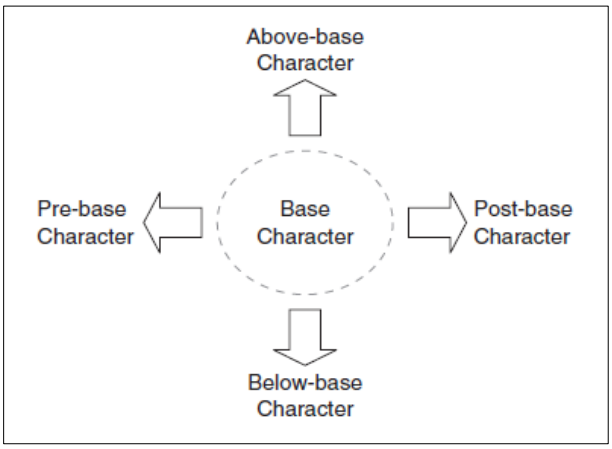


Figure 4.3 Positions of Characters in a Myanmar Syllable (Figure Source: [78])

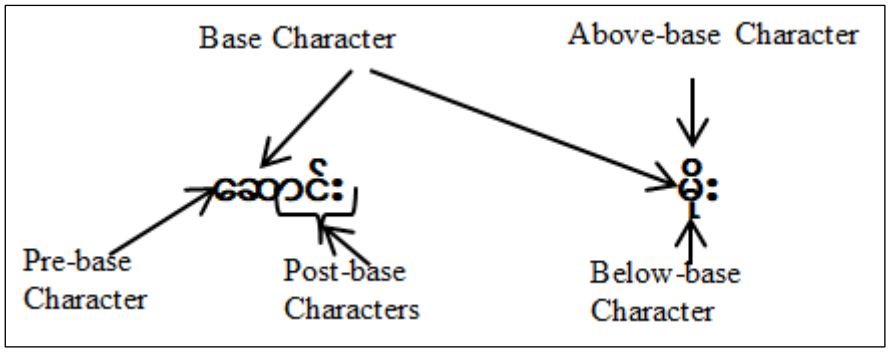


Figure 4.4 Two Myanmar Syllables

A syllable can contain multiple consonants, multiple medials and multiple vowels. These constituents can appear in different sequences; e.g. consonants followed by medials followed by vowels, or consonant, then medials, then vowels, followed by more consonants and vowels. In other words, a Myanmar syllable consists of one consonant as initial character, followed by zero or more medials, zero or more vowels as well as optional dependent various signs. [19] [31] [51].

As presented previous section, independent vowels and independent various signs can act as stand-alone syllables and do not need any medials or vowels to become a syllable or a word and these free standing vowels and digits can be syllables by themselves.

4.2.4.1 Syllabification for Myanmar Language

Syllabication is the task of breaking words into syllables. Syllables are the smallest linguistic units that are the building blocks of words. Syllable segmentation is essential for the language processing of Myanmar script. Languages differ considerably in the syllable structures that they permit. For most languages, syllabification can be achieved either by writing a set of rules or using annotated corpus which is a data-driven approach. Various syllabification algorithms have been proposed for different languages by using different approaches.

Moreover, many language-specific syllabification methods have been modeled by using finite state machine or neural networks and Finite State Transducers for multilingual syllabification. Some syllabications are performed based on phonetic representation. However, such phonetic syllabication cannot be used for some major Myanmar language processing tasks so that it is necessary to use orthographic syllabication for these tasks.

Automatic syllabication for Myanmar word is challenging. There are also Irregular words in Myanmar writing. Thus, it is necessary to handle syllabication of such irregular word forms. As for Irregular words such as stack words e.g., "တက္ကသိုလ်" (University), loan words e.g., "ဘတ်စ်ကား", usage of Great Sa in the word "ပြဿနာ"(problem), and contraction words e.g., "ယောကျ်ား" (man), it is difficult to syllable words based on phonological representation.

Moreover, apart from these kinds of irregular words, in Kinzi where final consonant can follow the main consonant and the resulting syllable is called closed syllable. In such words, the combination of consonant Nga "င" and vowel killer "ိ" is "င်" which is not written on the line as the usual way, but is placed above the first consonant of the next syllable (e.g., "အင်္ဂလန်" England).

In multisyllabic words derived from an Indian language such as Pali, where two consonants occur internally with no intervening vowels, the consonants tend to be stacked vertically, and the Asat sign is not used. Some words can be written in both in standard syllable structure and contracted form. In Unicode Technical Note [33], diacritic storage order of Myanmar characters in Unicode (which we refer as sub-syllabic components) is explained in detail. Such kind of language specific features makes Myanmar syllable segmentation task complicated.

Regarding Myanmar syllabification, corpus-based longest matching approach had been done in [34]. In this approach, the authors collected 4,550 syllables from different resources. The input texts are syllabified by using longest matching algorithm over their syllable list. They observed that only 0.04% of the actual syllables were not detected and described their failures because of three facts: (1) differing combinations of writing sequences, (2) loan words borrowed from foreign languages and (3) rarely used syllables not listed in their syllable list.

Rule-based Myanmar syllable segmentation is done by [51] in which input text strings are converted into equivalent sequence of category form and compares the converted character sequence with the syllable rule table to determine syllable boundaries. The authors tested 32,238 syllables in the Myanmar Orthography published by Myanmar language commission [97] and the experimental results show an accuracy rate of 99.96% for segmentation. However, their approach cannot solve for the segmentation of irregular words with traditional writing forms namely Kinzi, consonant stacking, Great Sa and English loan words with irregular forms.

Manually constructed context free grammar (CFG) with “111” production to describe the Myanmar syllable structure is presented in [30]. The authors made their CFG in conformity with the properties of LL (1) grammar so that conventional parsing technique called predictive top-down parsing can be utilized to identify Myanmar syllables. Myanmar syllable structure was presented in accordance with orthographic rules. The preprocessing step called contraction for vowels and consonant conjunctions was also discussed. They made LL (1) grammar in which “1” does not mean exactly one character of look ahead for parsing because of the above mentioned contracted forms. They used five basic sub syllabic elements to construct CFG and found that all possible syllable combinations in Myanmar Orthography can be parsed correctly using their proposed grammar.

The authors of [31] proposed a method that focuses on orthographic way of syllabification using finite state transducers (FST) to tackle syllabification of Myanmar words with standard syllable structure as well as words with irregular structures. Un-weighted finite state transducer to divide Myanmar words into syllables was proposed. Syllable structure model was represented in Chomsky`s regular grammar and deployed finite state transducers for automatic syllabification of Myanmar Unicode texts. Their FST based method was tested by using a text corpus containing 11,732 distinct words yielding 32,238 syllables covering all possible syllable structure in standard words and irregular words from Myanmar Orthography published by Myanmar Language Authority [97] on Stuttgart Finite State Transducer Tool (SFST) and achieved the accuracy of 99.93%.

4.3 Myanmar Named Entity Recognition

Myanmar NER is essential to the development of Myanmar NLP. Currently, there is no publicity available NER tool that can efficiently recognize names written in Myanmar language.

It is not easy to detect names in Myanmar text because of its language nature and some other reasons. As far as it is being considered, there were only two previous attempts on Myanmar NER.

4.3.1 Nature of Myanmar Names

In the naming systems of other countries, a person name typically consists of a family name and a given name. However, in most Myanmar names, such a type of naming system is not used. Generally, name affixes, for example general titles, can be used with a person name and are considered part of name. For instance, in many languages it is quite common for person names to be preceded by some kind of title. Likewise, in Myanmar language, generational titles precede the person name, e.g., ဒေါ်၊ ကို၊ မ၊ မောင် etc., even though some writing can omit these titles.

Several name affixes can be used with person names, which provide information about the person, indicate that the individual holds a position, educational degree, accreditation, office, or honor. These are usually not considered as parts of a name, but they provide examples of external evidence in recognition names in sentences. In Myanmar, these affixes stand before the name, e.g. General, Professor, and Doctor, etc. Thus, if an indicator is preceded by such a personal prefix, it is likely to be a person name. However, in Myanmar script, names occasionally use or not a preceding indicator word to describe a person's status, age and gender. In sentences, person names may be in a subject place, object place, possessive place, and comparison place and even in compatibility.

Location names can also contain typical affixes which can help recognize several location types. For example, the words such as လမ်း (street), မြို့ (city), (township), and နိုင်ငံ (country) usually follow after location names. But in some sentences, location names can appear without these kinds of affixes. In some case, two location names appear constitutively one after another without any words between them (e.g., ပြင်ဦးလွင်နန်းမြိုင်ဟိုတယ်တွင်ကျင်းပမည်။). In this sentence ပြင်ဦးလွင်(Pyin Oo Lwin) and နန်းမြိုင်ဟိုတယ် (Nan Myein Hotel) are names of location. In such case, it would be difficult to define rule to detect name with clue suffixes and prefixes.

Example sentences with person names and location names are described in the following sentences. Person names and location names are highlighted in bold.

- **မြသီတာ**သည် **သီဟ**ကို အကူအညီပေးသည်။

  Mya Thida gives a hand **Thiha**.

- ပါမောက္ခဦး**မောင်မောင်**သည်**ကျပန်**နိုင်ငံမှပြန်လာခဲ့သည်။

- ပါမောက္ခဦး**မောင်မောင်**သည်**ဂျပန်**မှပြန်လာခဲ့သည်။

  Prof. U **Maung Maung** came back from **Japan**.

Most organization names are complex phrases, consisting of several words. Recognizing organization names is quite difficult, sometimes even impossible without considering external evidence. However, there are organization names which share a common feature, such as typical suffixes like ဟိုတယ် (Hotel), ဘဏ် (bank), ကုမ္ပဏီ (company), ကော်မတီ (committee), ဝန်ကြီးဌာန (Mininstery), and ကုမ္ပဏီလိမိတက် (Co.Ltd). For instance:

- **အေးသူဇာ**သည် **ပညာရေးဝန်ကြီးဌာန** တွင်အလုပ်လုပ်သည်။

  Aye Thuzar woks for **Ministery of Education**.

- **ဧရာယူနတီဟိုတယ်**သည်**ဧရာကုမ္ပဏီလိမိတက်**မှပိုင်ဆိုင်သည်။

  Ayar Unity Hotel belongs to **Ayar Co.Ltd.**

There are more ambiguous facts to consider in NER for Myanmar Language. Some names make the NER process complicated in classifying into specific name types. As an example, the name "မြသီတာ" can be a person name or an organization name or even a location (street) name. Likewise, names of days, months and year can also be found as names of person, location or organization. Moreover, the spelling of names may not be consistent and differ from sentence to sentence. Number digits can also be found as location names. All of these are about the nature of names in Myanmar language that need to consider when processing NER.

4.3.2 Challenges in Myanmar Named Entity Recognition

To address the problem of recognizing names automatically in Myanmar written text is more complicated than other languages for many reasons. One of the reasons is the lack of linguistic resources which is necessary in language computing. The resources which are necessary for Myanmar NER such as NE annotated corpora, name lists, gazetteers or name dictionaries, etc. are not publicly available until now. Therefore, Myanmar language is said to be under-resourced language.

Besides, Myanmar language has distinct characteristics if compared with other languages which make NER not a simple task. There is no capitalization feature in Myanmar language; it the main indication of proper names for some other languages.

Another fact is that its writing structure has no definite order, thus making the NER a difficult process.

There is no definite spelling for names so that names appear in free writing form. There are wide variations of spelling in some Myanmar terms for not only proper names but also for some loanwords or transliterated words. Myanmar names also take all morphological inflections which can lead to ambiguity, and more, this ambiguity of NE may cause difficulty in assigning NEs with predefined NE categories. It can be said that how to address the issue of NER in Myanmar text automatically is not a simple task and still challenging. Myanmar NER should be addressed in order to develop Myanmar NLP.

4.4 Summary

To sum up, this chapter firstly describes the notable features of Myanmar language which is also known historically as the Burmese. Myanmar language is a tonal language and syllable-based language. Text run from left to right. Spaces are inserted to separate phrases rather than words. Type of writing system for Myanmar language is syllabic alphabet - each letter has an inherent vowel [a]. Other vowels sounds are indicated using separate letters or diacritics which appear above, below, in front of, after or around the consonant. Myanmar syllable structure is well-defined and unambiguous.

Syllabification for Myanmar language plays an important role for language processing. As for approaches for Myanmar syllabification, general rule-based as well as corpus-based had been carried out. Besides these approaches, finite state transducers (FST) based method had also been proposed.

Named Entity Recognition for Myanmar language is in initial state. Moreover, NER for Myanmar language is said to be quite difficult compared to other languages because of its complex nature. The proposed NER for Myanmar language is described in detail in Chapter 5.

CHAPTER 5
SYLLABLE-BASED NEURAL NAMED ENTITY RECOGNITION FOR MYANMAR LANGUAGE

Named Entity Recognition (NER) for Myanmar scripts is crucial to the area of Myanmar Natural Language Processing (NLP) research. Moreover, NER has been a challenging problem in Myanmar language processing. Currently, Myanmar NLP has been emphasized to be developed. However, required lexical resources are very little, and these resources are not publicly available. Myanmar language is morphologically rich and complex language as well as it has ambiguity. These factors can affect word segmentation and also the performance of NER. On the other hand, it is necessary to recognize names properly to assist for the sophisticated NLP systems. Moreover, the main motivation behind this thesis is that at present time, there is no available NER tool that can recognize NEs automatically in Myanmar written texts.

With the advances of deep learning, neural sequence labeling models have achieved state-of-the-art for many tasks. Neural sequence models have the ability of minimizing the burden of statistical approaches which is totally reliance on feature engineering because of the fact that relevant and appropriate features are extracted automatically through deep neural networks structures. Various neural network architectures were developed, proposed and applied for sequence labelling tasks and have been gained popularity in sequence labeling. With the fast evolution of deep learning, those kinds of deep neural networks have been revealed that they significantly outperform statistical algorithms in several recent research works.

Recurrent Neural Networks (RNNs) have gained popularity for modeling sequential data and they have been reported to be very efficient in NER as a sequential tagging task. Long Short-Term Memory (LSTM) neural network, a special kind of RNN, has been verified to be more powerful in modeling sequential data. At the same time, two independent LSTM layers are stacked to establish a bidirectional LSTM neural network with the idea of accumulating contextual information from both previous and upcoming directions. Bidirectional LSTM network has dominant ability in maintaining information of sequence for long periods from both directions; making great improvement in linguistic computation. Besides, the ability of Conditional Random Fields (CRF) which can use the past input features and sentence

level tag information and also the future input features, gives the better accuracy and it becomes the choice for decoding technique in the sequence tagging problem.

Based on the representation of input tokens (words) in a sentence, neural architecture for NER can be broadly classified into categories such as character-based or word-based. The input representations may be based on characters, words, other sub-word units or any combination of these. In this effort, syllables, sub-words units are considered for basic input representation. Moreover, NER for Myanmar language is viewed as a sequence tagging problem. This research investigates the benefits of deep neural networks on NER for Myanmar language. Experiments are conducted by applying several deep neural network architectures on syllable level Unicode Myanmar contexts.

This work contributes the first deep neural architecture for Myanmar NER that represents syllable-level (sub-word) units are given as input to networks and concatenates syllable embeddings with CNN over the characters of a syllable, and passing this representation through another sentence-level bidirectional LSTM, after that predicting the final label tags using CRF layer. For the experiments of this proposed neural NER to be performed and for further research on Myanmar NER, a very first manually annotated NE tagged corpus for Myanmar language is also constructed and proposed as part of the contribution of this research.

This research contributes the first evaluation of deep neural network models on NER problem for Myanmar language and also compares with the baseline statistical CRF model. This effort also intends to discover the usefulness and advantages of deep neural network approaches to Myanmar textual language processing as well as to promote further research on this underdeveloped language.

5.1 Work Flow of Syllable-based Neural NER Modeling and Myanmar NER System

The overview work flow of neural NER modeling and the implementation of Myanmar NER system is illustrated in Figure 5.1. The manually NE tagged corpus is firstly built in order to apply in modeling. An NER model is then developed by conducting various neural training experiments with several deep neural architectures. The best model obtained from the experiments is applied in implementation of Myanmar NER system.

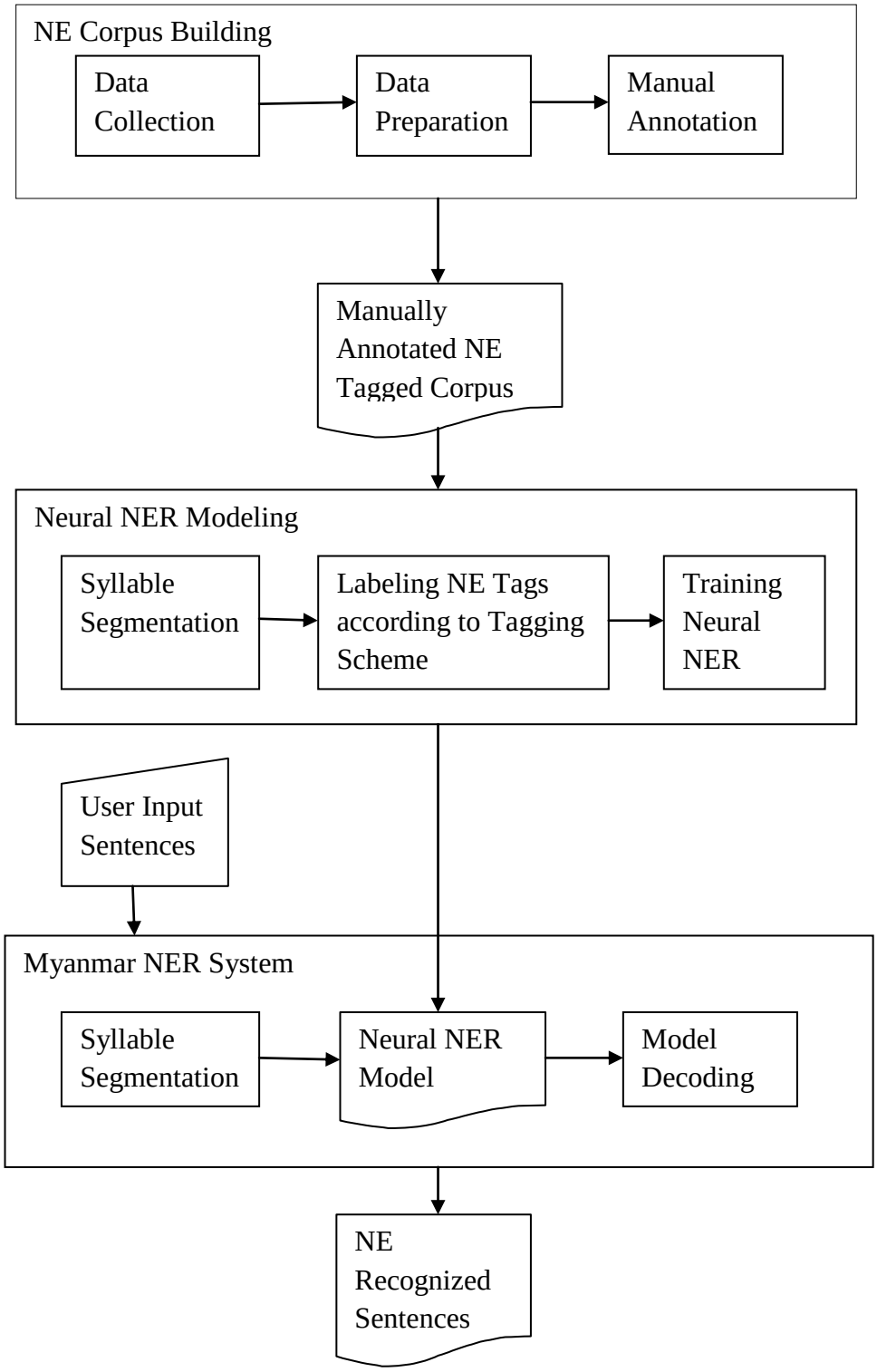


Figure 5.1 Work Flow of Neural NER Modeling and Myanmar NER System

5.2 Development of Myanmar NE Tagged Corpus

Annotated corpora are a vital resource for NLP and information extraction approaches which employ machine learning techniques. Building annotated NE tagged corpus is also the first step in training NER model and implementing NER system especially for low-resourced languages where there is no pre-created NE tagged corpus. Although corpora are available for other languages, resources for Myanmar NLP and Myanmar NER are still limited. This is one of the main reasons why Myanmar NLP lagged behind when compared to others.

As far as being aware, there is no publicly available NE tagged corpus for Myanmar language. However, there are many benchmark data resources available for other languages to perform NER. CoNLL-2003 dataset [80] which includes 1,393 English and 909 German news articles, MUC-6 and MUC-7 [98] provided through their Shared Task [65].

Besides, there were a lot of efforts to develop NE tagged corpus for various languages. Development of Bengali NE tagged corpus was described in [21]. The authors in [36] presented a workflow of building an English-Vietnamese NE corpus from an aligned bilingual corpus. In [29], a method to automatically build a NE corpus based on the DBpedia ontology was proposed. Likewise, construction of Portuguese NE corpus was proposed by using DBpedia as well [82].

In the following section, the development of the manually annotated NE corpus for Myanmar language to support the evaluation of syllable-based neural named entity extraction will be presented. It took nearly one year to annotate the NE corpus manually. Currently, over 60K sentences in total are manually annotated according to defined NEs tags. To be exact, there are totally 60,500 sentences and containing total number of 174,133 named entities. There is no other available Myanmar NE tagged corpus that has as much data as this proposed NE tagged corpus.

5.2.1 Data Collection

Nowadays, huge amount of web data are available and become the main source of data for computational research processing. In developing Myanmar NER corpus, news sentences written in Myanmar scripts within a range of year from 2016

to 2019, from online official news website such as BBC Burmese[1], 7days Daily news[2], Eleven media news[3], Mizzima Burmese[4], Kumudra[5], State Counselor[6], and Presidential Office[7], The Voice[8] , Myanmar Times[9], VOA Burmese[10], and so on are collected and used. Different types of new gender including business, crime, health, tourism, education, environment, technology, sport, religion, and also politics are organized.

In addition, sentences provided from ALT-Parallel-Corpus [96], which is one part of the Asian Language Treebank (ALT) project under ASEAN IVO, are also used. A lot of transliterated names appear in these sentences because sentences from ALT corpus are translated from International news.

5.2.2 Data Preparation

As data preparation, data cleaning is firstly carried out. All kinds of mistyped errors are corrected manually. Besides, different encodings need to be in a uniform encoding. Therefore, for encoding consistency, all the collected data are converted into standard Unicode encoding. In sentences, some typing errors are found. Especially, the digit "၀" (zero) and the consonant "o" ("Wa") are mistyped. Thus it is necessary to correct such kind of wrongly typed error because the quality of data strongly affects the performance.

5.2.2.1 Defined NE Types

To indicate NEs in sentences, each NE has to be annotated with NE tag. In this work, totally six types of NE tags are defined for manual annotation: PNAME, LOC, ORG, RACE, TIME, and NUM.

PNAME tag is used to indicate person names including nickname or alias, while LOC tag is defined for location entities. In this case, politically or geographically defined places (cities, provinces, countries, international regions,

[1] https://www.bbc.com/burmese
[2] https://7daydaily.com/
[3] https://news-eleven.com/
[4] http://www.mizzimaburmese.com/
[5] https://www.kumudranews.com/
[6] https://www.statecounsellor.gov.mm/
[7] https://www.president-office.gov.mm/
[8] http://thevoicemyanmar.com/
[9] https://myanmar.mmtimes.com/national-news.html
[10] https://burmese.voanews.com/

bodies of water, mountain, etc.) are considered as location entities. In addition, location entities include man-made structures like airports, highways, streets, factories and monuments, etc., as well. ORG tag is defined to annotate names of organizations (government and non-government organizations, institutions, agencies, corporations, companies and other groups of people defined by an established organizational structure).

In our Myanmar language, some location names (especially state names) and names of national races have same spelling in writing scripts. For example, the location name "ကရင်" (Kayin State) and one of the national races "ကရင်" (Kayin race). For this reason, the NE tag RACE is defined to specify names of national races. TIME is used for dates, months and years. In this case, not only words indicate time (e.g., နိုဝင်ဘာ) but also time format written in numeric form (e.g., ၁၅.၁၁.၂၀၁၉) are annotated with TIME tag. NUM tag is used to point out number format in sentences. Another tag O indicates words which are not part of any defined NE types in sentences. The symbol "|" separates the boundary of each tag. Some sample annotated sentences from the NE tagged corpus are described in Figure 5.2.

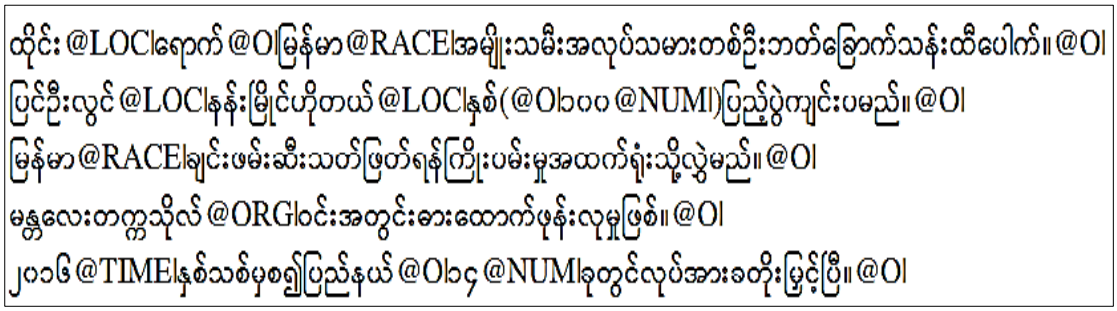

ထိုင်း@LOC|ရောက်@O|မြန်မာ@RACE|အမျိုးသမီးအလုပ်သမားတစ်ဦးဘတ်ခြောက်သန်းထီပေါက်။@O|
ပြင်ဦးလွင်@LOC|နန်းမြိုင်ဟိုတယ်@LOC|နှစ်(@O|၁၀၀@NUM|)ပြည့်ပွဲကျင်းပမည်။@O|
မြန်မာ@RACE|ချင်းဖမ်းဆီးသတ်ဖြတ်ရန်ကြိုးပမ်းမှုအထက်ရုံးသို့လွှဲမည်။@O|
မန္တလေးတက္ကသိုလ်@ORG|ဝင်းအတွင်းဓားထောက်ဖုန်းလုမှုဖြစ်။@O|
၂၀၁၆@TIME|နှစ်သစ်မှစ၍ပြည်နယ်@O|၁၄@NUM|ခုတွင်လုပ်အားခတိုးမြှင့်ပြီ။@O|

Figure 5.2 Example Sentences from Myanmar NE Tagged Corpus

Further, the description of defined NE tags categories and some sample usage of each NE category are shown in Table 5.1.

Table 5.1 Defined NE Types and their Usage

Defined NE Types	Example Usage
PNAME	ကြယ်စင်၊ အေးအေးမွန်၊ အိုဘားမား
LOC	မြန်မာ၊ မားစ်၊ ရန်ကုန်၊ မန္တလေး၊ မုံရွာ

ORG	ရန်ကုန်ကွန်ပျူတာတက္ကသိုလ်၊ ရိုးမဘဏ်
RACE	ဗမာ၊ ကချင်၊ ချင်း၊ ကိုရီးယား
TIME	နိုဝင်ဘာ၊ တန်ဆောင်မုန်း၊ သောကြာ၊ ၁၅.၆.၂၀၁၉
NUM	၅၆၇၊ ၁၀၀၀၊ ၃.၁၄

Table 5.2 lists the entities distribution in this manually annotated NE tagged corpus. The occurrence of each NE in the corpus is also shown in percentage. It is observed that location entity is the most occurred type in the corpus. The race type is the least found entity in the corpus; only 3% is appeared in the corpus.

Table 5.2 Corpus Data Statistics

Data	**Total number**	**Occurrence of Each NE (%) in NE Tagged Corpus**
Sentences	60,500	
Number of NE	174,133	
PNAME	35,405	20
LOC	63,492	36
ORG	19,740	11
RACE	5,720	3
TIME	29,472	18
NUM	20,304	12

The occurrence of each defined NE type in the annotated NE tagged corpus is also shown in Figure 5.3. Among all defined NE types, the LOC type is the most appeared entity in the corpus which is over 35% out of all NEs. The entity RACE is the least occurred entity which is less than 5%.

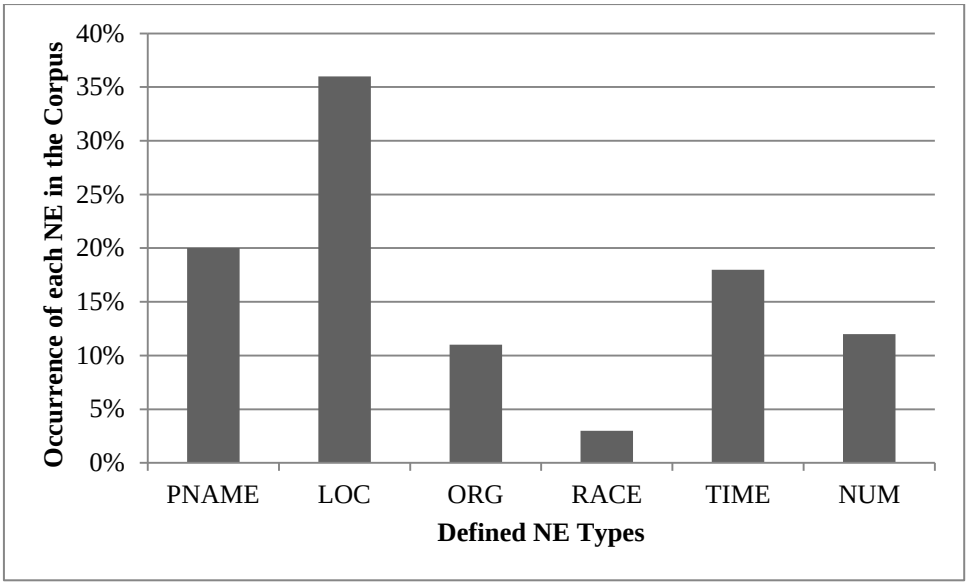


Figure 5.3 Occurrence of Defined NE Types in Myanmar NE Tagged Corpus

5.2.2.2 Syllable Segmentation

Experiments were conducted as syllable-based sequence labeling, in which various deep neural networks were trained. In order to convert the NER problem into a sequence tagging problem, a label is assigned for each token (syllable) to indicate which tokens are the named entities (NEs), and which are not the NEs in training and test data.

As for syllabication, the Myanmar syllable segmentation algorithm "sylbreak" of [43] is utilized on sentences for the syllable data representation and syllable-based labeling. Syllable-level segmented sentence is shown as an example in Figure 5.4. Each syllable is separated with white space in Figure 5.4.

ထိုင်း နိုင် ငံ ရောက် မြန် မာ အ လုပ် သ မား မ သီ ရိ ရွှေ စင် သည် ကံ ထူး ခဲ့ သည် ။

Figure 5.4 Example of Syllable-level Segmented Myanmar Sentence

5.2.2.3 Tagging Scheme

In order to transform the NER problem into a sequence labeling problem, each token (syllable) must be assigned with a label to indicate the NE boundary in sentence. As tagging scheme, BIOES (Beginning, Inside, Outside, End and Single) scheme is utilized for all the experiments (see Figure 5.5).

A single named entity could span several tokens within a sentence. Sentences are usually represented in the BIO format (Beginning Inside, Outside) where every token is labeled as B-label if the token is the beginning of a named entity, I-label if it is inside a named entity but not the first token within the named entity, or O otherwise.

However, in this work, it is decided to use the BIOES tagging scheme, a variant of BIO commonly used for named entity recognition, which encodes information about singleton entities (S) and explicitly marks the end of named entities (E). Using this scheme, tagging a word as I-label with high-confidence narrows down the choices for the subsequent word to I-label or E-label, however, the BIO scheme is only capable of determining that the subsequent word cannot be the interior of another label. The paper [65] showed that using a more expressive tagging scheme like BIOES improves model performance marginally.

Figure 5.5 shows the data format example of one sentence with BIOES tagging scheme for NER as sequence learning. Each token (syllable) must be represented in one line, with columns (features and labels) separated by white spaces or tabular characters. A sequence of token (syllable) becomes a sentence.

This work has two columns: token and label tags. To identify the boundary between sentences, an empty blank line is put. It has been defined 6 name entities for this work, and thus there are totally 25 label tags for NER as sequence learning in all experiments.

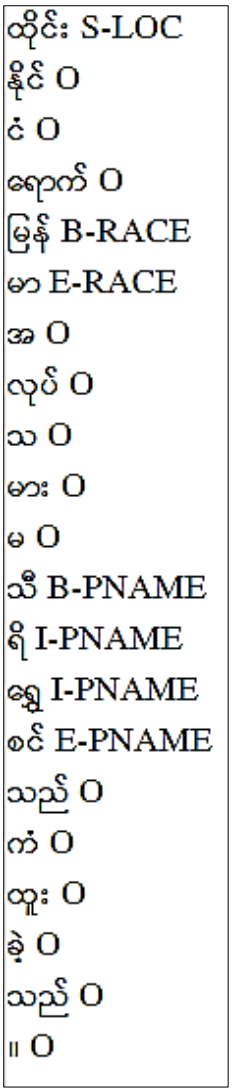


Figure 5.5 Data Format Example with BIOES Tagging Scheme

5.3 Neural Modeling for Myanmar NER

Detailed explanation of the proposed neural model architecture for Myanmar NER and training setup steps are described in following sections.

5.3.1 Experimental Setup

To implement the neural network model training, the neural libraries provided by the PyTorch framework [61] are utilized because it provides flexible choices of feature inputs and output structure. Experiments are run on Nvidia Tesla K80 GPU. Based on different parameter settings, the training time for each experiment is different.

Experiments are conducted by comparing different neural sequence models on syllable level text rather than word level. The performance results are compared with baseline statistical CRF models as well. During neural training, the power of various neural network architectures and also the effect of different parameters settings on Myanmar NER were investigated. For each training, syllable and character level features are automatically detected by applying different networks (bidirectional LSTM, CNN and also GRU architecture) one after another to eliminate the need for most feature engineering.

5.3.2 Input Representation

The input layer to the model is vector representations of individual tokens (characters/ syllables).

Learning independent representations for syllable types from the limited NER training data is inconvenient: there are simply too many parameters to be reliably estimated. For the reason that many languages have orthographic or morphological evidence that something is a name (or not a name), representations that are sensitive to the spelling of words give helpful information. For this reason, a model that constructs representations of syllables from representations of the characters they are composed of is used.

This approach of learning character-level features while training instead of hand-engineering prefix and suffix information of words is the main distinction of this work from most previous approaches. Learning character-level embeddings has the advantage of learning representations specific to the task and domain at hand. It has been found that learning character-level embedding is useful for morphologically rich languages and to handle the out-of-vocabulary (OOV) problem.

As to this setting, character representations are gained from training data by applying different neural networks (CNN, bidirectional LSTM and also GRU). In the experiments with CNN, the same structure as [49], one layer CNN structure with *max-pooling* is used to capture character-level representations. Further, as the character sequence layer, bidirectional LSTM is also used to capture the left-to-right and right-to-left sequence information, and the final hidden states of two LSTMs are concatenated as the encoder of the input character sequence.

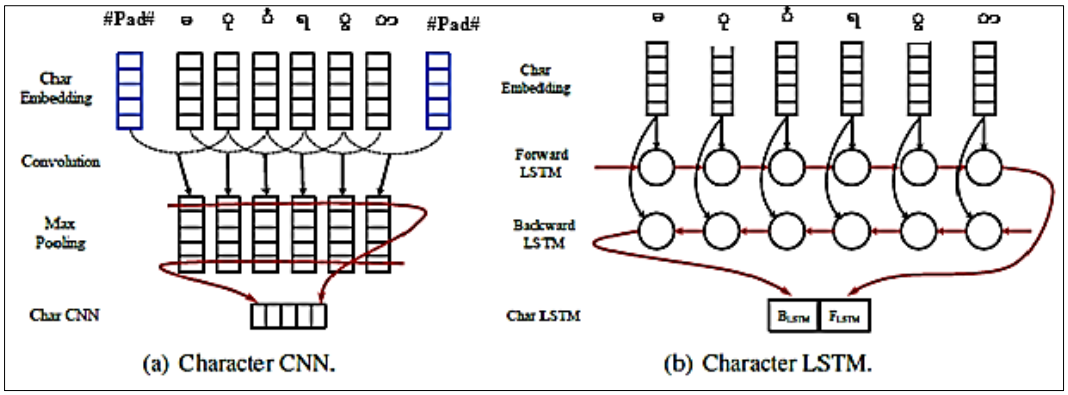


Figure 5.6 Neural Character Sequence Representation

Figure 5.6 (a) shows the CNN structure on characters representing the location name "မုံရွာ", containing two syllables and Figure 5.6 (b) demonstrates the utilization of bidirectional LSTM structure on the character sequence of each syllable as character sequence representations. This character-level representation is then concatenated with a syllable-level representation from a syllable lookup-table.

5.3.3 Pretrained Embeddings

The intuition about names is that names, which may individually be quite varied, appear in regular contexts in large corpora. Therefore, embeddings learned from a large corpus that are sensitive to word order are used.

As in Collobert et al. [16], pretrained word embeddings are utilized to initialize lookup table. Embeddings are pretrained using skip-n-gram, a variation of word2vec [53] that accounts for word order. These embeddings are fine-tuned during training. Word embeddings are trained using the 200K sentences, a variety of news data collection, wiki news data and Myanmar sentences from ALT-parallel-corpus data. An embedding dimension of 100, a minimum word frequency cutoff of 4, and a window size of 8 are used. However, unfortunately, pretrained embedding cannot give better results in the experiments because of numerous noise data in training data.

5.3.4 The Proposed Neural Network Architecture for Myanmar NER

Myanmar language has complex characteristics and it is a kind of morphologically rich language. On the other hand, well-prepared language resources required for Myanmar NLP research have not been sufficient until now. As one of agglutinative languages, Myanmar has complex morphological structures; thus the

models can suffer from data deficiency. Further, for models in which words are considered as basic units to construct distributed representation, there may probably be problems for those rich morphological words. On the other hand, out-of-vocabulary words and word segmentation problem are important problems needed to deal with during natural language computation for Myanmar language. Word segmentation is necessary as pre-processing for most textual processing for the reason that regular white spaces are not inserted between words in written Myanmar texts. It means that segmentation results will affect the level of NER performance if words are treated as basic units for distributed representation.

In Myanmar language, syllable is the smallest linguistic unit that can hold information about word. For these reasons, syllable is considered as the basic input unit for label tagging in all NER experiments. Given a training sequence, syllables are taken as basic training unit and they were projected into a d-dimension space and initialized as dense vectors.

It is a truly model requiring no task-specific resources, feature engineering, or data pre-processing beyond pre-trained syllable embeddings on unlabeled corpora. Thus, our model can be easily applied to a wide range of sequence labeling tasks on different domains.

In this neural architecture, there are three main parts: character sequence representation layer, syllable sequence representation layer and inference layer. For each input syllable sequence, syllables are represented with syllable embeddings.

The character sequence layer can be used to automatically extract syllable level features by encoding the character sequence within the syllable. As the input of the character sequence layer, character embeddings represent characters. CNN is first used to encode character-level information of a syllable into its character-level representation. With CNN, it takes a sliding window to capture local features, and then uses a max-pooling for aggregated encoding of the character sequence.

Moreover, for learning character embedding from training data, bidirectional LSTM network as well as GRU was applied in comparison. However, according to the conducted experiments, when CNN is applied in character sequence representation layer, the performance is slightly better. Evaluation analysis will be discussed in the next chapter. Therefore, CNN is proposed to be suitable as character sequence representation layer.

Syllable representations are the concatenation of syllable embeddings and character sequence encoding hidden vector. Then the syllable sequence layer takes the syllable representations as input; feeds them into bidirectional LSTM and extracts the sentence level features from left to right and also from right to left, which are fed into inference layer to assign a label to each syllable. Although GRU was also applied, the performance is not been satisfied as expected. Among all experiments, bidirectional LSTM can give the best performance. Therefore, bidirectional LSTM becomes our choice to apply as syllable sequence representation layer for this neural NER for Myanmar language.

Instead of using the softmax output from inference layer, a sequential CRF is used to jointly decode labels for the whole sentence as CRF can take into account neighbouring tags. The architecture with softmax decoding layer cannot perform as good as the one with CRF decoding layer.

This proposed neural architecture (see Figure 5.7) is the best in which its model gives the best performance among all other architectures that have been carried out during experiments. This proposed neural architecture will be referred as CNN_BiLSTM_CRF for short in the following sections and chapters. Likewise, other network architectures will also be referred in short form in accordance with the applied network, for instance BiLSTM_BiLSTM_CRF for the architecture in which bidirectional LSTM is applied in both character representation layer and syllable sequence layer; and CRF is jointly added as decoding layer. And short-form like CNN_BiLSTM will be used to refer for those models which used softmax decoding layer.

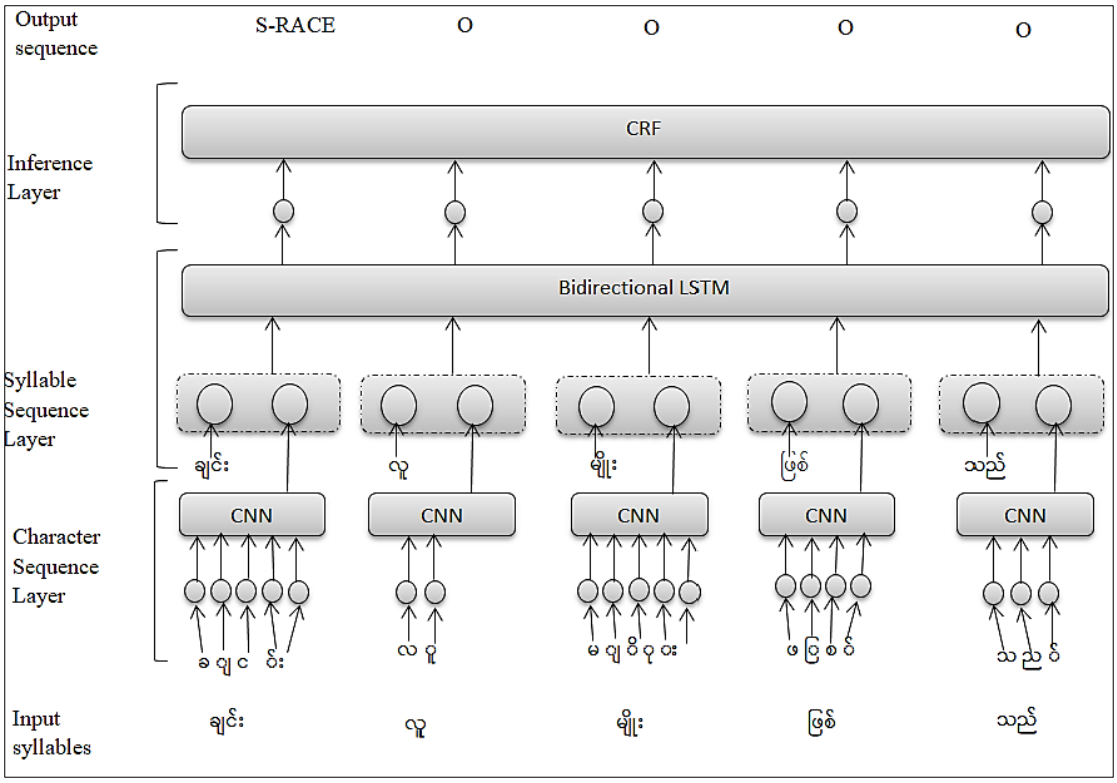


Figure 5.7 The Architecture of Syllable-based Neural Network for Myanmar NER

As to optimization, both the stochastic gradient descent algorithm (SGD) and Adam algorithm were tried. When with the SGD, the model was trained with initial learning rate of 0.015 and momentum 0.1. The learning decay rate was set as 0.05. For the Adam algorithm, the initial learning rate was set as 0.0015. Batch sizes was set as 10 for both optimizations.

During training, based on the performance on validation sets, early stopping was used so that it can stop as soon as the best performance happens. Finally, to prevent the models from depending too much on one representation, and training data or the other too strongly, dropout was also applied during training. It is found that dropout training is crucial for good generalization performance and for mitigating overfitting in the training process. A dropout rate of 0.5 was set for both embedding and output layers. The hidden dimension was set to 200 in the whole experiment.

To sum up, this proposed neural model training for Myanmar NER, given a Myanmar sentence, syllable is viewed as the basic training unit for label tagging. The input representation of character sequence within each input syllable is firstly learned with CNN, after that the learned representation plus syllable embedding are

concatenated as syllable representation. The syllable representation is taken as input and is put into bidirectional LSTM network for sentence level sequence learning. On top of the network, a CRF inference layer is added to determine the NE tag for each syllable with the maximum probability of the syllable.

5.3.5 Hyperparameters Tuning

It is commonly agreed that the selection of hyperparameters plays an important role in neural training. The authors of the paper [68] evaluated the importance of different network design choices and hyperparameters for linguistic sequence tagging tasks. Hyperparameters including learning rate, dropout rate, number of layers, hidden size, and so on can strongly affect the model performance. The experiments are conducted by tuning different hyperparameters setting. In this section, hyperparameters used in this neural training for Myanmar NER will be described. Table 5.3 summarizes the hyperparameters that give the best performance during training.

Table 5.3 Hyperparameters

Hyperparameters	**Value**
Learning rate	0.0015
Learning decay	0.05
Momentum	0.1
Hidden dimension	200
Character Hidden dimension	50
Average Batch size	10
LSTM layer	2
Dropout	0.5
Epochs	77

5.4 Implementation of Myanmar NER System

In order to make the proposed neural NER model available to be applied, it is deployed in the implementation of NER system. This Myanmar NER system takes sentences with Unicode encoding. The system automatically segments the input

sentences into syllables. The syllables are taken as input to model. The system recognizes names in input sentences by decoding model and then gives back the sentences with recognized NEs as output. During the decoding process, the Viterbi algorithm is used to search the label sequence with the highest probability.

5.5 Summary

This chapter presents the architecture of proposed neural named entity recognition for Myanmar language, detailed explanation of NE corpus building, data preparation, experiments of neural training, and different hyperparameters setting used in all experiments. During model training, CNN is firstly applied to encode character-level information of a token (syllable) into its character-level representation. Then character- and syllable-level representations are combined and feed them into bidirectional LSTM to model context information of each syllable. On top of bidirectional LSTM, a sequential CRF layer is added to jointly decode NE labels for the whole sentence. Myanmar NER system is implemented by applying the proposed neural NER model. The evaluation results of all conducted experiments will be discussed in Chapter 6.

CHAPTER 6
DISCUSSION OF EXPERIMENTAL RESULTS

In Chapter 5, the proposed neural model for Myanmar NER was described with the proposed neural architecture (CNN_BiLSTM_CRF). The development of the very first NE tagged corpus for Myanmar language was also presented. This corpus is built to solve resource deficiency.

In this chapter, all experiments conducted during this research will be discussed and evaluation of proposed syllable-based neural named entity recognition model for Myanmar language will be presented by comparing with other experiments. During experiments, a comprehensive comparison between the proposed neural model and the baseline statistical CRF model is also carried out.

Evaluating a machine learning model can be quite tricky so that cross validation, a very useful technique for accessing the performance of machine learning models, is also performed as evaluation technique. The cross validation technique helps in evaluating the quality of the model and can be used to compare the performance of different machine learning models on the same data. Therefore, 10-fold cross validation is also performed to compare the performance of the experiments. Moreover, another different test set is developed and tested on proposed neural model for Myanmar NER.

6.1 Data Partitioning for Experiments

Firstly, before 10-fold cross validation, experiments are conducted by splitting the data into three sets.

In the proposed Myanmar NE tagged corpus, there are totally 60,500 sentences and total number of 174,133 NE. These data are separated into three sets, 58,100 sentences for training (Train), 1,200 sentences for development (Dev) and another 1,200 sentences for testing (Test). Total number of NE in each set is stated as data statistics in Table 6.1. In train data set, total number of NE is about 96% of total number NEs in the corpus. In development set, it has about 2% of total NE. Similarly, the test set also contains 2% of total NEs.

Table 6.1 Data Statistics for Experiments

NE tags	Number of Named Entities		
	Train	Dev	Test
PNAME	34,266	622	517
LOC	60,916	1,365	1,211
ORG	19,084	375	281
RACE	5,359	200	161
TIME	28,386	556	530
NUM	19,508	363	433

6.2 Evaluation on Different Neural Architectures

As stated by the data partitions in section 6.1, experiments with different neural architectures are carried out. In this section, evaluation results on these experiments will be discussed.

In the following sections, CNN_BiLSTM_CRF represents a model using CNN to encode character sequence of input syllable, bidirectional LSTM for syllable sequence representation and CRF for inference layer, respectively. According to the three main parts (character sequence representation layer, syllable sequence representation layer and inference layer) of neural network architecture for Myanmar NER, all models will be referred in abbreviations such as BiLSTM_BiLSTM_CRF, CNN_GRU_CRF. And CNN_BiLSTM refers the model if the model makes use of softmax layer rather than CRF and so on.

For each input sequence to neural networks, not only characters but also syllables are treated as basic tokens and represented with embeddings. Therefore, there are character-based experiments and syllable-based experiments.

6.2.1 Character-based Neural Models

First of all, as part of experiments, characters are treated as basic input token to the network. However, these character-based models do not provide promising results compared to syllable-based models for Myanmar NER. Table 6.2 shows the experimental results from character-based models. Moreover, not surprisingly that it takes more time to train for character-based models than syllable-based models because input character sequence exhibits dependencies over long distance of hundreds of time steps.

As for character-level modeling, a variant of the neural network architecture first proposed by [7] and also reintroduced by Collobert et al., [16] for multiple NLP tasks was chosen to use. Figure 6.1 is the neural architecture that is being used in character-based modeling. The network takes the input sentence and discovers multiple levels of feature extraction from the inputs, with higher levels representing more abstract aspects of the inputs.

In this architecture, the first layer extracts the features for each Myanmar characters. The next layer extracts the features form a window of characters. The characters are fed into the network as indices that are used by a lookup operation to transform characters into their feature vectors. A fix-sized character dictionary is considered and the vector representations are stored in a character embedding matrix. The character embeddings are then taken as input to the bidirectional LSTM network. For each character in a sentence, a score is produced for every tag by applying several layers of neural network over the feature vectors produced by the lookup table layer.

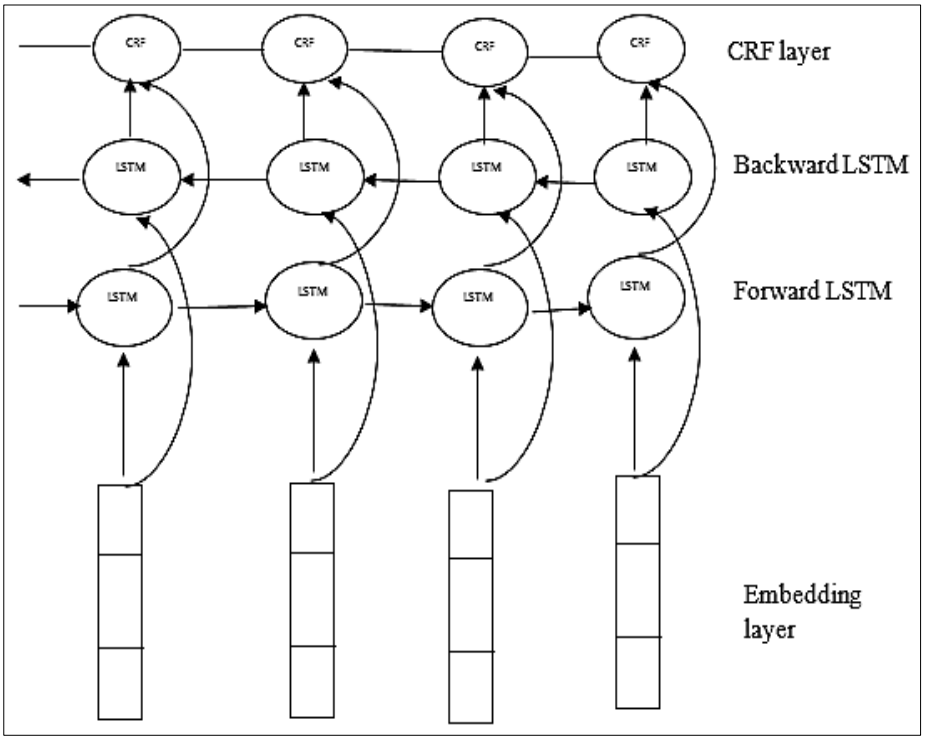


Figure 6.1: Neural Architecture for Character-based Modeling

As shown in Table 6.2, the F-score values resulted from character-based models are not satisfactory although they are reasonable. Model with CRF inference layer gives better performance.

Table 6.2 Experiment Results from Character-level Modeling

Models	Dev.			Test		
	Precision	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**
BiLSTM	82.32	83.04	82.68	83.71	84.4	84.05
BiLSTM_CRF	88.23	87.35	87.79	**89.02**	**88.29**	**88.65**

Although it is true that character-level neural models can offer a number of advantages, the vocabulary in a character-based model can be much smaller, as it only needs to represent a finite number of alphabet, and these types of models need no tokenization, freeing the experiment from one source of errors, character-level modeling is still challenging for some reasons. For example, the model must learn dependencies over long distances because character sequences are longer than word and syllable sequences and thus require significantly more steps of computation. Besides these facts, it is needed to consider how to improve the performance of the model. It is true that character features are important and can assist in modeling for such morphologically rich languages.

Myanmar language is rich in morphology and syllable-based language. Therefore, it is more preferable learning character-level features to character-level modeling for NER. For all of these reasons, this proposed NER modeling is a shift from character-level modeling to syllable-level modeling.

6.2.2 Syllable-based Neural Models

As for the syllable-based experiments, syllable-level tokens are considered as basic input units to the overall deep neural architecture for Myanmar NER which has three main layer units.

As for character sequence layer, several typical neural encoders such as CNN, GRU and LSTM are applied for character sequence information. For two variants of RNN (GRU and LSTM), the character sequence layer uses bidirectional to capture the left-to-right and right-to-left sequence information, and concatenates the final hidden states of the two RNNs as the encoder of the input character sequence. A sliding window is taken to capture local features, and then uses a max-pooling for aggregated encoding of the character sequence when CNN is applied in character sequence layer. It is the investigation of which network is more powerful in learning character representation.

Similar to the character sequence layer, both two variants of RNN and CNN are employed as the syllable sequence feature extractor. The input of the syllable sequence layer is a syllable representation, which may include syllable embeddings and character sequence representations. Both GRU and LSTM are applied as bidirectional to capture the left and right context information of each syllable. The hidden vectors for both directions on each syllable are concatenated to represent the corresponding syllable. When CNN is used, it utilizes the same sliding window as character CNN, while a nonlinear function is attached with the extracted features. Batch normalization and dropout are also applied to follow the features. Experiments were also performed in a form of stacked syllable sequence layer, building a deeper feature extractor.

The inference layer takes the extracted syllable sequence representations as features and assigns labels to the syllable sequence. Both softmax and CRF are utilized as the output layer. A linear layer firstly maps the input sequence representations label vocabulary size scores, which are used to either model the label probabilities of each syllable through simple softmax of calculate the label score of the whole sequence. Softmax maps the label scores into a probability space. In the training process, negative likelihood loss is used as loss function.

However, CRF captures label dependencies by adding transition scores between neighboring labels. During training, CRF trained with the sentence level maximum log-likelihood loss is applied. During decoding process, the Viterbi algorithm is used to search the label sequence with the highest probability. Experiments are carried out by testing different combinations of character representations and syllable sequence representations.

Moreover, experiments were carried out with different parameters and network settings. Achieving good or even state-of-the-art results with neural modeling is not straightforward, as it requires the selection and optimization of many hyperparameters, for instance tuning the number of recurrent units, choice of optimizer, the depth of network, the dropout rate and many more. Hyperparameters are mostly following the work [49] and almost keep the same in all these experiments.

Two popular optimization algorithms, the standard stochastic descent algorithm (SGD) with a decaying learning rate and Adam algorithm were utilized. Besides, the impact of Adagrad algorithm was also investigated.

As regularization, early stopping [10] by examining the performance on validation sets is used during training. A dropout rate was set for both embedding and output layers to prevent overfitting in the training process.

The following two tables list the F-score value of each neural model according to different neural architectures and various optimizers used. Table 6.3 shows the comparison of F-score values of different neural models that utilized SGD as optimizer. Similarly, the F-score values obtained when Adam was applied is presented in Table 6.4. When Adagrad was employed as optimizer, the resulted F-scores are not as good as the two others. Therefore, it is not listed in the tables.

Table 6.3 F-score Results Comparison among Different Models on Syllable-level Data (using SGD)

Models	Dev.			Test		
	Precision	Recall	F-score	Precision	Recall	F-score
BiLSTM_BiLSTM	85.19	84.22	84.7	91.62	91.09	91.36
BiLSTM_BiLSTM_CRF	85.80	86.03	85.92	92.12	92.53	92.32
CNN_BiLSTM	85.08	83.81	84.44	91.24	89.75	90.49
CNN_BiLSTM_CRF	**90.44**	**90.34**	**90.39**	**93.15**	**93.68**	**93.41**
BiGRU_BiLSTM	83.34	82.26	82.8	89.02	88.29	88.65
BiGRU_BiLSTM_CRF	84.53	84.99	84.76	91.14	90.97	91.05
CNN_BiGRU	83.6	82.81	83.2	89.8	88.51	89.15
CNN_BiGRU_CRF	87.28	87.38	87.33	91.19	92.18	91.68

Table 6.4 F-score Results Comparison among Different Models on Syllable-level Data (using Adam)

Models	Dev.			Test		
	Precision	Recall	F-score	Precision	Recall	F-score
BiLSTM_BiLSTM	88.09	87	87.55	93.06	92.88	92.97
BiLSTM_BiLSTM_CRF	**90.54**	**89.94**	**90.24**	**94.79**	**94.57**	**94.68**
CNN_BiLSTM	84.53	84.99	84.76	91.14	90.97	91.05
CNN_BiLSTM_CRF	**91.18**	**90.40**	**90.79**	**95.04**	**94.89**	**94.97**
BiGRU_BiLSTM	85.03	82.49	83.74	90.71	89.47	90.09
BiGRU_BiLSTM_CRF	85.48	84.62	85.05	90.72	90.52	90.62
CNN_BiGRU	84.92	84.36	84.64	90.84	90.49	90.66
CNN_BiGRU_CRF	87.82	83.93	85.83	93.47	90.87	92.15

If compared the results in Table 6.3 with those in Table 6.4, Adam works generally better than SGD except in some case. However, the performance of using

SGD optimization algorithm is slightly worse than using Adam, which you can see from Table 6.3 and Table 6.4.

As shown in Table 6.3 and 6.4, GRU based models consistently underperform. During training, the performance of GRU-based architecture constantly dropping as number of epochs grows. The results of GRU-based architecture stated in Tables are gained at initial or early epochs.

Likewise, models with CNN for syllable sequence representation also perform less well than models with bidirectional LSTM for syllable sequence layer, showing the advantages of bidirectional LSTM on capture global features. Therefore, the results from the models that CNN is applied for syllable sequence layer are omitted in tables.

Character information can improve model performance significantly, while both LSTM and CNN give similar improvement but CNN perform a little bit better than bidirectional LSTM. Due to the support of parallel decoding, softmax gives result approaching to CRF and works well. However, models with CRF inference layer are much more efficient than softmax.

According to the experimental results, BiLSTM_BiLSTM_CRF model and CNN_BiLSTM_CRF model achieved the comparable results and these two models provide better performance among others. In other words, CNN is suitable for character sequence layer, bidirectional LSTM is beneficial for syllable sequence layer and work together by adding CRF layer is powerful for inference layer.

To sum up, CNN_BiLSTM_CRF model with Adam optimizer achieved the best F-score among all the models during experiment. Therefore, this syllable-based neural framework architecture is suitable for Myanmar NER and offers the best performance.

6.3 Baseline Statistical CRF

Before neural training, NER for Myanmar language is also addressed with traditional statistical CRF approach. Figure 6.2 shows the work flow of statistical NER. The bold arrow shows the training time while the other represents the testing time.

In order to perform syllable-based CRF trainings, an open source toolkit [42] developed by Taku Kudo for linear chain CRF is used. During experiments, various

parameters were tuned with different features. Firstly, only the tokens and their neighbouring contents were used as features and the window size was set as 5 and the best F-score of 88.05% is obtained when the cut-off threshold parameter and hyper-parameter c were set 3 and 2.5, respectively.

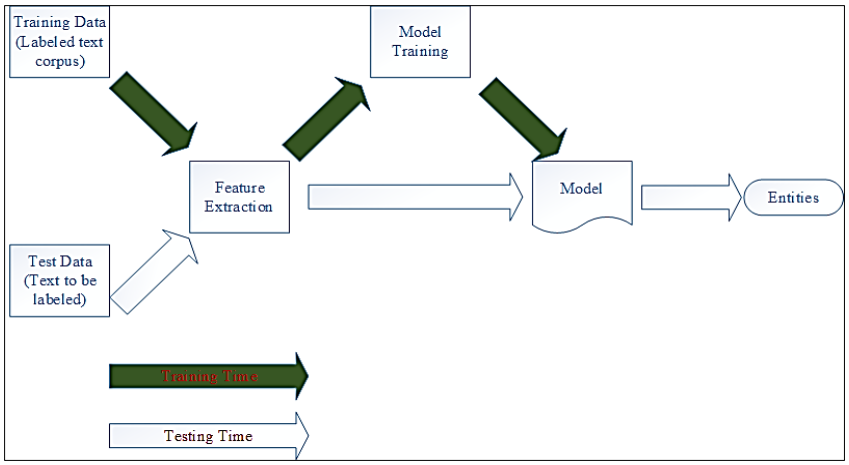


Figure 6.2: Work Flow of CRF NER

6.3.1 Experiment with External Features

Most NER systems of other languages such as English use the additional features like part-of-speech (POS) tags, shallow parsing, gazetteers, etc.

For this work, a small-sized named dictionary and clue words list are added as additional features into the CRF training. Myanmar person names are usually followed after words such as ဦး ၊ ကို ၊ မောင် for male and မ ၊ ဒေါ် for female. Likewise, some of the person names are appeared along with salutation words such as ဒေါက်တာ and ပါမောက္ခ. Such possible clue words for person names, location names and organization names are prepared first and use these as external features for the NER task. About 30 words are prepared as clue words list. Besides the clue words list, small size name dictionary which has about 300 names including person names, names of towns, distinct, villages, and countries are prepared and are also applied as external feature.

6.3.1.1 Preparation of Data with External Features

In order to process CRF training with external features, trainig data is prepared into trainable format as shown in Figure 6.3. According to the data for CoNLL shared

task, there may be as many columns as wishes, however the number of columns must be fixed through all tokens. Futhermore, there are some kinds of semantics among the columns. In this work, the first column is token (syllable), the second column is the value of named dictionary feature, the third column is the value of cule word list feature and the last represents the annotated named label.

To convert trainable data format, small-size named dictionary list and clue word list are fristly converted into tree structures from lists in order to be easy to find the maximum length from these. After that, the syllable is checked whether it is in the named dictionary tree or clue words tree. If the syllable is appeared in the node of the tree, searching is continued to get the maximum length of names of clue words from the dictionary. The x-x is defined as the first x is the length of names or clue words found in dictonary and the senond x denotes count of the syllables. 2-0 means the syllable is appeared in dictionary and there is another ayllable next to it.

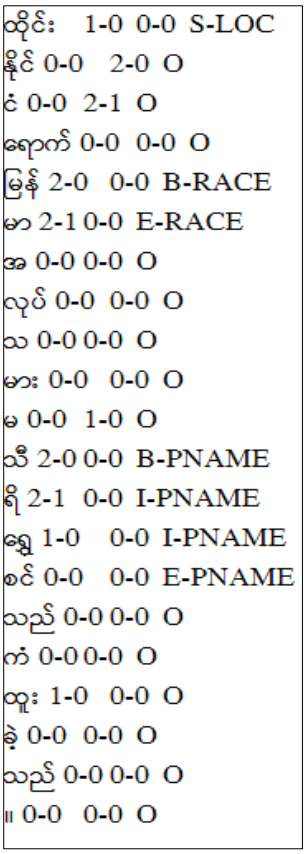


Figure 6.3 Example Data Format with External Features

After preparing this, the window size is set as 5 and the best F-score 90.45% is achieved when the parameter setting of cut-off threshold and hyperparameter c are set

as 3 and 2.5, respectively. The additional features help the F-score increase around 2.4 % (become 90.45 %) compared to the previous F-score of 88.05 %. It shows that CRF works the best when features are carefully selected and it totally relies on feature engineering. This result is even better character-based neural model.

6.4 Performance Comparison between Neural Models and Baseline CRF Model

When compared with proposed neural model, it is obvious that neural model provides the superior result. The difference of performance results on neural models and baseline model can be seen in Table 6.5. It is not surprisingly that neural models outperform baseline statistical model because deep neural networks are able to extract features automatically without human intervention.

Table 6.5 The F-score Results on Different Models

Models	F-score
Baseline CRF	90.45
CNN_BiLSTM_CRF (SGD)	93.41
BiLSTM_BiLSTM_CRF (Adam)	94.68
CNN_BiLSTM_CRF(Adam)	**94.97**

In Figure 6.4, the F-score results from different models are also compared.

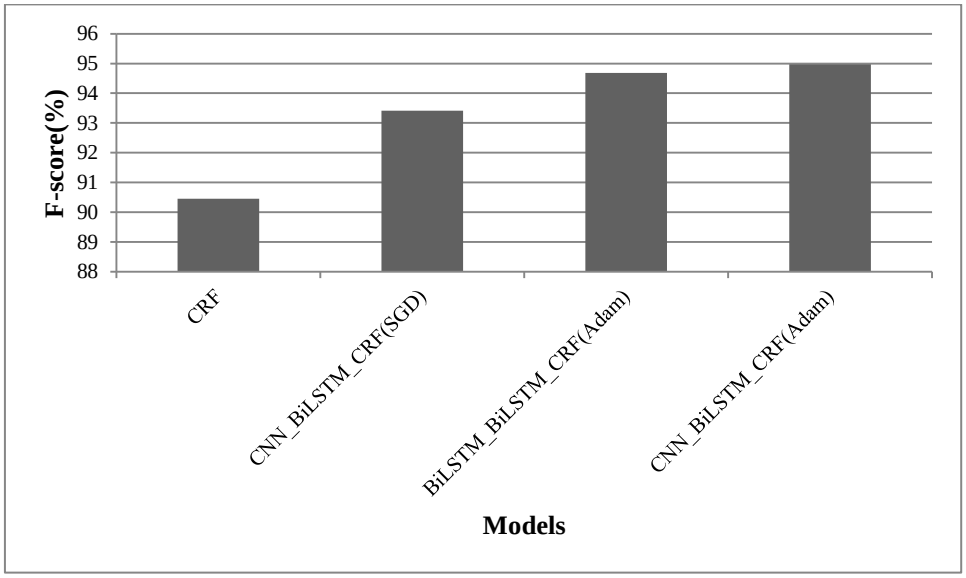


Figure 6.4 F-score Results Comparison of Different Models

6.5 10-fold Cross Validation

Cross validation is a model validation technique and is primarily used in applied machine learning for accessing how the results of a model will generalize to an independent data set. It is a popular method because it is simple to understand and because it generally results in a less biased or less optimistic estimate of the model skill which means that it helps to avoid overfitting and underfitting.

As 10-fold cross validation, firstly shuffle the dataset randomly and split the dataset into 10 equal size groups. Of the 10 groups, a single group is retained as the validation data for testing the model, and the remaining 9 groups are used as training data. The cross validation process is then repeated 10 times (the folds). The results from the 10-fold are then be averaged to produce a single estimation.

In Table 6.6 and Table 6.7, the results from 10-fold cross validation are presented.

Table 6.6 The Performance Result of 10-fold Cross Validation (BiLSTM_BiLSTM_CRF)

N-fold	Dev.			Test		
	Precision	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**
1-fold	87.79	86.83	87.31	94.34	94.19	94.27
2-fold	89.48	88.01	88.74	92.57	92.28	92.42
3-fold	88.63	87.84	88.23	94.74	94.22	94.48
4-fold	85.08	83.81	84.44	89.92	87.85	88.87
5-fold	83.60	82.81	83.20	90.44	88.88	89.65
6-fold	87.28	87.38	87.33	90.09	88.23	89.15
7-fold	89.00	88.44	88.72	91.47	90.97	91.22
8-fold	89.14	89.42	89.28	94.72	94.79	94.71
9-fold	88.09	87.01	87.55	89.69	87.96	88.81
10-fold	89.49	88.61	89.05	90.19	88.59	89.38
Average	**87.76**	**87.02**	**87.39**	**91.82**	**90.80**	**91.30**

Table 6.7 The Performance Result of 10-fold Cross Validation (CNN_BiLSTM_CRF)

N-fold	Dev.			Test		
	Precision	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**
1-fold	90.35	89.65	90.00	93.63	92.88	93.25
2-fold	86.43	86.29	86.36	94.39	93.39	93.89
3-fold	88.98	88.90	88.94	94.47	94.25	94.36
4-fold	90.80	88.77	89.77	91.24	89.75	90.49
5-fold	90.73	90.56	90.65	89.80	88.51	89.15
6-fold	91.39	89.72	90.55	93.15	93.68	93.41
7-fold	90.68	88.65	89.65	94.23	94.38	94.31
8-fold	89.44	88.93	89.19	94.10	94.67	94.38
9-fold	90.28	88.28	89.27	93.06	92.88	92.97
10-fold	91.38	90.67	91.02	94.98	94.86	94.92
Average	**90.05**	**89.04**	**89.54**	**93.31**	**92.93**	**93.11**

It has been revealed that the proposed model is powerful by looking at the results from 10-fold cross validation.

6.6 Evaluation on Different Test Sets

The model is also tested on another different test set which are totally different written style form training data. As mentioned in previous chapter, data in NE corpus are collected from news data. And therefore test data are also from new data. This means that the test data is a kind of open data but close domain. For this reason, different test sets are also prepared in order to test the performance of proposed neural model for generalization.

Another two test sets, each with 1,200 sentences are used to as open test sets. Test set 1 contains sentences written in conversation style while another Test set 2 has sentences taken from UCSY parallel corpus. In UCSY parallel corpus, besides from news data, most sentences are from School Text book and some sentences are from novels and also some are from travel blog. Therefore, sentences in UCSY corpus has the writing style of both formal and informal style. Sentences from the Test set 1 which is a conversation test set are totally different from training sentences. It can be said that the test data from these two test sets are kind of open data from open domain.

In Test set 1, there are 2,156 NEs in total and 2,449 NEs in UCSY Test set 2. Over 45% of total NEs in Test set 1 is the location entity. However, the organization and the number entities are just 6% each. In the same situation, the most frequent

found NE in Test set 2 is also the location entity; nearly 50% of all NEs. The least occurrence of NE in Test set 2 is the organization entity; it has only 5% of all NEs. The distribution of data in these two test sets are shown in Table 6.8 and Table 6.9.

Table 6.8 Data Statistic of Test Set 1

NE tags	Number of Named Entities	Occurrence of each NE (%)
PNAME	420	19
LOC	1,001	46
ORG	136	6
RACE	184	9
TIME	288	13
NUM	127	6
Total NEs	2156	

Table 6.9 Data Statistic of Test Set 2

NE tags	Number of Named Entities	Occurrence of each NE (%)
PNAME	501	20
LOC	1,205	49
ORG	112	5
RACE	195	8
TIME	301	12
NUM	135	6
Total NEs	2,449	

When compared the results in Table 6.10, it can be seen that F-score value drops significantly when tested on Test set 1 which is totally different from train data and train domain (type of open data from open domain). However, F-score value resulted from testing with Test set 2 is approaching to satisfactory result.

Although the training data contains only news data, it can be said that this model can inference and work on different domains. Even though the F-score of daily conversation test is not as good as news test set and Test set 2, it is a promising result.

Table 6.10 The F-score Results of Different Test Sets

Test data	F-score
Test set 1	85.21
Test set 2	90.01
News Test set	**94.97**

The performance comparison among different test sets is provided in Figure 6.5. By comparison, news test set scores the highest F-score value among others.

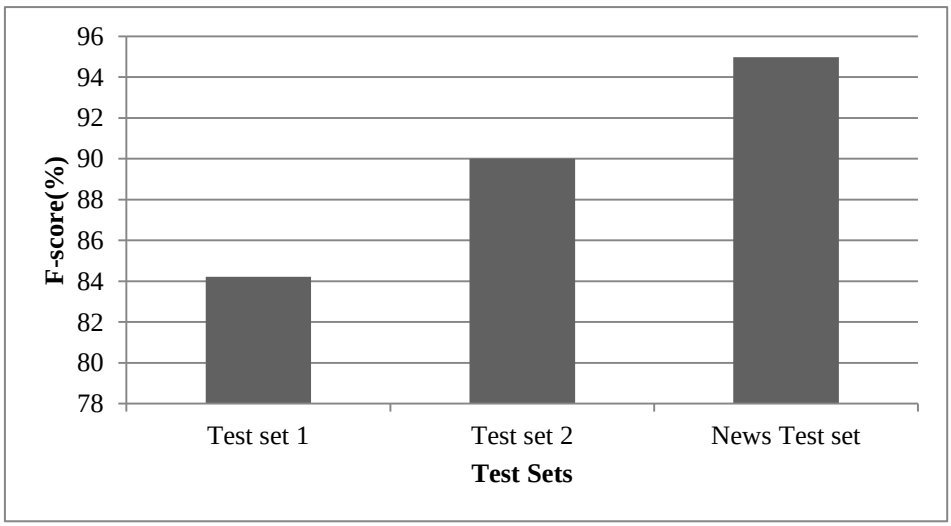


Figure 6.5 The Performance Comparison among Different Test Sets

6.7 Analysis on Evaluation

During the experiments, the influence factors to neural model accuracy such as pretrained embeddings, tag scheme, character sequence representations, syllable sequence representations, inference algorithm, different optimizers, and hyperparameter values etc. had been investigated.

The detailed analysis and comparison reveal that character information provides a significant improvement on accuracy. Moreover, CRF can improve the model accuracy.

For the reason that syllables as inputs hold more information than individual characters as inputs, it is reasonable that syllable-based models perform better than the character-based models. On syllable level data, extracting character features with

CNN from the data as additional information inputs, better outcomes are produced compared to not using character features or using bidirectional LSTM to extract character features (See Table 6.4 and Table 6.5, the best performance is highlighted). Learning char features with bidirectional LSTM is not as good as learning with CNN in our experiments.

The proposed model can also provide satisfactory result to another different test set. And thus, it can be said that this neural model can be applied in different domains. The pre-trained word embedding from word2vec also did not produce better results. This may be due to the noisy nature of data in training data. Therefore, proper data cleaning is necessary before training.

By comparison, the performance of syllable-based CRF model with additional features is approaching to the outcomes of neural models. Meanwhile, there are some ambiguous errors. This may be due to two reasons. Firstly, the size of training data used for neural network model training is not as big as other benchmark data. Normally large amount of data can help neural network learn better and produce superior outcomes. Secondly, due to time and data limit, the hyper-parameters used in the experiment may not be the best. This needs modelling experiences and also large amount of trial and error experiments to decide.

6.8 Error Analysis

Meanwhile, there are still errors in this NER neural models. The common errors are ambiguous errors. The errors can be roughly classified into three different types: NE ambiguous error, NE boundary conflict error, and unknown word error.

NE types ambiguous errors are commonly occurred in news test set. In the sentence of following Figure 6.6, the name "လီဗာပူးလ်" should be the organization name but wrongly tagged as person name. This error may probably because of the sentence written style.

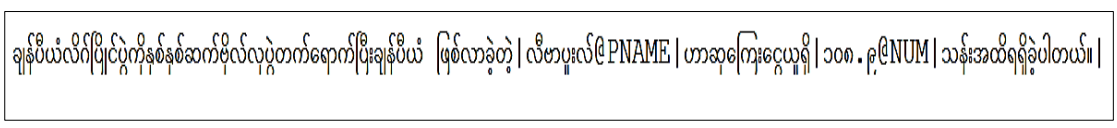


Figure 6.6 Example of NE Types Ambiguous Error

Another error, NE boundary conflict error commonly occurred in conversation style sentences. Such kinds of errors are shown in Figure 6.7. In this case, "ဟေး" is not part of NE. This type of error especially happens when person names are written directly before or after the interjection words.

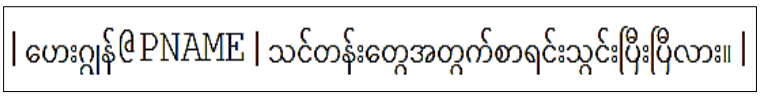


Figure 6.7 Example of NE Boundary Conflict Error

Some names are not recognized in some cases. Such kinds of errors are found when NE are the same as common words. Examples are shown in Figure 6.8. In this sentence the location name "အော်စတင်" is neither recognized as NE nor wrongly tagged with other NE tags.

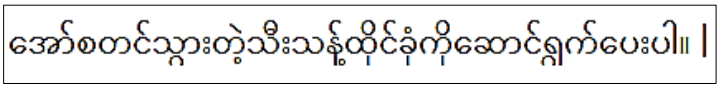


Figure 6.8 Example of Unknown NEs Error

6.9 Summary

According to the experimental results, it has been revealed that bidirectional LSTM based deep neural models are powerful for Myanmar NER. BiLSTM_BiLSTM_CRF model produces the similar accuracy as the CNN_BiLSTM_CRF model but it can be seen that CNN performs slightly better than bidirectional LSTM, in extracting character features in character sequence layer, which is out of the exception.

Bidirectional LSTM network is powerful in extracting sentence level features from syllable representations in syllable sequence layer.

From the experiments, it can be seen that neural network performs much better on syllable level data than on character level data. Although the model trained with softmax can also produce promising results, CRF inference layer is more powerful for Myanmar NER. By comparison, Adam performs slightly better than SGD.

The baseline statistical CRF model with additional features can also five the satisfactory results. However, it totally depends on the choice of features.

The proposed neural NER model is also validated by performing 10-fold cross validation. The result of 10-fold cross validation on the proposed model is satisfying. Additionally, this proposed model is tested on two different test sets which are open data from open domain, and the results revealed that the proposed model can give the promising results.

CHAPTER 7
CONCLUSION AND FUTURE WORK

This chapter is the description of summarization of the research work, including the advantages and limitations of the proposed dissertation.

The main contribution of the research work is the very first evaluation of deep neural network architecture on Myanmar Named Entity Recognition. As part of this research, NE tagged corpus for Myanmar language is manually developed and proposed. Syllables are taken as input tokens in neural NER modeling rather than characters or words.

7.1 Thesis Summary

In this work, a manually annotated NE tagged corpus for Myanmar language has been developed with the intension to develop resources for Myanmar NLP and to provide resource for further research. Myanmar is being regarded as an under-resourced language as there is no currently freely and commercially available corpora for Myanmar language, including NE corpus. As a consequence, textual processing for Myanmar language is still under-developed compared to other languages. For this reason, NE tagged corpus is manually annotated and constructed by collection news data from official online news websites. There are totally over 60K sentences and over 170K NEs in our NE corpus.

Additionally, the effectiveness of neural network on Myanmar NER has been explored and a systematic comparison is conducted between neural approaches and traditional CRF approaches on the proposed manually annotated NE tagged corpus.

Myanmar language is written without putting regular spaces between words and therefore word segmentation is necessary as preprocessing step in language processing. Likewise, word segmentation has an impact on NER performance and wrong segmentation can lead to errors in recognizing NE. In Myanmar language, syllables are the basic units that can carry information about words and its structure is not quite difficult to segment. Syllables are considered basic input units and thus all experiments process on syllable-level data.

The baseline statistical model is built by making use of CRF. Firstly, CRF model is trained by only considering content feature of constant window size. After

that, external features are added to the experiment. The performance improves when external features are applied. This shows that statistical CRF works well when feature engineering is carefully prepared and depends on it very much. However, the performance result is not as good as neural model.

Experimental results from this research have revealed that the performance of neural networks on Myanmar NER is quite promising, although neural models did not use any handcrafted features or additional resources. From the experiments, it can be seen that neural network models did much better while in the experiments of using CRF models, only by adding additional name list features and clue word list, produced the similar accuracy as the syllable-based neural models.

For neural training, various deep neural architectures are being tested for Myanmar NER. Among all experiments, the neural model that represents syllables as a combination of a syllable embedding and a convolution over the characters of the syllable, following this with bidirectional LSTM layer over the syllable representations of a sentence, and predicting the final label tags using CRF layer give the best performance. In the previous chapters, this model is referred as CNN_BiLSTM_CRF model. 10-fold cross validation is also performed and the result has proved that the proposed model outperforms others. The model is tested on different open test sets as well; and the result is also acceptable.

Anyhow, this exploration of using neural networks is the first work to apply neural networks on Myanmar NER. It showed us that deep neural network model obtained from processing on syllable level data jointly with CNN to extract character feature of a syllable and passing this representation through another sentence-level bidirectional LSTM and adding CRF layer above can facilitate Myanmar NER.

In order to make Myanmar NER system that can recognize names in Myanmar written text, the proposed neural model is applied.

7.2 Advantages and Limitations of the Proposed System

This proposed neural architecture for Myanmar NER provides better performance without any additional features. It performs better than traditional statistical models as it has been revealed in previous chapter.

Although the training data contains mostly news data, it is able to recognize names in daily conversation style sentences. It can also recognize person names that

are not preceded by title words and location names without any clue words surrounding around them. It can be said that this neural model has the ability to recognize and classify into predefined NE types correctly except for some minimal errors. Many OOV names can also be recognized by this model. For TIME type, there are many forms: number style or test style. Different TIME types are correctly recognized. Anyhow, this neural model for Myanmar NER can be intergraded into the development of Myanmar NER tool, IR system, entity linking, etc.

Moreover, development of manually annotated NE tagged corpus can contribute in future research on Myanmar NER. It can assist in development of Myanmar NLP research work. Currently, there are six defined NE tags in this NE corpus. From this annotated NE tagged corpus, names can be extracted and name lists can also be constructed if required.

In the meantime, there are still weaknesses and limitations in this neural NER for Myanmar language. As limitation, it can only process on sentences that are written in Unicode (e.g., Myanmar 3 and Padauk) encodings. Names written in combination of English characters and Myanmar Character (e.g Rဓာနည်) cannot be recognized because training data are not prepared for such case. Names in sentences that are written in conversation styles are not effectively recognized as well. There are some ambiguous errors in this model. This error will be taken into account for improvement of Myanmar NER performance.

7.3 Future Work

A recent trend in Deep Learning is emphasizing on Attention mechanisms. Attention mechanism has gained popularity recently in image, speech and NLP fields [13] [14]. For NER task, the paper [59] introduced the attention mechanism to enhance their model performances. Likewise, an attentive neural network for the task of NER in Vietnamese was proposed in [59]. In future, attention-based neural experiments will be conducted with the intention of improving performance of Myanmar NER.

Data in the manually annotated NE tagged corpus is not as much as other languages so that more and more data needs to be added as much as possible. In future, NE corpus will be constructed with many more defined NE tags and even in hierarchical structure.

Although our NE tagged corpus is not too big, neural network models produce better performance than CRF models for Myanmar NER, we still believe with more data and more experiments, advanced neural networks can learn better so as to produce better results. Besides, there is an intention to make it domain independent.

With more data and more experiments, better results will be reported in the future and deep neural networks will be kept exploring on Myanmar NER and also on other Myanmar NLP work, e.g., POS tagging and word segmentation, too. Moreover, in the future, Myanmar NER system is intended to build by end-to-end learning approach.

7.4 Conclusion

As the conclusion of this dissertation to be stated, there are three main contributions in this research. All these proposed contributions help this dissertation to meet its objectives which are described in Chapter 1.

To make available NE tagged corpus for future NER research and to address the resource deficiency issue in Myanmar NER, the very first manually annotated NE tagged corpus is created and proposed.

Since it processes on syllable-level tokens, it can avoid the need of word segmentation process.

In order to provide a good quality NER model for Myanmar language, neural NER model is trained and proposed. It does not need any additional features, human knowledge or domain experts in neural NER modeling so that it reduces the need of expensive additional feature engineering process. Moreover, deep neural networks make use of deep layers to induce NEs automatically. From the conducted experiments, it can be believed that deep neural learning is suitable for Myanmar NER and deep neural network can be applied in other area of Myanmar NLP research. This proposed neural model can be applied in other Myanmar NLP system.

Myanmar NER system is also developed by applying the proposed neural NER model in order to provide NER tool for Myanmar language. It is hoped that the result from this Myanmar NER system will be useful to other NLP research works for Myanmar language.

AUTHOR'S PUBLICATIONS

p[1] H.M. Mo, K.T. Nwet, and K.M. Soe, "CRF-Based Named Entity Recognition for Myanmar Language", 10th International Conference on Genetic and Evolutionary Computing (ICGEC 2016), pp. 204-211, 21 October 2016.

p[2] H.M. Mo, K.T Nwet, and K.M. Soe, "Exploring Features for Myanmar Named Entity Recognition", 15th International Conference on Computer Applications (ICCA), Yangon, Myanmar, pp. 429.433, February, 2017.

p[3] H.M. Mo, and K.M. Soe, "Syllable-Based Neural Named Entity Recognition For Myanmar Language", International Journal on Natural Language Computing (IJNLC), Vol.8, No.1, February 2019.

BIBLIOGRAPHY

[1] R. Alfred, L. C. Leong, C. K. On, P. Anthony, T. S. Fun, M. N. B. Razali and M. H. A. Hijazi, “A Rule-Based Named-Entity Recognition for Malay Articles,” ADMA 2013, Part 1, LNAI 8346, Springer-Verlag Berlin Heidelberg 2013, pp.288-299.

[2] M. Ali, G. Tan, and A. Hussain. "Bidirectional Recurrent Neural Network Approach for Arabic Named Entity Recognition," Future Internet 10, no. 12 (2018): 123.

[3] K.S. Bajwa and A. Kaur, “Hybrid Approach for Named Entity Recognition,” International Journal of Computer Applications, vol.118, no.1, pp-0975-8887, May. 2015.

[4] P. Basile, G. Semeraro, and P. Cassotti, "Bi-directional LSTM-CNNs-CRF for Italian Sequence Labeling," CLiC-it 2017 11-12 December 2017, Rome (2017): 18.

[5] I. E. Bazi, and N. Laachfoubi, "A Comparative Study of Named Entity Recognition for Arabic using Ensemble Learning Approaches," In 2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA), pp. 1-6. IEEE, 2015.

[6] Y. Benajiba, M. Diab and P. Rosso, “Arabic Named Entity Recognition: An SVM-based Approach,” In Proceedings of 2008 Arab International Conference on Information Technology (ACIT) 2008 (pp. 16-18). Amman, Jordan: Association of Arab Universities.

[7] Y. Bengio, R. Ducharme R, P. Vincent, and C. Jauvin, “A Neural Probabilistic Language Model,” Journal of Machine Learning Research, 2003;3(Feb):1137-55.

[8] D. Bonadiman, A. Severyn, and A. Moschitti, "Deep Neural Networks for Named Entity Recognition in Italian," CLiC it 51 (2015).

[9] A. Borthwick, J. Sterling, E. Agichtein, and R. Grishman, "Exploiting Diverse Knowledge Sources via Maximum Entropy in Named Entity Recognition," In Sixth Workshop on Very Large Corpora. 1998.

[10] R. Caruana, S. Lawrence, CL. Giles, “Overfitting in Neural Nets: Backpropagation, Conjugate Gradient, and Early Stopping,” In Advances in

Neural Information Processing ySstems 2001 (pp. 402-408).

[11] H.L. Chieu, and H.T. Ng, “Named Entity Recognition with a Maximum Entropy Approach,” In Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4 (pp. 160-163). Association for Computational Linguistics, 2003, May.

[12] P.C. J. Chiu, and E. Nichols, "Named Entity Recognition with Bidirectional LSTM-CNNs," Transactions of the Association for Computational Linguistics, vol. 4 (2016): pp.357-370, 2016.

[13] K. Cho, B. Van Merriënboer, D. Bahdanau, Y. Bengio, “On the Properties of Neural Machine Translation: Encoder-decoder Approaches,” arXiv preprint arXiv:1409.1259. 2014 Sep 3.

[14] J.K. Chorowski, D. Bahdanau, D. Serdyuk D, K. Cho, Y. Bengio Y, “Attention-based Models for Speech Recognition,” In Advances in Neural Information Processing Systems 2015. (pp. 577-585).

[15] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling,” arXiv preprint arXiv:1412.3555, 2014.

[16] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and Pavel Kuksa, "Natural Language Processing (almost) from Scratch," Journal of machine learning research 12, no. Aug (2011): 2493-2537.

[17] A. Das and U. Garain. "CRF-based Named Entity Recognition@ ICON 2013," [Online] Available: https://arxiv.org/abs/1409.8008.

[18] F. Dernoncourt, J. Y. Lee, and P. Szolovits, “NeuroNER: an easy-to-use program for named-entity recognition based on neural networks,” [Online] Available: https://arxiv.org/abs/1705.05487, May, 2017.

[19] C. Ding, Y. K. Thu, M. Utiyama, and E. Sumita, "Word Segmentation for Burmese (Myanmar)," ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), vol.15, no. 4 (2016): 22, May 2016.

[20] A. Ekbal, and S. Bandyopadhyay, "A Hidden Markov Model based Named Entity Recognition System: Bengali and Hindi as Case Studies," In International Conference on Pattern Recognition and Machine Intelligence, pp. 545-552. Springer, Berlin, Heidelberg, 2007.

[21] A. Ekbal, and S. Bandyopadhyay, "Development of Bengali Named Entity Tagged Corpus and its Use in NER Systems," In Proceedings of the 6th Workshop on Asian Language Resources, 2008.

[22] A. Ekbal, and S. Bandyopadhyay, "Named Entity Recognition using Support Vector Machine: A language independent approach," International Journal of Electrical, Computer, and Systems Engineering vol 4, no. 2 (2010): pp-155-170, 2010.

[23] A. Ekbal, R. Haque, and 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.

[24] R. Florian, A. Ittycheriah, H. Jing, and T. Zhang, "Named Entity Recognition through Classifier Combination," In Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4, pp. 168-171. Association for Computational Linguistics, 2003.

[25] V. Gayen, and K. Sarkar, "An HMM based Named Entity Recognition System for Indian Languages: the JU system at ICON 2013," [Online] Available: https://arxiv.org/abs/1405.7397.

[26] J. M. Giorgi, and G. D. Bader, "Transfer Learning for Biomedical Named Entity Recognition with Neural Networks," Bioinformatics, vol. 34, no. 23 (2018): pp-4087-4094, June 2018.

[27] A. Z. Gregoric, Y. Bachrach, and 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. 2018.

[28] W. Gunawan, D. Suhartono, F. Purnomo, and A. Ongko, "Named-Entity Recognition for Indonesian Language using Bidirectional LSTM-CNNs," 3rd International Conference on Computer Science and Computational Intelligence 2018, Procedia Computer Science 135 (2018): 425-432.

[29] Y. Hahm, J. Park J, K. Lim, Y. Kim, D. Hwang, and K. S. Choi, "Named Entity Corpus Construction using Wikipedia and DBpedia Ontology," InLREC 2014 May (pp. 2565-2569).

[30] T. H. Hlaing, "Manually Constructed Context-free Grammar for Myanmar

Syllable Structure," In Proceedings of the Student Research Workshop at the 13th Conference of the European Chapter of the Association for Computational Linguistics, pp. 32-37. Association for Computational Linguistics, 2012.

[31] T. H. Hlaing, and Y. Mikami, "Automatic Syllable Segmentation of Myanmar Texts using Finite State Transducer," *ICTer*, vol.6, no. 2 (2014).

[32] S. Hochreiter, and J. Schmidhuber, "Long short-term memory," Neural computation 9, no. 8 (1997): 1735-1780.

[33] M Hosken, M. Tuntunlwin, "Representing Myanmar in Unicode Details and Examples," 2007.

[34] 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.

[35] Z. Huang, W. Xu, and K.Yu. "Bidirectional LSTM-CRF Models for Sequence Tagging," arXiv preprint arXiv:1508.01991. 2015 Aug 9.

[36] Q. Hung Ngo, D. Dien, and W. Winiwarter, "Building English-Vietnamese Named Entity Corpus with Aligned Bilingual News Articles," January, 2014.

[37] J. P. Jayan, R. R. Rajeev, and E. Sherly, "A Hybrid Statistical Approach for Named Entity Recognition for Malayalam Language," In Proceedings of the 11th Workshop on Asian Language Resources, pp. 58-63. 2013.

[38] V. John, "A Survey of Neural Network Techniques for Feature Extraction from Text," arXiv preprint arXiv:1704.08531. 2017 Apr 27.

[39] R. Klinger, and K. Tomanek, "Classical Probabilistic Models and Conditional Random Fields. TU", Algorithm Engineering, 2007.

[40] J. Kravalov´a and Z. Z˘abokrtsky˘, "Czech Named Entity Corpus and SVM-based Recognizer," In Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration, pp. 194-201. Association for Computational Linguistics, 2009.

[41] A. Krogh, and J. A. Hertz, "A Simple Weight Decay Can Improve Generalization. In Advances in neural information processing systems (pp. 950-957), 1992.

[42] T. Kudo. CRF++: Yet another CRF toolkit (2005). Available under LGPL from the following URL: http://crfpp. sourceforge. net. 2015 Jun.

[43] Y. Kyaw Thu, “Syllable segmentation tool for Myanmar language (Burmese) by Ye.” https://github.com/ye-kyaw-thu/sylbreak.

[44] J. Lafferty, A. McCallum, and F.C. Pereira, “Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data,” 2001.

[45] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, “Neural Architectures for Named Entity Recognition,” In Proceedings of NAACL-HLT 2016, pp-260-270, Association for Computational Linguistics, June, 2016.

[46] Y. LeCun, B. Yoshua, and H. Geoffrey, "Deep Learning," Nature 521, no. 7553 (2015): 436.

[47] J. Y. Lee, F. Dernoncourt, and P. Szolovits, “Transfer Learning for Named-Entity Recognition with Neural Networks,”, In proceeding of the Eleventh International Conference on Learning Resources and Evaluation (LREC 2018), May, 2018.

[48] L. Li, Lishuang, R. Guo, S. Liu, P. Zhang, T. Zheng, D. Huang, and H. Zhou. "Combining Machine Learning with Dictionary lookup for Chemical Compound and Drug Name Recognition Task," In BioCreative Challenge Evaluation Workshop, vol. 2, p. 171. 2013.

[49] X. Ma, and E. Hovy, “End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF,” Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, vol.1, pp-1064-1074, August 2016.

[50] A. Mansouri, L. S. Affendey, and A. Mamat, "Named Entity Recognition Approaches," International Journal of Computer Science and Network Security 8, no. 2 (2008): pp-339-344.

[51] Z. M. Maung, and Y. Mikami, “A Rule-based Syllable Segmentation of Myanmar Text,” In Proceeding of the IJCNLP-08 Workshop on NLP for Less Privileged Languages, pp-51-58, January 2008.

[52] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed Representations of Words and Phrases and their Compositionality,” In Advances in Neural Information Processing Systems (pp. 3111-3119), 2013.

[53] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” Available at: arXiv

preprint arXiv:1301.3781, 2013.

[54] S. Misawa, M. Taniguchi, Y. Miura and T. Ohkuma, “Character-based Bidirectional LSTM-CRF with word and characters for Japanese Named Entity Recognition,” In Proceedings of the First Workshop on Subword and Character Level Models in NLP, pp-97-102, Association for Computational Linguistics, September 2017.

[55] H. Momin, S. Jain, and H. Doshi, “Review Paper on Named Entity Recognition and Attribute Extraction using Machine Learning,” International Journal on Recent and Innovation Trends in Computing and Communication, vol.4, Issue. 11, pp 41-46, November 2016.

[56] H. Moradi, F. Ahmadi, Feizi-Derakhshi MR, “A Hybrid Approach for Persian Named Entity Recognition,” Iranian Journal of Science and Technology, Transactions A: Science. 2017 Mar 1;41(1):215-22.

[57] S. Morwal, J. Nusrat and C. Deepti, "Named Entity Recognition using Hidden Markov Model (HMM)," International Journal on Natural Language Computing (IJNLC) vol.1, no. 4 (2012): pp-15-23, December, 2012.

[58] A. D. Nguyen, K. H. Nguyen, and V. V. Ngo, "Neural Sequence Labeling for Vietnamese POS Tagging and NER," In 2019 IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF), pp. 1-5. IEEE, 2019.

[59] K. A. Nguyen, N. Dong, and C. T. Nguyen, "Attentive Neural Network for Named Entity Recognition in Vietnamese," In 2019 IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF), pp. 1-6. IEEE, 2019.

[60] K. Nongmeikapam, T. Shangkhunem, N. M. Chanu, L. N. Singh, B. Salam, and S. Bandyopadhyay, "CRF based Name Entity Recognition (NER) in Manipuri: A Highly Agglutinative Indian Language," In 2011 2nd National Conference on Emerging Trends and Applications in Computer Science, pp. 1-6. IEEE, 2011.

[61] A. Paszke, S. Gross, S. Chintala, G, Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison A, L. Antiga, and A. Lerer, “PyTorch” : https://pytorch.org/

[62] P. Pathak, R. Goswami, G. Joshi, P. Patel, and A. Patel, "CRF-based Clinical Named Entity Recognition using Clinical NLP," In Proceedings of

International Conference on Natural Language Processing.

[63] T. H. Pham, and P. Le-Hong, “The Important of Automatic Syntactic Features in Vietnamese Named Entity Recognition,” In Proceedings of the 31st Pacific Asia Conference on Language, Information and Computation (PACLIC 31), pp.97-103, November 2017.

[64] L. Prechelt, “Early stopping-but when?,” In Neural Networks: Tricks of the Trade (pp. 55-69). Springer, Berlin, Heidelberg, 1998.

[65] L. Ratinov, and D. Roth, "Design Challenges and Misconceptions in Named Entity Recognition," In Proceedings of the Thirteenth Conference on Computational Natural Language Learning, pp. 147-155. Association for Computational Linguistics, 2009.

[66] L. F. Rau, "Extracting company names from text," In [1991] Proceedings. The Seventh IEEE Conference on Artificial Intelligence Application, vol. 1, pp. 29-32. IEEE, 1991.

[67] M. Rei, G. K Crichton, S. Pyysalo. “Attending to Characters in Neural Sequence Labeling Models,” arXiv preprint arXiv:1611.04361. 2016 Nov 14.

[68] N. Reimers, and I. Gurevych, “Optimal Hyperparameters for Deep LSTM-Networks for Sequence Labeling Tasks,” [Online], Available: https://arxiv.org/abs/1707.06799 , August, 2017.

[69] S. K. Saha, S. Sarkar, & P. Mitra, “A Hybrid Approach for Named Entity Recognition for Indian Languages,” In Proceedings of IJCNLP NERSSEAL shared task, 2008.

[70] C. N. Santos, V. Guimaraes, “Boosting Named Entity Recognition with Neural Character Embedding,” arXiv preprint arXiv:1505.05008. 2015 May 19.

[71] S. Sekine, K. Sudo, and C. Nobata, “Extended Named Entity Herarchy,” In LREC, May 2002.

[72] K. Shaalan and M. Oudah. "A Hybrid Approach to Arabic Named Entity Recognition," Journal of Information Science, vol.40, no. 1 (2014): pp-67-87, 2014.

[73] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,”

The Journal of Machine Learning Research, 15(1), pp.1929-1958.,2014.

[74] J. Straková, M. Straka, and J. Hajič. "A New State-of-the-art Czech Named Entity Recognizer," In International Conference on Text, Speech and Dialogue, pp. 68-75. Springer, Berlin, Heidelberg, 2013.

[75] B. Strauss, B. Toma, A. Ritter, M. C. D. Marneffe, and W. Xu. "Results of the WNUT16 Named Entity Recognition Shared Task," In Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT), pp. 138-144. 2016.

[76] C. Sutton, and A. McCallum, "An Introduction to Conditional Random Fields," Foundations and Trends® in Machine Learning, vol.4, no. 4 (2012): 267-373.

[77] T. T. Swe, and H. H, Htay, "A Hybrid Method for Myanmar Name Entity Extraction and Transliteration to English," 2010.

[78] T. T. Thet, J. C. Na, and W. K. Ko, "Word Segmentation for the Myanmar Language," Journal of information science, vol.34, no. 5 (2008): 688-704.

[79] A. Thida and T. Myint, "Name Entity Recognition and Transliteration in Myanmar Text," Ph.D. dissertation, UCSM, 2004.

[80] E. F. Tjong Kim Sang, and F. D. Meulder, "Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition," CONLL '03 Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, vol.4, pp- 142-147, 2003.

[81] H. M. Wallach, "Conditional random fields: An introduction," Technical Reports (CIS) (2004): 22.

[82] C. Weber, and R. Vieira, "Building a Corpus for Named Entity Recognition using Portuguese Wikipedia and DBpedia," In I Workshop on Tools and Resources for Automatically Processing Portuguese and Spanish 2014 (pp. 9-15).

[83] A. S. Wibawa and A. Purwarianti, "Indonesian Named-entity Recognition for 15 Classes Using Ensemble Supervised Learning," In Proceeding of 5th Workshop on Spoken Language Technology for Under-resourced Languages, SLTU 2016, Procedia Computer Science, vol.81, pp-221-228, May 2016.

[84] Y. Wu, M. Jiang, J. Lei, and H. Xu, "Named Entity Recognition in Chinese

Clinical Text using Deep Neural Network," Studies in Health Technology and Informatics 216 (2015): 624.

[85] Y. Wu, M. Jiang, J. Xu, D. Zhi, and H. Xu, "Clinical Named Entity Recognition using Deep Learning Models," In AMIA Annual Symposium Proceedings, vol. 2017, p. 1812. American Medical Informatics Association, 2017.

[86] V. Yadaw, and S. Nethard, "A Survey on Recent Advances in Named Entity Recognition from Deep Learning Models," In Proceedings of the 27th International Conference on Computational Linguistics, pp-2145-2158, Association for Computational Linguistics, August 2018.

[87] J. Yang, and Y. Zhang, "NCRF++: An Open-source Neural Sequence Labeling Toolkit," In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics-System Demonstrations, Association for Computational Linguistics, pp-74-79, July 2018.

[88] J. Yang, S. Liang, and Y. Zhang, "Design Challenges and Misconceptions in Neural Sequence Labeling," In Proceeding of the 27th International Conference on Computational Linguistics, Association for Computational Linguistics, pp-3879-3889, August, 2018.

[89] Z. Yang, R. Salakhutdinov, W. Cohen, "Multi-task Cross-lingual Sequence Tagging from Scratch," arXiv preprint arXiv:1603.06270. 2016 Mar 20.

[90] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent Trends in Deep Learning based Natural Language Processing," IEEE Computational Intelligence magazine, vol.13, no. 3 (2018): 55-75.

[91] A.Yu, "How to Teach A Computer to See with Convolutional Neural Networks," https://towardsdatascience.com/how-to-teach-a-computer-to-see-with-convolutional-neural-networks-96c120827cd1

[92] Z. Zhai, D. Q. Nguyen, and K. Verspoor, "Comparing CNN and LSTM character-level embeddings in BILSTM-CRF models for chemical and disease named entity recognition," In Proceedings of the 9th International Workshop on Health Text Mining and Information Analysis (LOUHI 2018), pp-38-43, Association for Computational Linguistics, October 2018.

[93] Y. Zhang, and J. Yang, "Chinese NER Using Lattice LSTM," In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics,

pp.1554-1564, Association for Computational Linguistics, July 2018.

[94] G. Zhou, and J. Su, "Named Entity Recognition using an HMM-based Chunk Tagger," In proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 473-480. Association for Computational Linguistics, 2002.

[95] X. Zhu, "CS838-1 Advanced NLP: Conditional Random Fields. Technical report", The University of Wisconsin Madison, 2007.

[96] Asian Language Treebank (ALT) project: http://www2.nict.go.jp/astrec-att/member/mutiyama/ALT/

[97] Myanmar Language Commission. (2006), Myanmar Orthography. Third Edition, University Press, Yangon, Myanmar.

[98] MUC-6,"Named Entity Task Definition", 02 June 1995, https://cs.nyu.edu/cs/faculty/grishman/NEtask20.book_1.html

[99] "Unicode Character Code Table for Myanmar Language", [Online] Available: https://unicode.org/charts/PDF/U1000.pdf

APPENDICES

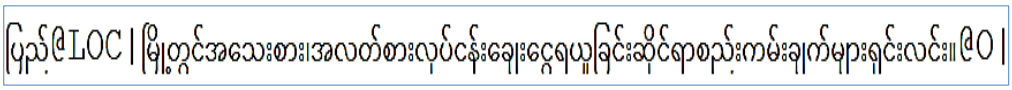
Appendix A: Development of Myanmar NE Tagged Corpus

Totally six types of Named Entity (NE) tags are defined and used to refer names in text. PNAME, LOC, ORG, RACE, TIME, and NUM tags are used to annotate respective NEs in text ant the tag O is used to annotate text that is not NEs. In this appendix, steps involved in developing Myanmar NE tagged corpus and preparation of input data format for neural sequence labeling will be described.

1. Development of Myanmar NE Tagged Corpus

News sentences from online official News websites as well as sentences from ALT parallel corpus were collected. Collected data has such noise as encoding inconsistency and typing errors and so on. All collected data were manually corrected in order to clean noisy data.

After data cleaning process, the training corpus was prepared by tagging sentences with defined NE tags manually.



2. Data Preparation for Neural Sequence Labeling

For the syllable segmentation, the python script described at: https://github.com/ye-kyaw-thu/sylbreak/tree/master/python is applied.

Input data format for train, development, and test data was prepared as the standard CoNLL-2003 data format. The data files contain two columns separated by a single space. Each syllable has been put on a separate line and there is an empty line after each sentence. The first token on each line is a syllable, and the second a NE tag. The named entity tags have the BIOES format.

To convert the NE tagged corpus into training data format, a python script written by the author was run as follows:

```
python tagSchemeLabel.py data
```

Appendix B: Experimental Setup for Neural Training

All experiments were conducted on Nvidia Tesla K80 GPU. As requirements, Python 2 or 3 must be installed. To download and install python, follow the installation note at python official website: https://www.python.org/.

For neural training, PyTorch, an open source deep learning platform that provides a seamless path from basics all the way into constructing deep neural networks, was applied.

To install the PyTorch binaries, it is needed to use at least one of two supported package managers: Anaconda and pip. Commands to install from binaries via Conda or pip wheel are on the PyTorch official website: https://pytorch.org.

In this setup for the experiments, PyTorch was installed from source: https://github.com/pytorch/pytorch#from-source. To do so, as prerequisites, Anaconda was firstly installed. To install Anaconda, Anaconda installation guide documentation can be referenced at https://docs.anaconda.com/anaconda/.

Once Anaconda had been installed, NVIDIA CUDA 9 was installed. After that, the required dependencies were installed via the following command:

```
conda install -c pytorch magma-cuda90
```

To get the PyTorch Source, run the following commands.

```
git clone --recursive https://github.com/pytorch/pytorch
cd pytorch
# if you are updating an existing checkout
git submodule sync
git submodule update --init --recursive
```

To install PyTorch, run the following commands.

```
export CMAKE_PREFIX_PATH=${CONDA_PREFIX:-"$(dirname $(which
conda))/../"}
python setup.py install
```

To check whether PyTorch was successfully installed or not, run the following commands.

```
Python
>>import torch
>>torch.cuda.is_available()
True
```

If True is returned, it is ready to do neural training on the machine.

And then, neural NER model was trained by configuring various neural architectures and different hyperparameter settings.

When the neural model training is finished, the decoding processing is performed by using the trained model.

Appendix C: Experimental Setup for Baseline CRF

For CRF training, CRF++: Yet Another CRF toolkit is used. The official web address is: https://taku910.github.io/crfpp/. CRF++ is an open source implementation of Conditional Random Fields for sequence learning and can be applied to a variety of NLP tasks. It is simple and customizable and is designed for generic purpose. Features sets can be easily redefined.

C++ compiler (gcc 3.0 of higher) is required. To make and install, follow the following commands.

```
% ./configure
% make
% su
# make install
```

Before training, a feature template which describes what features are used in training and testing must be prepared in advance. After preparing the feature template, and training data file, training process can be started by the following command.

```
% crf_learn template_file train_file model_file
```

The command crf_learn generates the trained model file. If parameters are needed to control the condition, run by inserting parameters as follows:

```
% crf_learn -f 3 -c 1.5 template_file train_file model_file
```

where –f and –c are parameters.

For decoding, use the crf_test command.

```
% crf_test -m model_file test_files
```

where model_file is the file crf_learn creates.

Appendix D: Some Examples Output of the Proposed Syllable-based Neural NER Model

Human Annotated Reference	Model Predicted Output
|စက်တင်ဘာ@TIME||၁၅@TIME|ရက်၌ ဒီမိုကရေစီရေး ဆန္ဒပြသူများသည် |ဟောင်ကောင်@LOC|မြို့သူမြို့သားများကို ကာကွယ်စောင့်ရှောက်ပေးရန် နှင့် လွတ်လပ်ခွင့်များ လျော့ပါးလာစေခြင်းနှင့် ပတ်သက်၍ |တရုတ်@LOC|ကို ဖိအားပေးမှု မြှင့်တင်ရန် |ဖြိတိန်@LOC|ကိုတောင်းဆိုလျက် |ဟောင်ကောင်@LOC|ရှိ|ဖြိတိန်ကောင်စစ်ဝန်ရုံး@LOC|အပြင်ဘက်တွင်စုဝေးဆန္ဒပြခဲ့ကြသည်။|	|စက်တင်ဘာ@TIME||၁၅@TIME|ရက်၌ ဒီမိုကရေစီရေးဆန္ဒပြသူများသည် |ဟောင်ကောင်@LOC|မြို့သူမြို့သားများကို ကာကွယ်စောင့်ရှောက်ပေးရန် နှင့် လွတ်လပ်ခွင့်များလျော့ပါးလာစေခြင်းနှင့် ပတ်သက်၍|တရုတ်@LOC|ကိုဖိအားပေးမှု မြှင့်တင်ရန်|ဖြိတိန်@LOC|ကိုတောင်းဆိုလျက် |ဟောင်ကောင်@LOC|ရှိ|ဖြိတိန်ကောင်စစ်ဝန်ရုံး@LOC|အပြင်ဘက်တွင်စုဝေးဆန္ဒပြခဲ့ကြသည်။|
|သာကေတ@LOC|မြို့နယ်|ဧရာဝဏ်@LOC| လမ်းမ ကြီးပေါ်ရှိ|မြန္မာညှို@LOC|အိမ်ရာအနီး လမ်းလျှောက်လာသည့်အမျိုးသမီးတစ်ဦးထံမှ ရွှေဆွဲကြိုးဖြတ်ပြေးသည့် အမျိုးသားအား အနီး ပတ်ဝန်းကျင်ရှိလူအများကဝိုင်းဝန်းဖမ်းဆီးမှု |စက်တင်ဘာ@TIME||၁၃@TIME|ရက်ည|၈@TIME|နာရီ|၁၅@TIME|မိနစ်ခန့်က ဖြစ်ပွားခဲ့သည်။|	|သာကေတ@LOC|မြို့နယ်|ဧရာဝဏ်@LOC| လမ်းမ ကြီးပေါ်ရှိ|မြန္မာညှို@LOC|အိမ်ရာအနီး လမ်းလျှောက်လာသည့်အမျိုးသမီးတစ်ဦးထံမှ ရွှေဆွဲကြိုးဖြတ်ပြေးသည့် အမျိုးသားအားအနီး ပတ်ဝန်းကျင်ရှိလူအများကဝိုင်းဝန်းဖမ်းဆီးမှု |စက်တင်ဘာ@TIME||၁၃@TIME|ရက်ည|၈@TIME|နာရီ|၁၅@TIME|မိနစ်ခန့်က ဖြစ်ပွားခဲ့သည်။|
|၁၉၉၇@TIME|ခုနှစ်|ဟောင်ကောင်@LOC|ကို| တရုတ်@LOC|သို့လွှဲပြောင်းပေးခြင်း မပြုမီ |ဖြိတိန်@LOC|နှင့်လက်မှတ်ရေးထိုးထား သောသဘောတူညီချက်တစ်ရပ် အရ |ဟောင်ကောင်@LOC|ကိုနှစ်|၅၀@NUM| အတွင်းသူ၏သီးခြားလွတ်လပ်ခွင့်များအား ဆက်လက်ခွင့်ပြုထားပေးရမည်ဖြစ်သည်။|	|၁၉၉၇@TIME|ခုနှစ်|ဟောင်ကောင်@LOC|ကို| တရုတ်@LOC|သို့လွှဲပြောင်းပေးခြင်း မပြုမီ |ဖြိတိန်@LOC|နှင့်လက်မှတ်ရေးထိုးထား သောသဘောတူညီချက်တစ်ရပ် အရ |ဟောင်ကောင်@LOC|ကိုနှစ်|၅၀@NUM| အတွင်းသူ၏သီးခြားလွတ်လပ်ခွင့်များအားဆက် လက်ခွင့်ပြုထားပေးရမည်ဖြစ်သည်။|
|ရှမ်း@LOC|ပြည်နယ်(တောင်ပိုင်း)|တောင်ကြီး@LOC|ခရိုင်|ညောင်ရွှေ@LOC|မြို့နယ်ရှိပြည်တွင်းပြည်ပခရီးသွားများလာရောက်လည်ပတ်လျက်ရှိသည့်|အင်းလေးကန်@LOC|ရေရှည်တည်တံ့ခိုင်မြဲစေရန်|အင်းလေးကန်@LOC| ရေဝေရေလဲဒေသရှိသစ်ပင်ပေါက်ရောက်မှုမရှိသောတောင်ကတုံးနေရာ၈က|၂၅၀@NUM| ပေါ်တွင်သစ်စေ့မျိုးပေါင်း|၂၀@NUM| ထည့်သွင်းထားသည့်မြေလုံး(SeedBalls)ပေါင်း| ၁၄၆၂၉၇@NUM|လုံးကို|စက်တင်ဘာ@TIME| |၁၄@TIME|ရက်နံနက်|၉@TIME|နာရီက ရဟတ်ယာဉ်ဖြင့်ကြဲချစိုက်ပျိုးခဲ့သည်။|	|ရှမ်း@LOC|ပြည်နယ်(တောင်ပိုင်း)|တောင်ကြီး@LOC|ခရိုင်|ညောင်ရွှေ@LOC|မြို့နယ်ရှိပြည်တွင်းပြည်ပခရီးသွားများလာရောက်လည်ပတ်လျက်ရှိသည့်|အင်းလေး@LOC|ကန်ရေရှည်တည်တံ့ခိုင်မြဲစေရန်|အင်းလေးကန်@LOC| ရေဝေရေလဲဒေသရှိသစ်ပင်ပေါက်ရောက်မှုမရှိသောတောင်ကတုံးနေရာ၈က|၂၅၀@NUM| ပေါ်တွင်သစ်စေ့မျိုးပေါင်း|၂၀@NUM| ထည့်သွင်းထားသည့်မြေလုံး(SeedBalls)ပေါင်း| ၁၄၆၂၉၇@NUM|လုံးကို|စက်တင်ဘာ@TIME| |၁၄@TIME|ရက်နံနက်|၉@TIME|နာရီက ရဟတ်ယာဉ်ဖြင့်ကြဲချစိုက်ပျိုးခဲ့သည်။|
|သယံဇာတနှင့်သဘာဝပတ်ဝန်းကျင်ထိန်းသိမ်းရေးဝန်ကြီးဌာန@ORG|ကပုံပိုးကူညီကာ |သစ်တောဦးစီးဌာန@ORG|နှင့်|ရှမ်း@LOC|	|သယံဇာတနှင့်သဘာဝပတ်ဝန်းကျင်ထိန်းသိမ်းရေးဝန်ကြီးဌာန@ORG|ကပုံပိုးကူညီကာ |သစ်တောဦးစီးဌာန@ORG|နှင့်|ရှမ်း@LOC|

ပြည်နယ်အစိုးရအဖွဲ့တို့ကဖြည့်ဆည်းဆောင်ရွက်ပြီး၊ထူးကုမ္ပဏီအုပ်စု@ORG၊ပိုင်ရဟတ်ယာဉ်ဖြင့်ကြဲချခဲ့ခြင်းဖြစ်သည်။| ပြည်နယ်အစိုးရအဖွဲ့တို့ကဖြည့်ဆည်းဆောင်ရွက်ပြီး၊ထူးကုမ္ပဏီအုပ်စု@ORG၊ပိုင်ရဟတ်ယာဉ်ဖြင့်ကြဲချခဲ့ခြင်းဖြစ်သည်။|
၊ရှမ်း@LOC၊ပြည်နယ်(မြောက်ပိုင်း)၊နမ့်ဆန်@LOC၊မြို့နယ်၊ဟိုချောင်း@LOC၊ကျေးရွာအုပ်စု၊တောင်ရိုးစေတီ@LOC၊အနီးတွင်၊စက်တင်ဘာ@TIME၊၁၃@TIME၊ရက်နံနက်၊၅@TIME၊နာရီမှစ၍လက်နက်ကိုင်အဖွဲ့နှင့်၊တပ်မတော်@ORG၊တို့အကြားတိုက်ပွဲထိတွေ့မှုများကြောင့်၊နမ့်ဆန်@LOC၊မြို့၊ဇယန်းကြီးကျောင်းတိုက်@LOC၊သို့ တိုက်ပွဲရှောင်များ ရောက်ရှိလာရာ၊၁၄@TIME၊ရက်ညနေအထိ၊၇၁၅@NUM၊ဦးရှိသည်။| ၊ရှမ်း@LOC၊ပြည်နယ်(မြောက်ပိုင်း)၊နမ့်ဆန်@LOC၊မြို့နယ်၊ဟိုချောင်း@LOC၊ကျေးရွာအုပ်စု၊တောင်ရိုးစေတီ@LOC၊အနီးတွင်၊စက်တင်ဘာ@TIME၊၁၃@TIME၊ရက်နံနက်၊၅@TIME၊နာရီမှစ၍လက်နက်ကိုင်အဖွဲ့နှင့်၊တပ်မတော်@ORG၊တို့အကြားတိုက်ပွဲထိတွေ့မှုများကြောင့်၊နမ့်ဆန်@LOC၊မြို့၊ဇယန်းကြီးကျောင်းတိုက်@LOC၊သို့ တိုက်ပွဲရှောင်များ ရောက်ရှိလာရာ၊၁၄@NUM၊ရက်ညနေအထိ၊၇၁၅@NUM၊ဦးရှိသည်။|
၊ကူမင်း@LOC၊မြို့သည်၊တရုတ်@LOC၊အနောက်ပိုင်းဒေသတွင်စီးပွားတိုးတက်မှုကောင်းမွန်နေသောဖြင့် IKEAစတိုးကိုဖွင့်လှစ်ရန်ရွေးချယ်ခြင်းဖြစ်ကြောင်း IKEA၊တရုတ်@LOC၊ဌာနခွဲဒါရိုက်တာ၊လူကတ်ဇ်အိုထရိုစကီ@PNAME၊ကပြောသည်။| ၊ကူမင်း@LOC၊မြို့သည်၊တရုတ်@LOC၊အနောက်ပိုင်းဒေသတွင်စီးပွားတိုးတက်မှုကောင်းမွန်နေသောဖြင့် IKEAစတိုးကိုဖွင့်လှစ်ရန်ရွေးချယ်ခြင်းဖြစ်ကြောင်း IKEA၊တရုတ်@LOC၊ဌာနခွဲဒါရိုက်တာ၊လူကတ်ဇ်အိုထရိုစကီ@PNAME၊ကပြောသည်။|
၊ဂါနီ@PNAME၊သည်လူထုစုဝေးပွဲတွင်ရှိနေခဲ့သော်လည်းထိခိုက်ဒဏ်ရာရရှိခဲ့ခြင်းမရှိကြောင်းဂင်း၏ မဲဆွယ်စည်းရုံးရေးတာဝန်ခံကပြောကြားခဲ့သည်။| ၊ဂါနီ@PNAME၊သည်လူထုစုဝေးပွဲတွင်ရှိနေခဲ့သော်လည်းထိခိုက်ဒဏ်ရာရရှိခဲ့ခြင်းမရှိကြောင်းဂင်း၏ မဲဆွယ်စည်းရုံးရေးတာဝန်ခံကပြောကြားခဲ့သည်။|
၊စက်တင်ဘာ@TIME၊၂၈@TIME၊တွင်သမ္မတရွေးကောက်ပွဲများကျင်းပရန်၊အာဖဂန်@LOC၊ကပြင်ဆင်နေချိန်တွင်တိုက်ခိုက်မှုများဖြစ်ပွားခဲ့ခြင်း ဖြစ်သည်။| ၊စက်တင်ဘာ@TIME၊၂၈@TIME၊တွင်သမ္မတရွေးကောက်ပွဲများကျင်းပရန်၊အာဖဂန်@LOC၊ကပြင်ဆင်နေချိန်တွင်တိုက်ခိုက်မှုများဖြစ်ပွားခဲ့ခြင်း ဖြစ်သည်။|
၊မိုးကုတ်@LOC၊မြို့မှာ၊ဗမာ@RACE၊၊ရှမ်း@RACE၊၊လီဆူ@RACE၊၊ပလောင်@RACE၊၊ကေါ်ရခါး@RACE၊၊တရုတ်@RACE၊တို့အများဆုံးနေထိုင်ကြပါတယ်။| ၊မိုးကုတ်@LOC၊မြို့မှာ၊ဗမာ@RACE၊၊ရှမ်း@RACE၊၊လီဆူ@RACE၊၊ပလောင်@RACE၊၊ကေါ်ရခါး@RACE၊၊တရုတ်@RACE၊တို့အများဆုံးနေထိုင်ကြပါတယ်။|