MYANMAR WORD STEMMING AND POS TAGGING USING RULE BASED APPROACH

KYAW HTET MINN

M.C.Sc. **MARCH 2019**

MYANMAR WORD STEMMING AND POS TAGGING USING RULE BASED APPROACH

By

KYAW HTET MINN

B.C.Sc. (Hons:)

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree

of

Master of Computer Science

(M.C.Sc.)

University of Computer Studies, Yangon.

MARCH 2019

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my sincere thanks to those who helped me with various aspects of conducting research and writing this thesis.

First and foremost, I would like to express my gratitude and my sincere thanks to **Prof. Dr. Mie Mie Thet Thwin**, Rector, the University of Computer Studies, Yangon, for allowing me to develop this thesis.

I would like to express my special appreciation and my sincere special thanks to **Dr. Thi Thi Soe Nyunt**, Professor and Head of Faculty of Computer Science, the University of Computer Studies, Yangon, for her kind administrative supports and encouragements in development of the thesis.

I am heartily thankful to my supervisor, **Dr. Khin Mar Soe**, Professor, Natural Language Processing Lab, the University of Computer Studies, Yangon, for her strong technical supports, invaluable guidance, encouragement, superior suggestion and supervision on the accomplishment of this thesis.

I am indebted to **Daw Hnin Yu Mon,** Lecturer, Language Department of the University of Computer Studies, Yangon, for her guidance and editing my thesis from the language point of view.

I would also thank our teachers from the University of Computer Studies, Yangon, for their thoughtful and help for my seminar presentation. Moreover, I would like to extend my thanks to all my teachers for their support not only for the fulfillment of the degree of M.C.Sc. but also for my life. I also thank all the departmental staff for helping me during these years of my study both academically and officially.

It is a pleasure to thank my family, friends and colleagues for supporting in various ways to complete this thesis. Lastly, I offer my regards and blessings to all of those who supported me in any way during the completion of the thesis.

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 Kyaw Htet Minn

ABSTRACT

Myanmar language is spoken by more than 33 million people and use it as a verbal and written communication which is an official language of the Republic of the Union of Myanmar. With the rapid growth of digital content in Myanmar Language, applications like machine learning, translation and information retrieval become popular and it required to obtain the effective Natural Language Processing (NLP) studies. The NLP field on Myanmar language still has a big challenge. Segmenting, stemming and Part-Of- Speech (POS) tagging are pre-processing steps in Text Mining applications as well as a very common requirement of Natural Language processing functions. In fact, it is very important in most of the Information Retrieval systems. The main objective of this thesis is to study Myanmar words morphology, to implement n-gram based word segmentation and to propose grammatical stemming rules and POS tagging rules for Myanmar language. This thesis proposed the word segmentation, stemming and POS tagging based on n-gram method and rule-based stemming method that has the ability to cope the challenges of Myanmar NLP tasks. This system not only generates the segmented words but also generates the stemmed words with POS tag by removing prefixes, infixes and suffixes. It provides 82 % accuracy. The data are collected from several online sources and the system is implemented using Python language.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

CHAPTER 1

INTRODUCTION

With the rapid growth of the information technology, the availability of the research projects on language processing have been considerably increased. The most commonly researched tasks in NLP are Lemmatization, Tokenization, Segmentation, Stemming, Part-of-speech (POS) tagging, Machine learning, Translation, Optical character recognition(OCR), Classification, Text-to-speech, Sentiment analysis, Categorization, and so on. Most of the developed countries' languages have a various successful research in NLP. But for the developing countries such as Myanmar, the language still needs to study and research in order to achieve useful information. The proposed approach is evaluated on segmenting, stemming and POS tagging and they have a vital role in many applications of NLP i.e. Information Retrieval (IR) system, Machine Translation, Building index, Domain Analysis and Search Engine.

Myanmar language is a national language of the Republic of the Union of Myanmar. Myanmar language has two distinguishable registers known as Literary High form and Spoken Low form. Both of them are written from left to right and no spaces between words, although informal writing often contains spaces after each clause. It is syllabic alphabet and written in circular shape. It has sentence boundary marker. It is a free-word-order and verb final language, which usually follows the subject-object-verb (SOV) order. [19] In particular, preposition adjunctions can appear in several different places of the sentence. Myanmar language users normally use space as they see fit, some write with no space at all. There is no fixed rule for word segmentation. Word segmentation is an essential task in preprocessing stage for text mining processes. Many researchers have been carried out Myanmar word segmentation in many ways including both supervised and unsupervised learning. We use Myanmar Word Segmentation with N-gram matching approach in this thesis.

Word stemming is another important feature supported by present day indexing and search systems. Indexing and searching are in turn part of Text Mining applications, Natural Language Processing (NLP) systems and Information Retrieval (IR) systems. The main purpose of stemming is to reduce different grammatical forms / word forms of a word like its noun, adjective, verb, adverb etc. to its root form. The main idea is to improve recall by automatic handling of word endings by reducing the words to their

word roots, at the time of indexing and searching. Stemming is usually done by removing any attached affixes from index terms before the actual assignment of the term to the index. Since the stem of a term represents a broader concept than the original term, the stemming process eventually increases the number of retrieved documents in an IR system. Text normalization, Syllabification and Segmentation also require this conversion as part of the pre-processing before actually applying related algorithm. [17] Part-of-speech(POS) tagging is the especially crucial for the disambiguation of a word. In this paper, we tackle the POS tagging with stemming in same framework by matching the clues word.

1.1 Related Works

Stemmer has an important role to improve the efficiency as well as performance of the IR systems. Until now, many of stemming methods have been researched for a wide variety range of languages such as Arabic, English, Persian, Dutch, etc. Lovin's Stemmer [7] was the very first stemmer, which was published and written by Julie Beth Lovin's in 1968. This paper was popular and achieved remark for its early date and has great influence on later work in the area of stemming. In this stemmer, Lovin's defined 260 rules for stemming English word. Lovin's stemmer processed the stem of English words in two phases. In the first phase of this stemming method, maximum matching suffix defined in suffix table is removed. Spelling exclusions are dealt in second phase. Dawson [8] employed another rule-based stemming method. It is an extension of J.B. Lovin's stemmer covers a comprehensive list of 1200 suffixes. Then in July 1980, Martin Porter from the University of Cambridge developed the "Porter Stemmer", which is a rule based stemmer with five steps using a set of rules. [12] This stemming technique eliminates the suffixes form words by using suffix list and some conditions are applied to find out which suffixes are needed to be removed. Porter reduced the Lovin's rules up to 60. Porter found out the problems of over-stemming and under-stemming based on suffix removal. This stemmer was widely used and became the standard algorithm applied for English language. Later, other few stemmers were developed by Paice [4] & Husk, Dawson and Krovetz. [16] Most of the above stemmers were rule based stemmers which followed the Suffix Stripping approach. Among these, the Porter Stemmer has proved to be an extremely useful resource to researchers who work on stemmers and it has been also applied to languages other than English.

Since the establishment of the concept of stemming in 1968, a lot of work had been done in English and other European languages. Rule based stemmer for Asian languages such as Japanese, Thailand, Malay, Indian and Nepali are also found but a little work had been done in Myanmar language.

Natural Language Processing research in Myanmar Language started in the year (2006) at University of Computer Studies, Yangon (UCSY) under Ministry of Science and Technology with the release of the first English-Myanmar translation project. Corpus building and annotation for Myanmar, word segmentation, Text-To-Speech System for Myanmar, digitized Myanmar dictionary and many other research projects were also started in the followed years. [6] These works motivated researcher to do further research and implement of Natural Language Processing functions in Myanmar language. Even though works on other fields are researched, so far up to the best of author's knowledge, no works on only focused on Stemmer is being found for the Myanmar language yet. So, this stemmer is the first work in Myanmar language which has both prefixes, infixes and suffixes stemming capabilities.

1.2 Overview of the System

This system has five steps: Normalization, Syllabification, Segmentation, Stemming and POS Tagging.

The very first step to the system is to polish the input sentence. Normalization is a process that converts a list of words to a more uniform sequence.

After the normalization, the next step is syllabification. Syllabification is the task of breaking a sequence of words into syllables.

The third step is segmentation. In this study, N-gram matching techniques is used. By matching with the predefined prefix and suffix lexicons, the input sentence is segmented with the correct lexicon and result as a space separated sentence.

After processing all the above steps, the segmented words are ready to stem. The proposed stemmer was developed using look up based approach using three corpus which are prefix, infix and suffix.

The final steps is POS tagging. POS tagging is working under the same framework with stemming algorithm. The POS tag is attached to the word according to rule indicator. In this system, not only the stemmed word but also the stripped word is tagged and stored in a separated file.

1.3 Objectives of the Thesis

The objectives of the thesis are as follows:

- To study Myanmar words morphology
- To implement n-gram based word segmentation
- To propose grammatical stemming rules and POS tagging rules for Myanmar language
- To be able to use this system as a pre-processing tool in Myanmar text processing such as Information Retrieval, Machine Translation, Search Engine and other NLP research and projects

1.4 Organization of the Thesis

This thesis is organized into five chapters.

Chapter 1 includes introduction of the proposed system, overview and objectives of this system.

Chapter 2 describes the background theory of this system, Myanmar Language, normalization, syllabification, segmentation, stemming and POS tagging with N-gram and rule based approach.

Chapter 3 explains the overall design of the system, class diagram and flow diagram of the system.

Chapter 4 presents implementation of proposed system which includes system algorithms and screen designs of the system.

The final chapter, Chapter 5 presents the conclusion of this thesis, benefits and, limitations and further extension of the system.

CHAPTER 2

THEORETICAL BACKGROUND

In this chapter, background theory of Natural Language Processing work used in this research such as segmentation, stemming and POS tagging are mainly discussed.

Natural Language Processing deals with the interactions between human languages and computer languages. Every work of NLP is applied in the field of computer science, artificial intelligence and computational linguistics such as optical character recognition, script recognition, part of speech (POS) tagging, sentiment analysis, social media analysis, information retrieval, information extraction and so on. Although natural language processing is not a new research science work, the information technology nowadays is rapidly developing into an increased amount in human-to-machine communications, and also an availability of big data, advanced programming language support and powerful hardware computer are pushing natural language processing research to continue. Natural Language Processing makes a lot of support in communication between computers and humans in their own language. [18] Moreover, other language-related tasks has been improved because of Natural Language Processing. For example, NLP makes it happen for computers to hear text, read text, translate, interpret, classify, measure sentiment and determine the key point. Human being language is enormously complex and diverse. The language itself expresses in infinite ways, both in writing and verbally. There is a unique set of syntax rule, grammar, slang and terms in not only with each language but also in every different languages across the world. When we write these language, we make misspell, omit punctuation or abbreviate words. The NLP has to accomplish these errors or tasks in many different ways. Making computer to be received and working in the real natural language meaning is the most difficult things to do in NLP work. There have a big challenges in each language since they have problems such as one single word having several meanings (polysemy) or different words having similar meanings (synonymy). It is a researcher work to encode rules to make the language easier and simpler so that the computer can process the language very well. [18]

2.1 Myanmar Language

The Myanmar language is a national language of the Republic of the Union of Myanmar and it is also known as Burmese. It has been used for more than one thousand years old. it is spoken by 33 million people and also the 38th most spoken language in the world. Myanmar language is a member of Tibeto-Burman languages, which is also a subfamily of the Sino-Tibetan family. It is phonologically based script and they are adapted from Mon and descended from the Brahmi script of ancient South India. Myanmar language has two distinguishable registers called as Literary High and Spoken Low. Both of these form are expressed and written from left to right and no white spaces between syllables or between words, but in formal writing style, it often contains spaces after each clause. Characters in Myanmar language are round in shape. According to Dr. Judson, the pure Myanmar language is monosyllabic that is every word consist of only one syllable. But the combination of the Pali language which is also call Buddhistic religion found many polysyllabic words into Myanmar language. It contains sentence boundary marker. It usually follows the subject-object-verb (SOV) order. Myanmar word consists of one or more morphemes that are more or less tightly linked together. Myanmar syllable can also be composed of multiple characters. So, syllable segmentation is also an important process have to make by means of an algorithm by computer. Besides the syllable segmentation, word segmentation is also an essential task for preprocessing of Natural Language Processing works. [13]

2.1.1 Basic Consonants

A Myanmar sentence is a series of characters without explicit word boundary markup, written in sequence from left to right without regular inter-word white spacing, although white spacing may sometimes be used. Myanmar characters can be categorized into three groups: consonants, vowels and medials. The basic consonants in Myanmar can be multiplied by medials. Syllables or words are formed by consonants combining with vowels. However, some syllables can be formed by just consonants, without any vowel. Other characters in the Myanmar script include special characters, numerals, punctuation marks and signs.

There are 33 basic consonants in the Myanmar script, as described in Figure 2.1. They are known as “Byee” in the Myanmar language. Consonants serve as the base characters of Myanmar words.

က	ka	ဋ	ṭa	ပ	pa
ခ	kha	ဌ	ṭha	ဖ	pha
ဂ	ga	ဍ	ḍa	ဗ	ba
ဃ	gha	ဎ	ḍha	ဘ	bha
င	ṅa	ဏ	ṇa	မ	ma
စ	ca	တ	ta	ယ	ya
ဆ	cha	ထ	tha	ရ	ra
ဇ	ja	ဒ	da	လ	la
ဈ	jha	ဓ	dha	ဝ	va
ည	ñña	န	na	သ	sa
				ဟ	ha
				ဠ	ḷa
				အ	’a

Figure 2.1. Basic Consonants

2.1.2 Vowels

Vowels are known as “Thara”. Vowels are the basic building blocks of syllable formation in the Myanmar language, although a syllable or a word can be formed from just consonants, without a vowel as shown in Figure 2.2. Like other languages, multiple vowel characters can exist in a single syllable. There have ten vowels and eight independent vowels as described in the figure 2.2. [19]

◌ာ, ◌ါ	◌ā	◌ေ	◌e	ဣ	i
◌ိ	◌i	◌ဲ	◌ai	ဤ	ī
◌ီ	◌ī	◌ော, ◌ေါ	◌o	ဥ	u
◌ု , ◌ုံ	◌u	◌ော်, ◌ေါ်	◌o‘	ဦ	ū
◌ူ , ◌ူံ	◌ū	◌ို , ◌ိုံ	◌ui	ဧ	e
				ဩ	o
				ဪ	o‘

Figure 2.2. Vowels

2.1.3 Medials

Medials are known as “Byee Twe” in Myanmar. There are 4 basic medials and 7 combined medials in the Myanmar script. So the 11 medials can modify the 34 basic consonants to form 340 additional multi-clustered consonants. Therefore, a total of 374 consonants exist in the Myanmar script, although some consonants have the same pronunciation. [19]

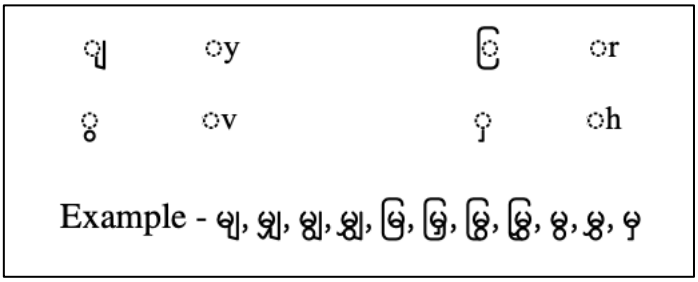

Figure 2.3. Medials

2.1.4 Other Characters

Other characters for Myanmar language are final symbols, tone marks and punctuation. Final symbols are used over any of the syllable-final consonants when no stacking takes place. Tone marks are used for encompassing a variety of pitches. The two punctuation in Myanmar language are like comma and period in English language. [19]

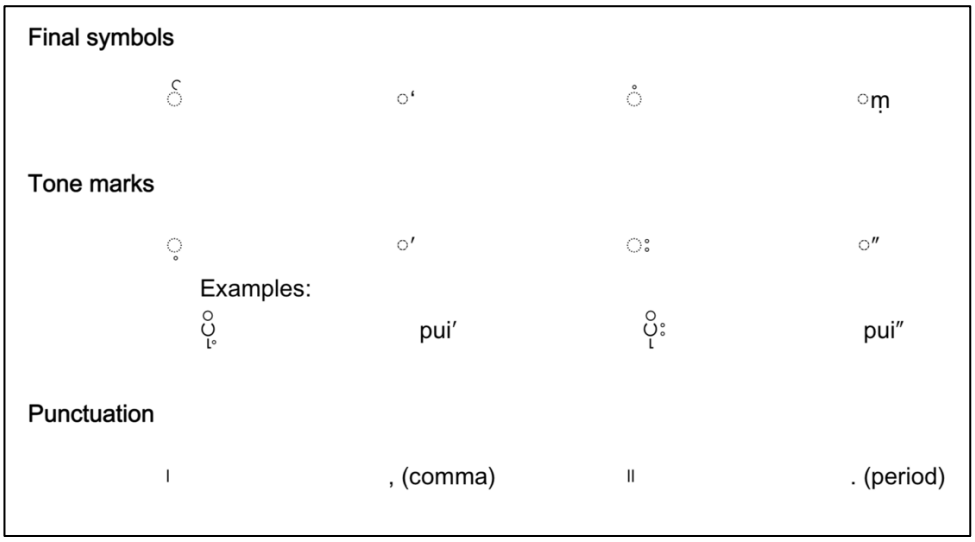

Figure 2.4. Other Characters

2.1.5 Grammar

There are nine Part-of-Speech in Myanmar language grammar. They are Nouns, Pronouns, Adjectives, Verbs, Adverbs, Conjunctions, Particles, Postpositional Markers and Interjections. From the English language point of view, Myanmar grammar classifications and terminologies are not easy to explain because of particles and postpositional markers which do not exactly fit nicely into the pattern of English grammar rules and classifications.

2.1.6 Computer Fonts and Standard Keyboard Layout

The Myanmar language can be typed from a standard QWERTY keyboard, which can be read and written in all computers and smartphones. Even though the most popular Myanmar font, Zawgyi, has been introduced in 2007, it does not follow the Unicode encoding rule. So, the ubiquitous use of Zawgyi causes a hard time in Myanmar languages by preventing efficient and effective of searching, sorting, analyzing and processing. So, lately, Myanmar3 layout has been established as the national standard keyboard layout. It was published along with the Myanmar3 Unicode font which is developed by the Myanmar Unicode and NLP Research Center.

2.2 Normalization

Normalization is the action of converting text into a single canonical form that it might or might not have had before. Normalizing text before processing or storing is performing in order to be consistent before operations are performed on it. Normalization normally requires to make a set of specific rule to be done by deciding which type of text is to be normalized and how it has to be processed afterwards; there is no standardized purpose of normalization procedure. What kind of input needs to be normalized only depends on the researcher. Text normalization is generally applied in transforming text to speech, dates, numbers, acronyms, abbreviations, slang and non-standard words.

Nowadays, social media have been increasingly used by people which has resulted a new form of written text called microtext. Moreover, in Myanmar language there have two main different encoding systems named Zawgyi and Unicode. These two facts pose new challenges to natural language processing research which is usually designs for formal or correct text.

The techniques for handling the normalization are commonly using statistical machine translation, spelling correction and automatic speech recognition.

2.2.1 Statistical Machine Translation

This method is regarding microtext as a foreign language that needs to be translated, this means that normalization is processed through a statistical machine translation (SMT) task. This technique is very straightforward and superior as it develops into context dependent one-to-many model. However, the statistical machine translation still overlooks few features of the task, especially the case that lexical creativity verified in social media content is hardly captured in a fixed sentence board.

2.2.2 Spelling Correction

This technique of correction is working on a word per word basic as a spelling checking task. This model obtained a considerable attention in the past and also a diversity of correction techniques that have been developed by E. Cambria and D. R. Recupero. Alternatively, research on sentiment analysis of social media text with rule based model and unsupervised translation was used to predict their probability of occurrence. So, the spelling correction technique for normalization has become popular and in later advanced with the hidden Markov model which topology takes into account both omissions of repeated letters, typos and spelling that resemble the word's pronunciation. However, the spelling checking model only focuses on the normalization of words but ignores their respective context.

2.2.3 Automatic Speech Recognition

The last metaphor examines that microtext is seen to be a closer approximation of the phonemic representation of a word when contrasting to its standard spelling. This technique of normalization is very similar to speech recognition which contains a word sequence decoding in a phonetic framework. For example, observation on text content as normalization presents many phonetic spellings. Although the calculation of a phonemic representation of the text content is excessively valuable, it does not resolve completely all the challenges of microtext normalization. For example, misspellings and acronyms do not feature their respective word's phonemic representation.

2.3 Syllabification

Syllabification is the work on splitting a word into syllables. Myanmar script are written with no space between words and syllable segmentation or syllabification stand out a significant process in many NLP research such as sorting, searching, segmenting, line breaking and so on. Segmentation rules were created based on the syllable structure constructed in Myanmar script and a syllabification algorithm was designed based on the created rules. There is no official standard encoding of Myanmar language and very few previous studies on the syllable segmentation of Myanmar language. Most syllabification approaches use a dictionary. However, the accuracy of syllabification depends on the quality and amount of the word contained in dictionary. So, unknown words and micro text can reduce the performance. Generally, the method used for syllabification are maximal onset principle, sonority sequence principle, finite state automaton and regular expression.

2.3.1 Regular Expression

Regular expression is one of the best method to explicit the languages acquired by the finite automaton which is an algebraic notation for defining a set of strings. Some important properties of regular expression which are used for researching with algorithms relating to syllabification are described in equation 2.1 and equation 2.2.

Let L, L_1, and L_2 are sets of strings from $\sum^*$ and $\sum$ be a finite set of symbols. The concatenation of L_1 and L_2 denoted by L_1L_2 is the set {xy|x is in L_1 and y is in L_2}. The strings in L_1L_2 are generated by selecting string L_1 and following it by a string in L_2, in all possible combinations. Define L^0={ ε} and L^i= LL^{-i} for i≥1. The Kleene closure of L, denoted L^*, is the set

$$L^* = \bigcup_{i=0}^{\infty} L^{-i} \qquad 2.1$$

and the positive closure of L, denoted by L^+, is the set

$$L^+ = \bigcup_{i=0}^{\infty} L^i \qquad 2.2$$

To explain, L^* denotes words constructed by concatenating any number of words from L. Similarly L^+ also represents the same but the zero words case whose concatenation is defined to be ε, is excluded. It is known that L^* contains ε if L does.

If r and s are regular expressions denoting the regular language R and S respectively, the (r+s), (rs), (r^*) are regular expressions that denote the sets $R \cup S$, RS, and R^*, respectively.

2.3.2 Keynotes about Regular Expression

The languages for regular expression are closed under difference, intersection, reversal and complementation operations. The regular expression languages are greatly applied in pattern matching. Some important keynotes about regular expression are:

1. They are case sensitive.
2. A simple regular expression pattern is a sequence characters delimited by forward slash.
3. Any sequence of characters delimited by square braces will match any one character from the sequence. If the first character is '^', it means it cannot be.
4. The use of '?' matches "zero or one instance of previous character"
5. The use of '.' matches any single character except a carriage return.
6. ^, $ are special characters used as anchors. The '^' matches the start of line and '$' at the end of the lines. \b matches both boundaries.
7. | is used for disjunction.
8. *,+,?,{} are used as counters.
9. *,\.,\?,\n,\t, \s are used to match asterisk, a period, a question mark, a new line, a tab, and white space respectively.

2.4 Segmentation

Properly, the work of Myanmar word segmentation is the operation of the most essential auxiliary task of natural language processing. The Myanmar script includes in the family of Brahmic, consists of 33 basic consonant letters. Each independent consonant letter can form a complete syllable by itself with the help of an inherent or instinctive vowel. The inherent vowel can be transformed to other vowels through using a variety of diacritic marks. Further diacritic marks contain dependent consonant marks and tone marks for syllable onset clusters, and a virama ("asat", which signify kill in Burmese) used to cover up the instinctive vowel of a consonant letter for nasal or glottal syllable. Again, similar as the syllabification process in Myanmar language, there is no certain standard rule. Therefore, a lot of researches have been taking out with various approach as described in follow. [2]

2.4.1 Dictionary Based Matching

Maximum matching by looking up in a predefined dictionary and matching the longest string from an input sentence is an ordinary word segmentation approach for other languages. The matching can be handled from the starting of a sentence to its ending and also in reversing. The former technique is indicated as forward maximum matching (FMM) and the latter as reverse (or backward) maximum matching (RMM). These two directional procedures can be joined to create a bi-directional maximum matching (BMM), in which further searching rules are used to choose the better result from between FMM and RMM. Although the approach to segmentation using maximum matching technique is simple, its performance can be inferior, and it is normally used as a standard approach in word segmentation works. Because the speed and simplicity of this typical approach is a high advantage, the approach is still used in practical engineering widely and has been researched in a lot of studies.

2.4.2 Machine Learning Approach

Word segmentation process can be regarded as a classification procedure in a machine learning framework. And more specifically, a sequence labeling task, according to the properties of textual data, and a lot of standard learning frameworks have been developed and established. Almost them, Support Vector Machine (SVM) in the KyTea toolkit and Conditional Random Fields (CRF) in the CRF++ toolkit are

the most popular toolkits for machine learning approaches for segmentation. Feature engineering for input and tag-set design for output are import issues for machine learning approaches.

2.4.3 Expectation Maximization Approach

Segmenting a raw character sequences with no boundary sentences into words have been researched in many unsupervised methods. Most recent approaches are based on using some form of expectation maximization (EM) to learn a probabilistic of sentences and then using Viterbi-decoding-like procedures to segment input raw sentences into words. One of the reasons that expectation maximization approach is extensively adopted for unsupervised learning is that it is assured to be a good probabilistic model that locally maximizes the posterior or likelihood probability of the training corpus. For the issue of word segmentation, expectation maximization model is generally used by extracting a set of words from a given training data, initializing a probability distribution over this set of words, and then applying the iteration to adjust the probabilities of the set of words to gain the higher posterior probability of the training corpus.

2.4.4 Statistical Approach

The approach of statistical language model for word segmentation is more acceptable and reasonable than dictionary based matching, because it adapts with the probabilities of a sequence of words in real life textual data. Therefore, segmentation results consist of more common words which are higher than containing ambiguous words. Additionally, statistical approaches require enormous amount of training data than a dictionary-based approach. Among a variety of statistical language model, for example, an N-gram approach, the word segmentation procedure becomes a scanning task to regulate the segmentation with the highest or longest matching probability.. One of the simple methods such as Viterbi-like dynamic programming algorithm can be used by searching an input sentence to produce the best segmentation with each syllable until the best segmentation of the entire sentence is built up. Other N-gram language models are the maximum-likelihood approximated uni-gram model, the absolute discounting uni-gram model, and the adjustable discounting models such as uni, bi, and tri-gram models.

2.4.5 N-gram Approach

This N-gram approach is a string similarity approach that involves the system manipulating a measure of similarity between an input data and each of the definite words in the training corpus or predefined database. Those corpus or database data that have a great similarity to an input data are then displayed to the user for possible inclusion in the input. N-gram matching approach is one of the most popular approaches for segmentation. An n-gram is a set of 'n' successive characters extracted from a word. The main concept behind this approach is that, similar words will have a higher proportion of n-grams in general. Typical values for n are two or three, these corresponding to the use of di-grams or trigrams, respectively. [11] There is (n+1) di-grams and (n+2) trigrams in a word contained according to 'n' characters. It is depend on the research work that how much the 'n' will be defined. For example, the word "COMPUTER", the results in the computation of the di-grams

C, CO, OM, MP, PU, UT, TE, ER, R

and the trigrams

C, *CO, COM, OMP, MPU, PUT, UTE, TER, ER*, R

where '*' stands for a padding space.

2.5 Stemming

Stemming is the development of reducing distinct grammatical forms or word forms of a word such as its noun, verb, adjective, adverb etc. to its original or minimum root form. For example, in English language, the stem of "works" is "work" since -s is an inflectional suffix. The algorithm or model that is used for stemming is called stemmer. Stemmers are mainly based on three approaches: rule-based, statistical, and hybrid. [15]

Rule based stemmers, as the name denoted, extract the stem forms by using manually defined rules. The earliest established stemmers are using rule-based method. One of the oldest statistical stemmers is developed by Xu and Croft. Their technique developed the use of words occurs to deal with words grouped in correlated classes that are constructed by intrusive stemming. Hybrid stemmers are developed by combining different methods in a single unit of model. In 2010, Popat et al. propose a hybrid stemmer that combines statistical and rule-based models.

There are several types of stemming algorithms which are mentioned below; each of these groups has a typical way of finding the stems of the word variations. [5]

2.5.1 Brute Force Algorithms

In Brute force stemmers, a lexicon which encompasses a relation between inflected form and root form. To perform stemming, the lexicon is interrogated to find an identical inflection. If an identical inflection is found, the correlated root form is returned as output.

2.5.2 Suffix Stripping Algorithms

Suffix stripping algorithms occupied on a typically short list of "rules" stored in a lexicon or an array. This list brings a passage for the algorithm to detect the root form of a disposed input word. The stem word is generated by stripping the suffix from the input word. [10]

2.5.3 Lemmatization Algorithms

This lemmatization process consists of determining the part of speech category (e.g.: noun, verb, adjective, adverb etc.) of the input, and processing different normalization rules for each part of speech category. The part of speech is found out in the very first step and in later the stemming rules switch depending on the part of speech of the word according to its category.

2.5.4 Stochastic Algorithms

In this approach, the machine needs to train the inflected word along with root word firstly and defines some probabilistic internal set of rules. Based on these internal rules, the algorithm detects the most possible stem word from an input word. In most of the research works, the stemmer using this approach eliminates the affixes from the input word. [9]

2.5.5 Affix Stemmers

In linguistics world, the term affix specifies to prefix, suffix and infix. This can be regard as an extension to working with only suffix stripping method. It depends on the nature of the language whether suffix stripping is enough or not to generate the stem

word. If proposed stemmer can strip prefix, infix and as well as suffix then these stemmers can be labeled as affix stemmer.[1]

2.5.6 Matching Algorithms

This kind of algorithms applies a stem database which contains a tons of stem words. These stems are not require to be valid words but rather common sub strings. In order to stem a word, the algorithm is running to scan the input word with its matching stems from the database, integrated with various filters or rules, such as on the corresponding length of the applicant stem within the word. This method is sometime called as table lookup method.

2.5.7 Hybrid Approaches

By combining any two or more of the approaches described above. For example, one hybrid stemmer can use a suffix stripping algorithm which first considers a lookup table using brute force technique.

2.5.8 Stemming Errors

There are two particular types of error when stemming is used for language processing. The first one is under-stemming errors, in which words which refer to the same concept are not reduced to the same stem which can also called as a false negative. This is an error where word should be stemmed to the root, but are not. The second type of error is over-stemming errors, in which words are stemmed to the same stem even though they refer to distinct concepts. This kind of error is known as a false positive. In designing a stemming algorithm there is a trade-off between these two kinds of error. A light stemmer plays safe in order to avoid over-stemming errors, but consequently leaves many under-stemming errors. A heavy stemmer boldly removes all sorts of endings, some of which are decidedly unsafe, and therefore commits many over-stemming errors. [3]

2.6 Part of Speech Tagging

Part of speech (POS) tagging is the process of assigning particular word in sentences with a specific tag that describes how the tagged word means in the sentences. That indicates POS tagging allows whether a tagged word is used as a noun, verb, adjective, etc. A POS tagger attempts to attach the corresponding POS tag to each word in sentences, taking into account the context in which this word appears. A different number of technique such as rule-based tagging, memory-based tagging, transformation- based tagging, maximum entropy tagging and probabilistic tagging can be used for POS tagging. Numbers of tag set and tagged keyword definition may be different from one another. One of the biggest problem or challenge in POS tagging is word disambiguation. That is the exact identical word that can be absolutely different part of speech or meaning, such as one is a verb and other is a noun. Elementary word sense disambiguation is achievable if the researcher can tag words with their basic POS tags. Word-sense disambiguation (WSD) is describing which sense of a word is played in a sentence, when the word has multiple meanings. All of the POS tagging algorithms can be classified into two groups: rule-based POS Taggers and stochastic POS Taggers. Typical rule-based POS tagging approaches use contextual information to assign tags to unknown words which is also called ambiguous word. Generally, disambiguation is performed by analyzing the linguistic features of the word, its previous word, its successive word, and other aspects. The stochastic POS tagger can refer to any number of different approaches to the problem of POS tagging. Any model which somehow applying probability or incorporates frequency may be properly called stochastic approach. According to the requirement of the research, the level of POS tagging can also be differentiate as basic POS tagging, standard POS tagging and specific POS tagging.

2.6.1 Basic Part of Speech Tagging

Because of the data insufficiency, most of the researchers use stem words in the lexicon for POS tagging. According to Myanmar dictionary books and grammar books, there are nine Part of Speech tags in Myanmar language. Basically, we have defined each word with these nine basic POS tags and we have built a POS tagged corpus in which every word is tagged with corresponding basic POS tags. Furthermore, a corpus which contains stem words with basic POS tag-set that can be used for tagging the input

untagged words with all possible tags. As shown in table 2.1, there are basic nine Part-of-Speech of Myanmar language namely noun, pronoun, adjective, adverb, verb, conjunction, particles, postpositional marker and interjection. The number of basic POS can be differed from one language to another. [14]

Table 2.1 Basic Part of Speech Tags

No.	POS Description	Example
1.	Noun	ယောကျ်ားလေး (boy)
2.	Pronoun	သူ (he)
3.	Adjective	ကောင်း (good)
4.	Adverb	အလွန် (very)
5.	Verb	ပြေး (run)
6.	Conjunction	နှင့် (and)
7.	Particle	သော (that)
8.	Postpositional Marker	သို့ (to)
9.	Interjection	ဟယ် (wow)

2.6.2 Standard Part of Speech Tagging

The basic POS tagging is defined as the tag in the lexicon is attached to a stem word. Although many of the basic POS tagging cannot cope with all words which are constructed from the combination of two or more stem words. In order to perform the direct translation from one language to another, the POS tags of the combined words are must know. It is difficult and even unable to do the translation process with only basic tags. That is the reason of standard POS tagging is required than basic POS tagging. On the other hand of disambiguation, standard POS tags need to be applied to tag to the combinations of words if the patterns of combination are matched. [14]

Table 2.2 Standard Part of Speech Tags

No.	Description	Example
1.	Plural Noun	လူများ (people)
2.	Plural Pronoun	သူတို့ (they)
3.	Comparative Adjective	ပိုကောင်း (better)
4.	Superlative Adjective	အကောင်းဆုံး (best)
5.	Comparative Adverb	ပိုမြန် (faster)
6.	Superlative Adverb	အမြန်ဆုံး (fastest)
7.	Negative Verb	မလုပ် (not do)
8.	Negative Adjective	မကောင်း (not good)
9.	Noun (Adjective Convert)	အကောင်း (goodness)
10.	Noun (Verb Convert)	ပြုလုပ်ခြင်း (doing)
11.	Adverb (Verb Convert)	လျင်လျင်မြန်မြန် (quickly)
12.	Adverb (Adjective Convert)	ပျော်ပျော်ပါးပါး (happily)

2.6.3 Specific Part of Speech Tagging

There have more than one subclass of specific category in each of the basic POS tag. For example, the word "boy" is tagged as a noun in basic and it is also classified as a person in specific. Therefore, in order to build a specific part of speech tagging, we have to categorize every word with specific class and general class. The basic classes of basic POS classes have nine tag and the specific classes are required to describe the detail of each POS class. To construct a well working specific part of speech tag, it would require to work with language professionals and linguistic professionals.

Table 2.3 Specific Part of Speech Tags

No.	POS Name	Category Name	
1.	Noun	• Common • Personal • Animals • Time • Body	• Building • Object • Location • Cognition • Attribute
2.	Pronoun	• Personal • Subjective • Objective • Possessive • Indefinite	• Relative • Intensive • Demonstrative • Interrogative • Reflexive
3.	Adjective	• Demonstrative • Quantity • Question	• Place • Time • Object
4.	Adverb	• Time • Manner • State	• Quantity • Question
5.	Verb	• Common	• Compound
6.	Conjunction	• Sentence • Mean	• Chunk
7.	Particle	• Type • Common • Number • Support	• Interjection • Negative • Quantity • Example
8.	Postpositional Marker	• Subject • Object • Leave • Direction • Arrive • Used • Cause	• Accept • Place • Time • Agree • Extract • Possessive • Time Start/End
9.	Interjection	• Greeting • Joy • Attention	• Surprise • Sorrow

2.7 Performance Measure

Performance measure is an essential step for every research work and it is the process of collecting, organizing, analyzing and reporting data or information regarding the performance of individual, group, organization, component or system. One of the most important things needed to be considered in the performance measure process is that the performance measures is working based on both qualitatively and quantitatively to provide the helpful and useful information about unit work, product, process, component and system. Implementation of performance measure is the best way to understand, manage and improve the result made by a certain group, organization, component and system.

2.7.1 Accuracy

Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if a system has high accuracy then that system is best. The accuracy is a great measure but only when the system has symmetric datasets where values of false positive and false negatives are almost same. Equation 2.3 is for calculating accuracy.

$$\text{Accuracy} = (\text{TP}+\text{TN})/(\text{TP}+\text{FP}+\text{FN}+\text{TN}) \quad 2.3$$

where,

TP = True Positive

TN = True Negative

FP = False Positive

FN = False Negative

True positive and true negatives are the result that are correctly processed. False positive and false negatives, these values happen when actual result contradicts with the predicted result.

True Positives (TP) – These are the correctly predicted positive values which means that the value of actual result is ‘yes’ and the value of predicted result is also ‘yes’. For example, if actual result value indicates that this student passed the exam and predicted result tells the same thing.

True Negatives (TN) – These are the correctly predicted negative values which means that the value of actual result is ‘no’ and value of predicted result is also ‘no’. For example, if actual result says this student did not pass the exam and predicted result tells the same thing.

False Positives (FP) – When actual result is ‘no’ and predicted result is ‘yes’. For example, if actual result shows this student did not pass the exam but predicted result tells that this student will pass the exam.

False Negatives (FN) – When actual result is ‘yes’ but predicted result in ‘no’. For example, if actual result valued points out that this student passed the exam and predicted result tells that this student will fail.

2.7.2 Precision, Recall and F1 Score

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question is that in this metric answer of all students which labeled as passed, how many students actually passed. This high prevision relates to the false positive rate as described in equation 2.4.

$$\text{Precision} = \text{TP}/\text{TP} + \text{FP} \qquad 2.4$$

Recall is the ratio of correctly predicted positive observations to the all observations in actual result. The answer for recall is that of all the students that truly passed, how many did label. This recall is described in equation 2.5.

$$\text{Recall} = \text{TP}/\text{TP} + \text{FN} \qquad 2.5$$

F1 Score is the weighted average of Prevision and Recall. Therefore, this score takes both false positives and false negatives into account. Clearly, it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it’s better to look at both Precision and Recall. The equation for calculating F1 Score is described in equation 2.6.

$$\text{F1 Score} = 2*(\text{Recall} * \text{Precision}) \,/\, (\text{Recall} + \text{Precision}) \qquad 2.6$$

CHAPTER 3

DESIGN OF THE PROPOSED SYSTEM

The system is designed for stemming and POS tagging on Myanmar Language. In this system, user can also segment Myanmar word from Myanmar sentences as line by line paragraph or as a whole paragraph by paragraph with or without space. The output result of this system can be utilized in area such as information retrieval and search engine optimization.

3.1 Overview of the System Design

In this system, starting from normalization, syllabification and segmentation as preprocessing steps and followed by stemming and POS tagging respectively. Figure 3.1 shows the overview of the system design. After the system received the input sentences, the normalization step starts processing. It uses its normalizations rules to work on input sentences. After this step, Myanmar syllable pattern rule is used to break down the sentences. In segmentation step, N-gram matching algorithm is applied using prefix, suffix lexicon and the step of stemming is executed by rule based stemmer using prefix, infix and suffix lexicon. The final POS tagging step work with grammatical rules. After all these steps is completed, the output result is generated.

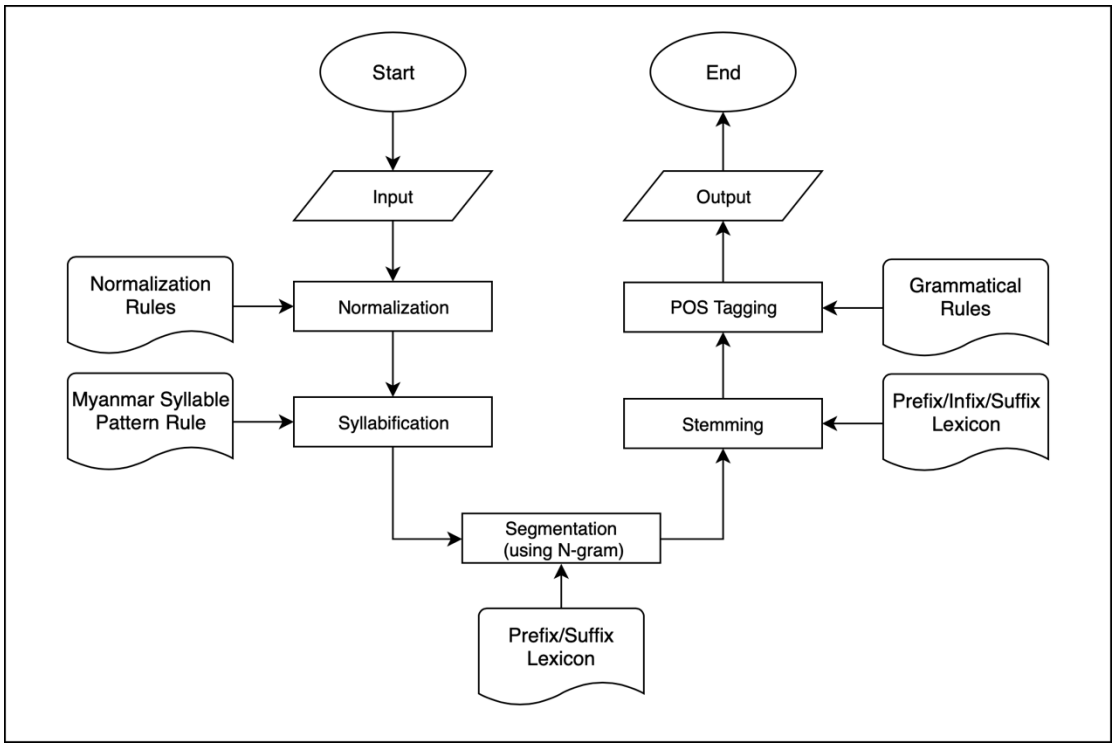

Figure 3.1 Overview of the System Design

Three types of corpus are predefined into consideration: prefix, infix and suffix. Words in the corpus are manually created. A set of hand written rules are also defined for stripping prefix and suffix especially for inflected and derived word. Infix detection algorithm is also included.

All of the predefined corpus is collected from Myanmar Grammar Book published by Myanmar Language Commission. For N-gram segmentation, corpus of 176 prefixes and 266 suffixes are predefined. For stemming and POS tagging, 196 prefixes, 30 infixes and 259 suffixes, totally 485 words are used in this algorithm as a corpus. The words containing in this corpus have various number of ranging from one syllable to five syllables. [13]

Before actually running the stemmer algorithm, it will be efficient to perform the pre-processing steps for the better outcome. The preprocessing steps will be done step by step, starting from the normalization and following by syllabification and segmentation.

3.2 Class Diagram of the System

There are six classes in this system: Main, Normalization, Syllabification, Segmentation, Stemming and POSTagging. The Main class includes finalResultList, finalTaggedList and checkValue as attributes and initUI, onNormalization, onSyllabification, onSegmentation, onStemming and onTagging as function. The Normalization class is composed of inputString and normalizedString as attributes and normalize as function. The third class is defined as Syllabification. This class contains attributes named consonant, enChar, otherChar, symbol, ngaThat, aThat and seperatorSymbol. The function called syllabification operates as function in this class. The next class is Segmentation class and it consists prefixList, suffixList, rawText, matchText and resultText as the class attributes. The functions of this class are generateNgrams, callSegment and doSegmentation. The fourth class, Stemming class has rawData, resultList and checklist as attributes and generateNgrams, callStem and callRuleList perform as class function. The last class is POSTagging. It contains taggedList and tagList as class attributes and callTag serves as function for the class.

The figure 3.2 shows the class diagram of the overall system.

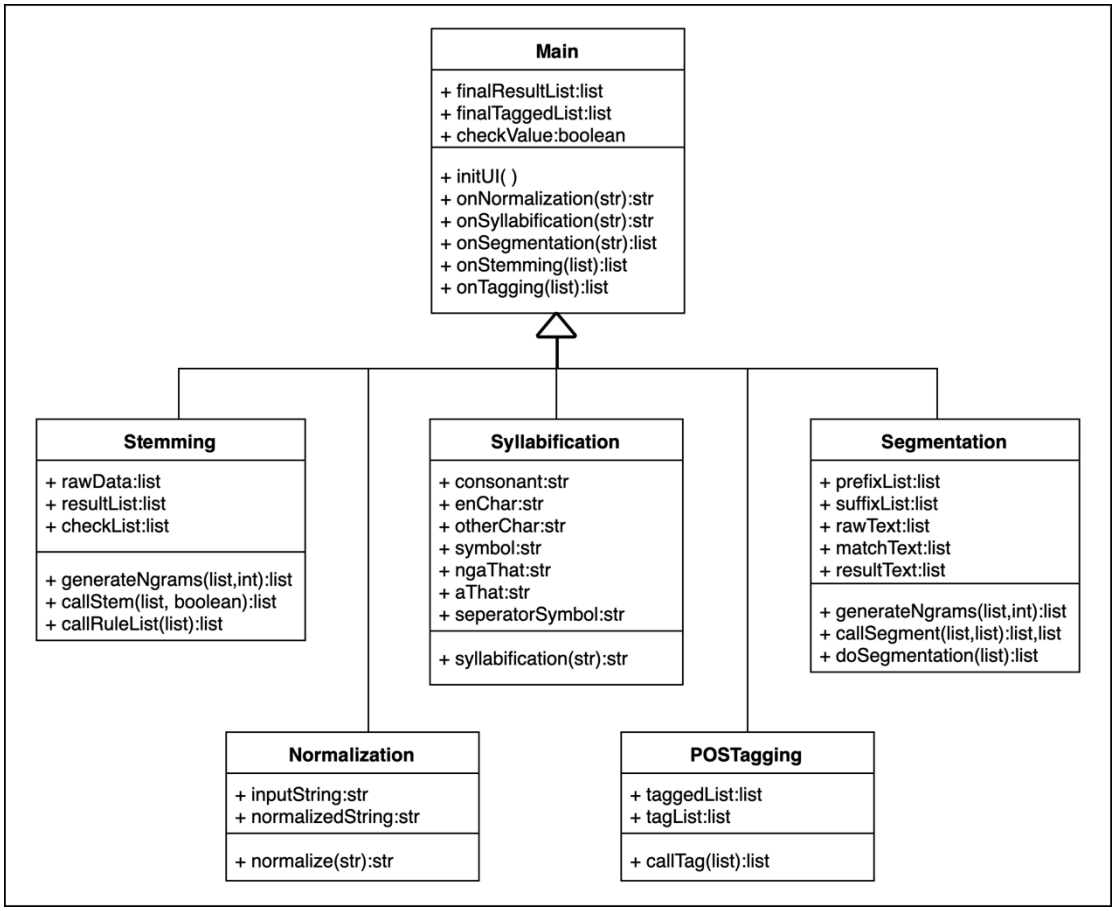

Figure 3.2 Class Diagram of the System

3.3 Procedure of the System

The system accepts Myanmar sentences as an input and the output results are stem words with POS tag. There are five main functions which perform step by step in order to complete the whole procedure of the system. These five functions are Normalization, Syllabification, Segmentation, Stemming and POS Tagging.

In the latter explanations, the following sentence is used as an example. ‘သူသည်ပင်ပန်းသောကြောင့်ကျောင်းကိုကားဖြင့်သွားသည်။’ In English meaning, it is translated as “He goes to school by car because of being tired.”

သူ(he) သည်(null) ပင်ပန်း(being tired) သောကြောင့်(because of) ကျောင်း(school) ကို(to) ကား(car) ဖြင့်(by) သွား(goes) သည်(null) ။ (.)

3.3.1 Normalization

The very first step to the system is to polish the input sentence. Normalization is a process that converts a list of words to a more uniform sequence. By normalizing the words to a standard format, the matching process and searching process will simplify. Normalization in Myanmar word is applied because the input may contain the converted text from different encoding and typing error. In this study, 40 rules are using for normalizing character reordering. One of the rules is explained as an example.

E.g. In 'ပုံ' word, correct from 'ပ+ံ+ု' to 'ပ+ု+ံ'

E.g. In 'ဖြင့်' word, correct from 'ဖ+ြ+င+္+်' to 'ဖ+ြ+င+်+္'

3.3.2 Syllabification

After the normalization, the next step is syllabification. Syllabification is the task of breaking a sequence of words into syllables. Syllabification has no completely agreed definition of syllable boundaries. In this study, the following syllable structure of Burmese is used, C(G)V((V)C), which is also to say that the consonant is optionally followed by a glide. Only one monophthong is consisted in the rime.

CV / သူ C=သ, V=ူ

CVC / မိန်း C=မ, V=ိ, C=န်း

CGV / မြေ C=မ, G=ြ, V=ေ

CGVC / မြိန် C=မ, G=ြ, V=ိ, C=န်

CVVC / မောင် C=မ, V=ေ, V=ာ, C=င်

CGVVC /မြောင်း C=မ, G=ြ, V=ေ, V=ာ, C=င်း

The output gives syllables with separated space character.

Input =

သူသည်ပင်ပန်းသောကြောင့်ကျောင်းကိုကားဖြင့်သွားသည်။ (no space contained)

Output = သူ သည် ပင် ပန်း သော ကြောင့် ကျောင်း ကို ကား ဖြင့် သွား သည် ။ (space is added within each syllable)

3.3.3 Segmentation

Word segmentation is an essential first step in processing languages. Among the various approaches, in this study, N-gram matching techniques is used for segmentation. N-gram is a set of n consecutive characters extracted from a word. Typical values for n are 2 or 3, these corresponding to the use of di-grams or tri-grams,

respectively. In this study, five-grams are used for matching the prefix/suffix and output the segmented sentence. By matching with the predefined prefix and suffix lexicons, the input sentence is segmented with the correct lexicon and result as a space separated sentence.

Input = သူသည်ပင်ပန်းသောကြောင့်ကျောင်းကိုကားဖြင့်သွားသည်။ (unsegmented sentence)

Five-grams set = [‘သူသည်ပင်ပန်းသော’, ‘သည်ပင်ပန်းသောကြောင့်’, ‘ပင်ပန်းသောကြောင့်ကျောင်း’, ‘ပန်းသောကြောင့်ကျောင်းကို’, ‘သောကြောင့်ကျောင်းကိုကား’, ‘ကြောင့်ကျောင်းကိုကားဖြင့်’, ‘ကျောင်းကိုကားဖြင့်သွား’, ‘ကိုကားဖြင့်သွားသည်’, ‘ကားဖြင့်သွားသည်။’]

Output = သူသည် ပင်ပန်းသောကြောင့် ကျောင်းကို ကားဖြင့် သွားသည်။

The following figure 3.3 shows how the segmentation process executed using N-gram.

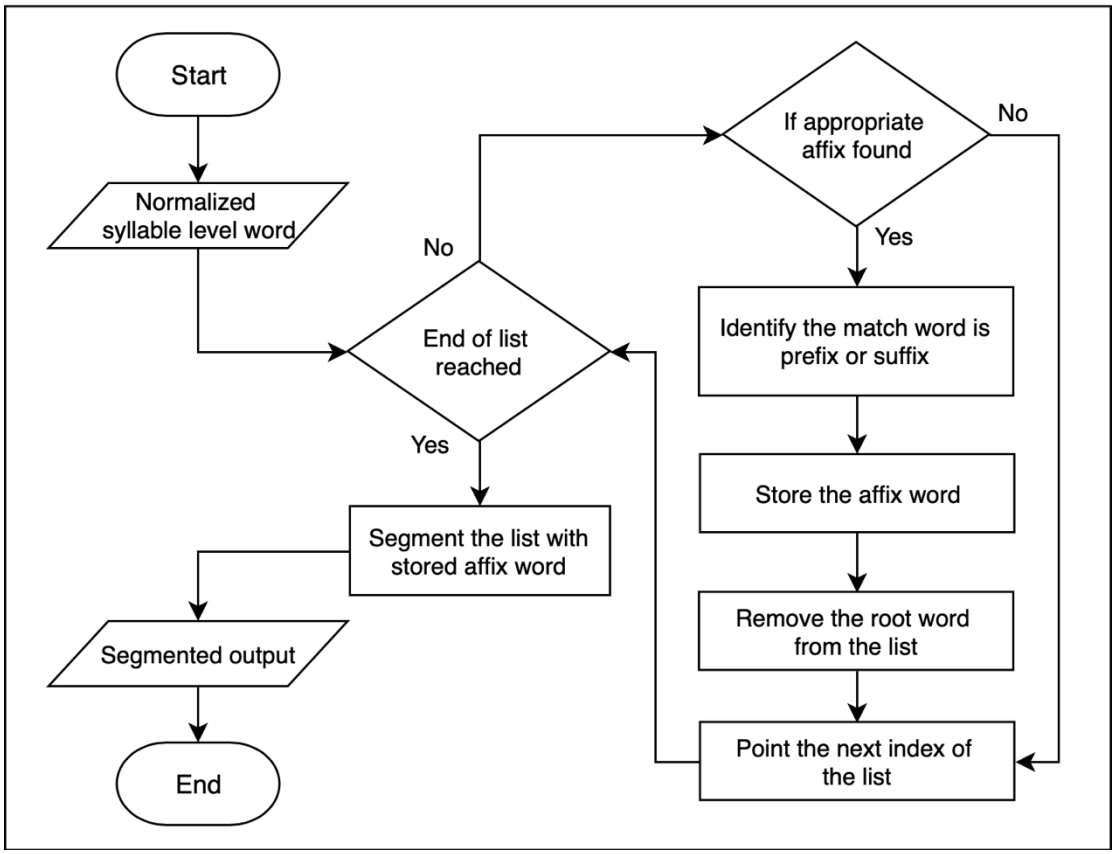

Figure 3.3 Flowchart of Segmentation using N-gram

3.3.4 Stemming

After processing all the above steps, the segmented words are ready to stem. In this system, the stemmer was developed using look up based approaching using three corpus prefix, infix and suffix. In this step, 8 kinds of grammar rules and 8 kinds of additional rules are applying for stemming. The eight kinds of grammar rule contain

conjunction rule, postpositional marker rule, particle rule, interjection rule, adjective rule, adverb rule, verb rule and pronoun rule. The list of additional rules includes symbols rule, gender rule, counter rule, number rule, number units rule, months rule, day rule and negative rule.

Grammar rules contain 37 rules for of Prefix/Infix/Suffix stemming. This stripping is done by matching the word with the input phrase. Additional rules are constructed by 12 rules. All the rules are using 485 predefined prefix/infix/suffix corpus for respectively. These segmented words are applied with the above rules sequentially. If the input word contains the word from corpus list, then it is considered as a root word. This process is continuing until it finds the root word, its prefix/suffix or till the word is stripped into minimum word length that is one. The algorithm does affix striping the word according to rule and tag with the rule indicator.

Input = သူသည် ပင်ပန်းသောကြောင့် ကျောင်းကို ကားဖြင့် သွားသည်။

Output = [‘သူ’, ‘ပင်ပန်း’, ‘ကျောင်း’, ‘ကား’, ‘သွား’]

The word ‘သည်’, ‘သောကြောင့်’, ‘ကို’, ‘ဖြင့်’, ‘သည်’, ‘။’ are removed as they are prefix and suffix to the stemmed word.

The figure 3.4 described in below indicates how the stemming and POS tagging work.

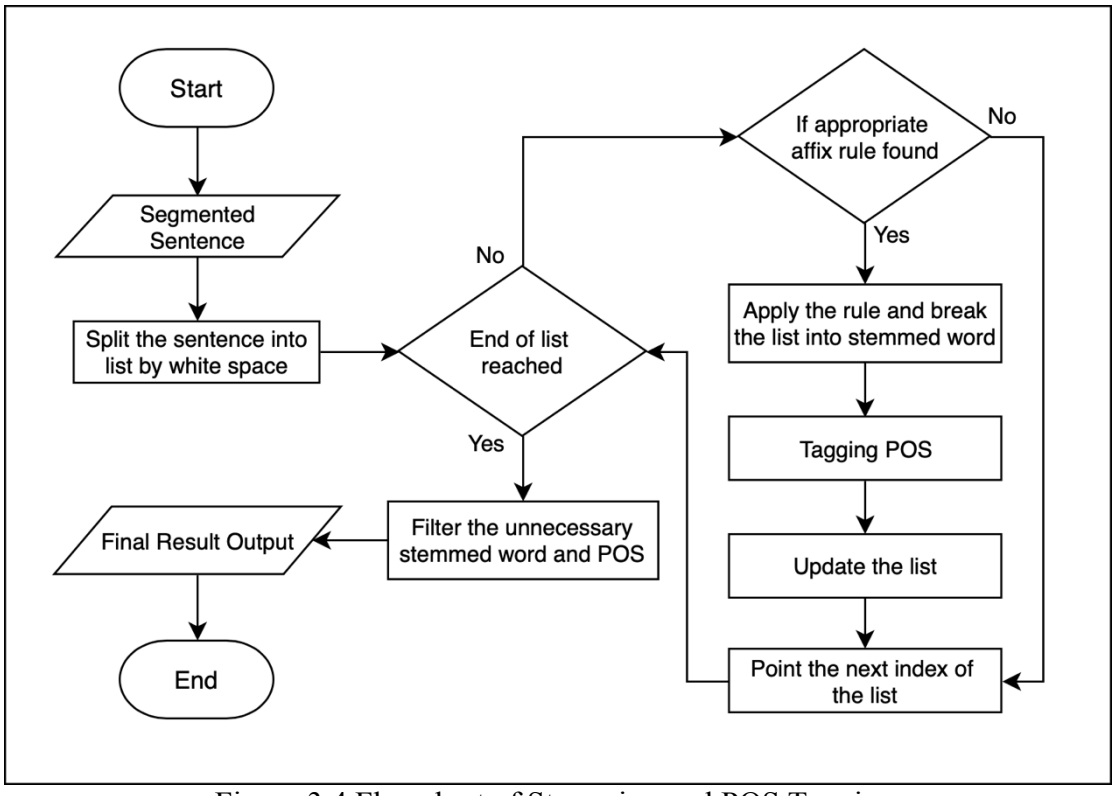

Figure 3.4 Flowchart of Stemming and POS Tagging

3.3.5 POS Tagging

In this stage, 14 kinds of tag-sets are tagged according to the rule of stemming for POS tagging. They are Nouns (NN), Pronouns (PRON), Verbs (VB), Adverbs (ADV), Adjectives (ADJ), Conjunctions (CONJ), Particles (PART), Postpositional Markers (PPM), Interjections (INTJ), Symbols (SB), Numbers (NUM), Text Number (TN), Punctuation (PUNC) and Foreign Word (FW). Addition to the above tag-sets, there is also an ambiguous POS corpus. Since the POS tagging is working under the same framework with stemming algorithm, the POS tag is just attached to the word according to rule indicator. In this system, not only the stemmed word but also the stripped word is tagged and stored in a separated file. The stemmed word only is defined as output (1) and the whole sentences which include stripped word is described as output (2).

Input = သူသည်ပင်ပန်းသောကြောင့်ကျောင်းကိုကားဖြင့်သွားသည်။

Output 1 = [‘သူ-PRON’, ‘ပင်ပန်း-VB’, ‘ကျောင်း-NN’, ‘ကား-NN’, ‘သွား-VB’]

Output 2 = [‘သူ-PRON’, ‘သည်-PRON/PPM’, ‘ပင်ပန်း-VB’, ‘သောကြောင့်-CONJ’, ‘ကျောင်း-NN’, ‘ကို-PPM’, ‘ကား-NN’, ‘ဖြင့်-PPM’, ‘သွား-VB’, ‘သည်-PPM’, ‘။-PUNC’]

CHAPTER 4

IMPLEMENTATION OF THE PROPOSED SYSTEM

This chapter presents how the system is implemented. There are five main algorithms used in the system. They are normalization algorithm for text cleaning and normalization work, syllabification algorithm for syllabification, segmentation algorithm using N-grams for segmenting the input sentences, stemming algorithm for stemming the segmented words into stemmed word and POS tagging algorithm for tagging POS into stemmed word. The whole program are written in Python language with version 3.6.1.

4.1 Normalization Algorithm

In normalization algorithm, the input is raw Myanmar text, this step reorders, removes and replaces incorrect input and the output is the normalized Myanmar text which has been applied by 38 rules. From line number 12 to 19 is to clean the consecutive word. Line number 26 to 31 is to replace with letter Wa and number zero in Myanmar words which are very similar and usually typed by mistake. Another similar common mistake in typing is letter Ya and number seven which are corrected by rules described in line number 32 to 37.

1. function normalization(text)

Input : raw text

Output : normalized text

Process :

2. **begin**

3. normalizedString = rawText

4. reorder ([ါ-ှ]) and ([ျ-ြ])

5. reorder (ံ)and (ု)

6. reorder ([က-ဥ])(ွ)(ေ)(ြ) to ([က-ဥ])(ွ)(ြ)(ေ)

7. reorder (်) and (့)

8. reorder (ျ) and (ြ)

9. reorder (ု) and (ာ)

10. reorder (ု) and (ူ)

11. reorder (ံ) and (ွ)

12. let DoubleWord = normalizedString containing (ါ-သ)

13. i = 0

14. while i < length(normalizedString):

15. match = search DoubleWord in normalizedString

16. if match:

17. replace matched word in normalizedString

18. else:

19. i += 1

20. remove double or more (ိ)

21. remove double or more (ီ)

22. reorder ([က-အ])(ူ)(ီ) to ([က-အ])(ီ)(ူ)

23. replace ([က-အ])(ူ)(ီ) to ([က-အ])(ီ)(ူ)

24. reorder (ူ) and (ီ)

25. reorder ([က-အ])(ု)(ေ)(ျ) to ([က-အ])(ျ)(ု)(ေ)

26. replace (ဝ) in matching with (ဝီ)(?! ?/) pattern

27. replace (ဝ) in matching with ([^ဝ-ဪ])ဝ([^ဝ-ဪ]|[။]) pattern

28. replace (ဝ) in matching with (ဝ)([ါ-ဲ]) pattern

29. replace (ဝ) in matching with (ဝ)(ံ) pattern

30. replace (ဝ) in matching with (ဝ)(ိ) pattern

31. replace (ဝ) in matching with (ဝ)(ု) pattern

32. replace (ရ) in matching with (ဂ)([က-လ သ-ဩ ာ ီ-သ ၌-၏]) pattern

33. replace (ရ) in matching with (ေ◌ါ) pattern

34. replace (ရ) in matching with (◌ါ)([ိ-ဲ]) pattern

35. replace (ရ) in matching with (◌ါ)(◌ံ) pattern

36. replace (ရ) in matching with (◌ါ)(◌ီ) pattern

37. replace (ရ) in matching with (◌ါ)(◌ု) pattern

38. reorder ([က-အ])(◌ံ)(◌ွ)(◌ု) to ([က-အ])(◌ွ)(◌ံ)(◌ု)

39. reorder ([က-အ])(◌ီ)(◌ု)([က-အ]) to ([က-အ])([က-အ])(◌ီ)(◌ု)

40. reorder ([က-အ])(◌ံ)(◌ု) to ([က-အ])(◌ု)(◌ံ)

41. reorder (ငြ)(ေ◌)(◌ီ)(◌ု)([က-အ]) to (ငြ)(◌ီ)(◌ု)([က-အ])(ေ◌)

42. replace (ဥ) and (ဥ)

43. reorder ([က-အ])(ေ◌)(◌ွ) to ([က-အ])(◌ွ)(ေ◌)

44. reorder ([က-အ])(ေ◌)(◌ု)(◌ျ) to ([က-အ])(◌ု)(◌ျ)(ေ◌)

45. remove (\u200B) Zero Width Space

46. remove (\u200C) Zero Width Non-Joiner

47. remove (\u202C) Pop Directional Formatting

48. remove (\u00A0) No-Break Space

49. return normalizedString

50. **end**

Figure 4.1 Algorithm for Normalization

4.2 Syllabification Algorithm

In this algorithm, Myanmar consonant, English characters and other special character are defined. And the input normalized sentence are breaking into syllable level by using the Myanmar syllable pattern which is defined as ((?<!Virama) [Consonant](?![aThat|Virama])|[engChar|otherChar]) pattern.

```
1.  function syllabification(text)
Input      : normalized text
Output     : syllable word
Process    :
2.    begin
3.        Consonant = r"က-အ"
4.        engChar = r"a-zA-Z0-9"
5.        otherChar = r"ဣဤဥဦဧဩဪသၐၑၒၓ၏ ၀-၉။!-/:-@[-`{-~"
6.        Virama = r'္'
7.        aThat = r'်ှ'
8.        seperator = " "
9.        remove all the unnecessary space from the input text
10.       break input text into syllable level using
11.           ((?<!Virama)[Consonant](?![aThat|Virama])|[engChar|otherChar])
12.       seperate each syllable using seperator symbol
13.       return output
14.   end
```

Figure 4.2 Algorithm for Syllabification

4.3 Segmentation Algorithm

In this segmentation algorithm, it contains three parts of function. The function described in figure 4.3 is to generate a set of five grams word from given sentences. The second function is named tagPreSuf function and it is showed in figure 4.4. This function is to tag the input segment of word as prefix or suffix and return the tagged list. The last function is main function which accepts the input sentences and returns the segmented word list. The algorithm for this function is described in figure 4.5.

```
1.  function generate_ngrams (wordslist, int)
Input      : text from segmentation function
Output     : text list with five gram
Process    :
2.    begin
3.        create empty ngrams_list
4.        for num in range(0, len(wordslist)):
```

```
5.                  increament the index of word and trim by five ngrams
6.                  join the trimmed word with space
7.                  store each list into ngrams_list
8.          return ngrams_list
9.    end
```

Figure 4.3 Algorithm for Generating N-grams

```
1.  function tagPreSuf (input_list)
Input       : five gram list
Output      : text list with prefix and suffix tag
Process     :
2.    begin
3.          get prefix data
4.          get suffix data
5.          for num in range(0, len(input_list)):
6.                  match input_list with prefix rule
7.                  if match:
8.                          store the match data in templist with prefix tag
9.                  match input_list with suffix rule
10.                 if match:
11.                         store the match data in templist with suffix tag
12.         return templist
13.   end
```

Figure 4.4 Algorithm for Tagging Prefix and Suffix

```
1.  function segmentation(text)
Input       : normalized syllable word
Output      : segmented word list
Process     :
2.    begin
3.          create firstlevellist and secondlevellist
4.          while indexa < len(inputtext):
```

```
5.                  call function generate_ngrams(firstlevellist)
6.                  store return value into secondlevellist
7.                  call function tagPreSuf(secondlevellist)
8.                  compare return list with prefix and suffix
9.                  if match:
10.                         store the match words in resultlist
11.                         delete stored words from secondlevellist
12.                 else:
13.                         reset the index of secondlevellist
14.         return resultlist
15.  end
```

Figure 4.5 Algorithm for Segmentation

4.4 Stemming Algorithm

This algorithm is the main part of the system. Segmented words list are accepted from the previous algorithm and the final output will be the stemmed word list. Basically, in this main algorithm, the sub functions are called for the specific rule in order to stem. There are altogether 31 rules, so that, there are 31 functions which have to work sequentially.

```
1.  function stemming(resultlist)
Input      : segmented words list
Output     : stemmed words list
Process    :
2.    begin
3.        resultlist = call_ForeignWord_Infix_rule(resultlist)
4.        resultlist = call_Exception_Word_rule(resultlist)
5.        for i in range(0, 1):
6.            resultlist = call_Grammar_Adj_Infix_rule(resultlist)
7.            resultlist = call_Grammar_Adv_Infix_rule(resultlist)
8.            resultlist = call_Grammar_Adv_Prefix_rule(resultlist)
9.            resultlist = call_Grammar_Negative_Infix_rule(resultlist)
10.           resultlist = call_Count_Prefix_rule(resultlist)
```

```
11.             resultlist = call_Months_Prefix_rule(resultlist)
12.             resultlist = call_Days_Prefix_rule(resultlist)
13.             resultlist = call_Grammar_Verb_Suffix_rule(resultlist)
14.             resultlist = call_Grammar_Conj_Prefix_rule(resultlist)
15.             resultlist = call_Grammar_Conj_Suffix_rule(resultlist)
16.             resultlist = call_Grammar_PPM_Suffix_rule(resultlist)
17.             resultlist = call_Grammar_Part_Suffix_rule(resultlist)
18.             resultlist = call_Grammar_Interj_Prefix_rule(resultlist)
19.             resultlist = call_Grammar_Interj_Suffix_rule(resultlist)
20.             resultlist = call_Grammar_Adj_Prefix_rule(resultlist)
21.             resultlist = call_Grammar_Adj_Suffix_rule(resultlist)
22.             resultlist = call_Grammar_Adv_Prefix_rule(resultlist)
23.             resultlist = call_Grammar_Pron_Preffix_rule(resultlist)
24.             resultlist = call_Grammar_Adj_Suffix_rule(resultlist)
25.             resultlist = call_Grammar_Adv_Suffix_rule(resultlist)
26.             resultlist = call_Grammar_Duplicate_Prefix_rule(resultlist)
27.             resultlist = call_Grammar_Duplicate_Suffix_rule(resultlist)
28.             resultlist = call_Gender_Prefix_rule(resultlist)
29.             resultlist = call_Gender_Suffix_rule(resultlist)
30.             resultlist = call_Number_Infix_rule(resultlist)
31.             resultlist = call_NumberUnit_Suffix_rule(resultlist)
32.             resultlist = call_Symbol_Suffix_rule(resultlist)
33.             resultlist = call_Symbol_Infix_rule(resultlist)
34.         resultlist = DoubleWordStem.call_DoubleWordStem(resultlist)
35.         eliminate all space to show in clear format after stemming finished
36.         return resultlist
37.     end
```

Figure 4.6 Algorithm for Stemming

The following algorithm is one of the rule functions from the main stemming algorithm. In this function, line number 19 and 23 point out that the tagging process is working along with stemming algorithm.

```
1.  function call_Grammar_Verb_Suffix_rule (resultlist)
Input       : segmented words list
Output      : stemmed words list which applied by the specific rules
Process     :
2.      begin
3.          listindex = 0
4.          while listindex < len(resultlist):
5.              match with Grammar Verb Suffix Rule pattern
6.              store the matching pattern in match1 variable
7.              if listindex < len(resultlist) - 1:
8.                  temptext = resultlist[listindex + 1]
9.              else:
10.                 temptext = ""
11.             if match1:
12.                 checker = True
13.                 while checker:
14.                     match again with Grammar Verb Suffix Rule pattern
15.                     store the matching pattern in match2 variable
16.                     if match2:
17.                         if len(match2.group(1)) > 0:
18.                             store match2.group(1) value into listindex of resultlist
19.                             tag the applied rule keyword into the list.
20.                             listindex += 1
21.                         if len(match2.group(2)) > 0:
22.                             store match2.group(2) value into listindex of resultlist
23.                             tag the applied rule keyword into the list.
24.                             listindex += 1
25.                         remove the original word from the resultlist
26.                         listindex -= 2
27.                     else:
28.                         checker = False
29.                         if temptext != "":
30.                             listindex = resultlist.index(temptext)
```

```
31.                     else:
32.                         listindex = len(resultlist)
33.             else:
34.                 listindex += 1
35.         return resultlist
36.     end
```

Figure 4.7 Algorithm for Grammar Verb Suffix Rule

4.5 POS tagging Algorithm

This POS tagging is the final algorithm of the system. Since the POS tagging is working under the same framework with stemming algorithm. The POS tag is attached to the word according to rule indicator. In this system, not only the stemmed word but also the stripped word is tagged and stored in a separated file.

```
1.  function postagging(list)
Input       : stemmed words list
Output      : POS tagged words list
Process     :
2.      begin
3.          for i in range(len(resultlist))
4.              check the tag code for each word
5.              store the stemmed word and tag set in taggedlist
6.          for i in range(len(taggedlist))
7.              check the ambiguous tag code
8.              match the ambiguous word
9.              tag the ambiguous tagset
10.         for i in range(len(taggedlist))
11.             clean the unneccessary keyword from the taggedlist
12.             write taggedlist in textfile which only contain stemmed words.
13.         return taggedlist
14.     end
```

Figure 4.8 Algorithm for POS Tagging

4.6 Implementation of the System

The system is implemented using python language under the 3.6.1 version and PyCharm 2018.1 Professional Edition is used as an Integrated Development Environment(IDE). This system can accept Myanmar fonts encoded in Unicode Standard. The testing data are extracted from the various categories of Myanmar Wikipedia website, Myanmar Computer Federation (MCF) dataset and Myanmar Grammar Book published by Myanmar Language Commission which are typed in Myanmar3 font. The input text can be used as line by line format or one paragraph format. The main page can be shown in figure 4.9. It contains input text area, output result text area and five executable buttons for five main functions. The data can be processed step by step by clicking each button.

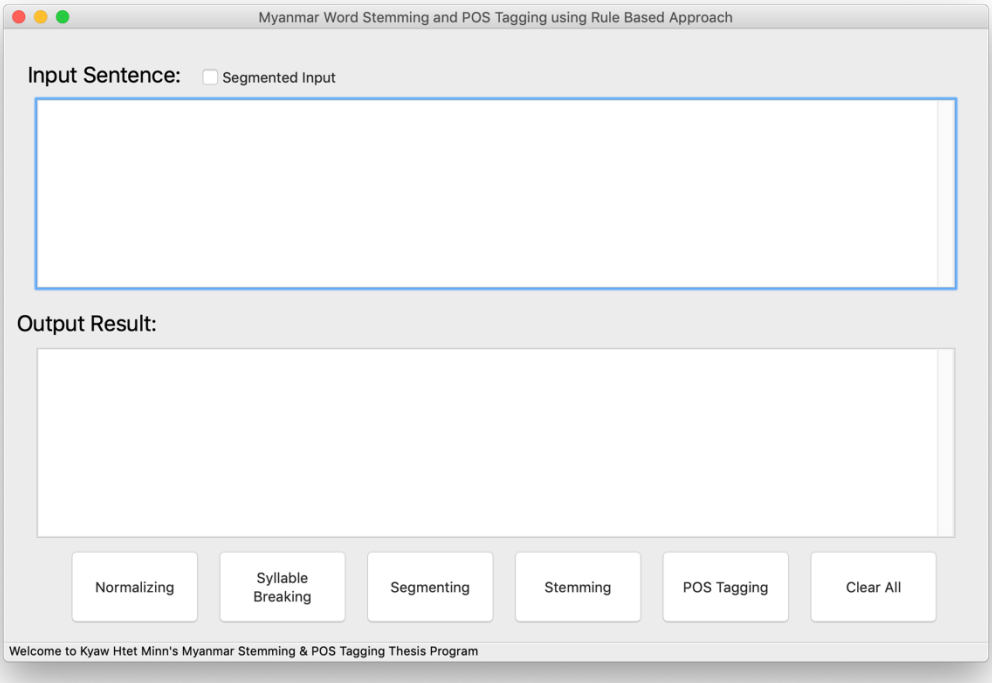

Figure 4.9 Main Screen of the System

4.6.1 Evaluation of the Unsegmented Myanmar Sentences

This system can accept both unsegmented and segmented sentences. The input text can be placed in the Input Sentence text box. For the unsegmented sentences, leave uncheck in the checkbox located above the input text area as shown in figure 4.10. The following testing data contains six sentences which are from the categories of social, sports, novel, business, news and law. The input sentence is typed in line by line format.

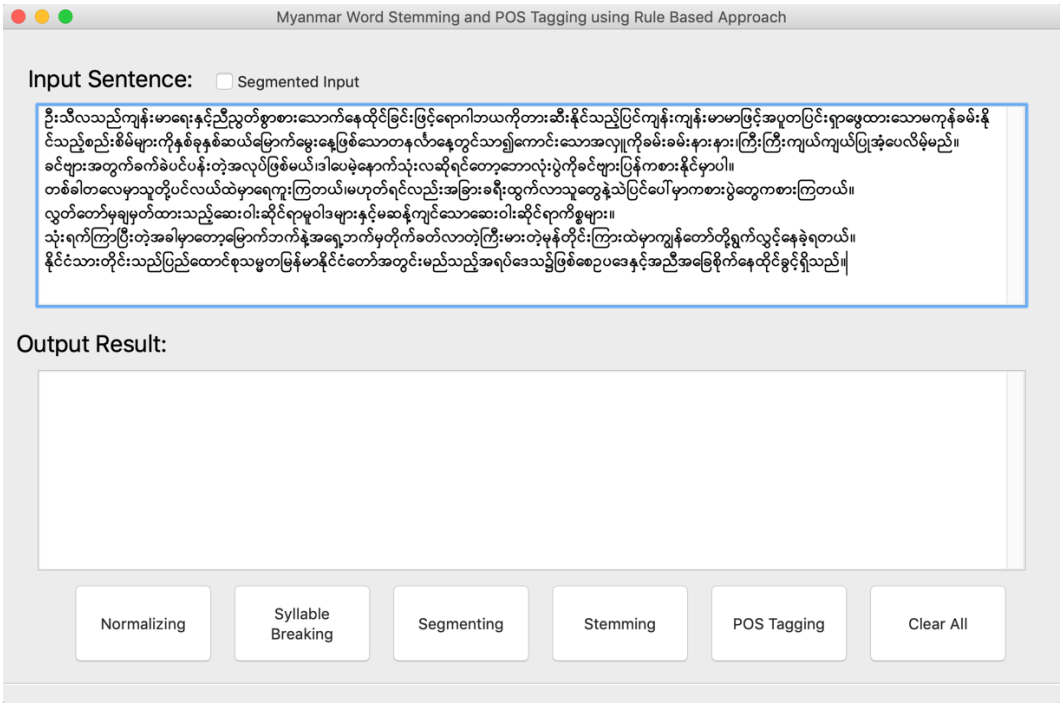

Figure 4.10 Input text of Unsegmented Sentences

After entering the input text and clicking the normalizing button, the result normalized text will be shown as in figure 4.11.

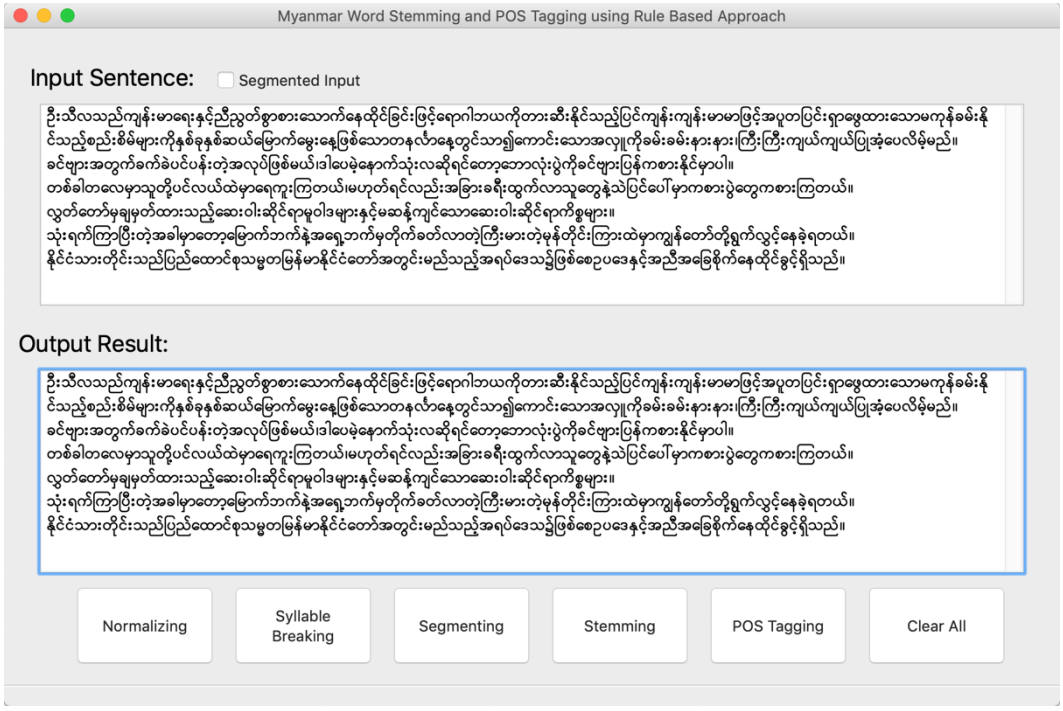

Figure 4.11 Output of Normalization tested with Unsegmented Sentences

After normalizing function, user can click the Syllable Breaking button. The result will be overwrite the previous data as described in figure 4.12.

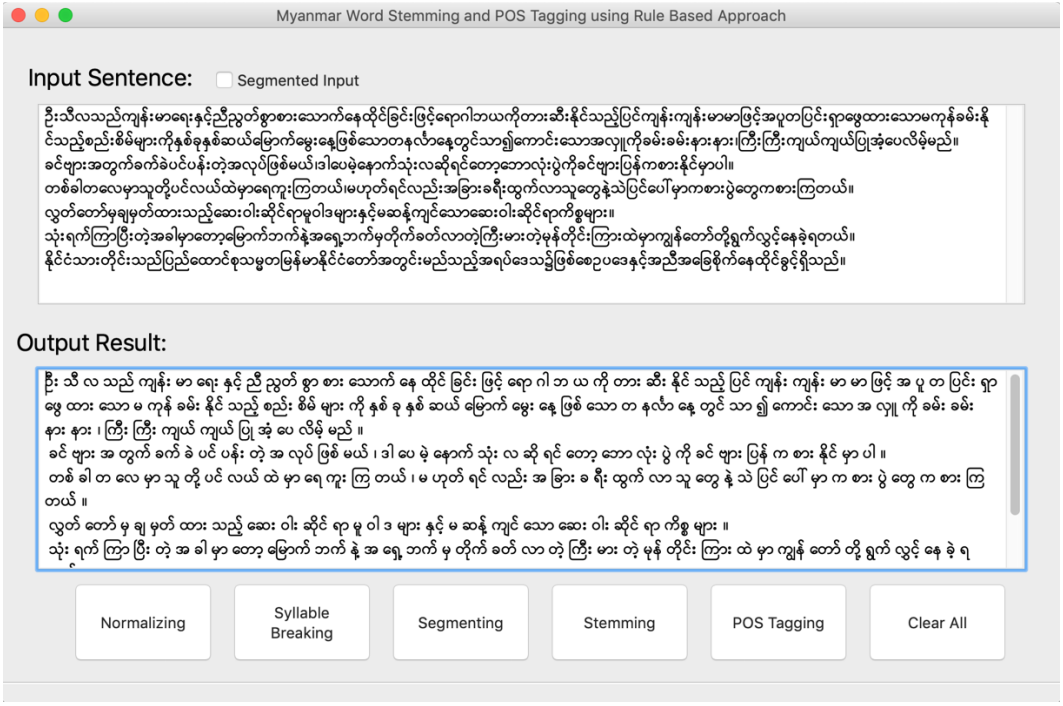

Figure 4.12 Output of Syllabification tested with Unsegmented Sentences

The next step is segmenting. The output of segmented text will be displayed as paragraph as shown in figure 4.13.

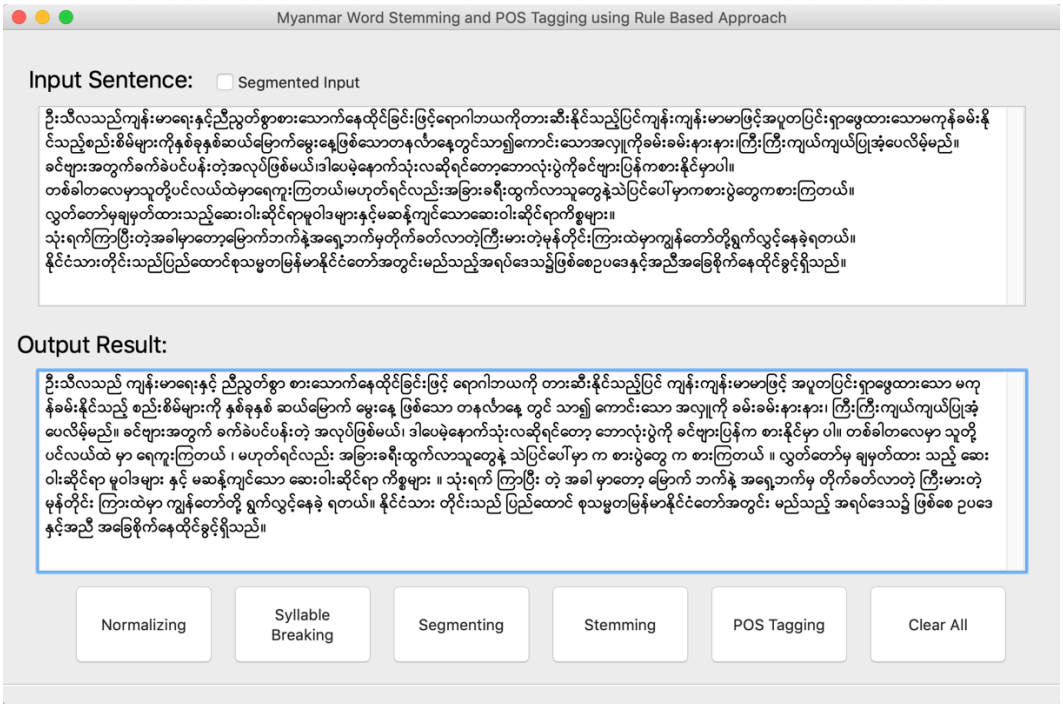

Figure 4.13 Output of Segmentation tested with Unsegmented Sentences

The stemmed word is stored as list format so that each stemmed word is displayed within single quotations marks as seen in figure 4.14.

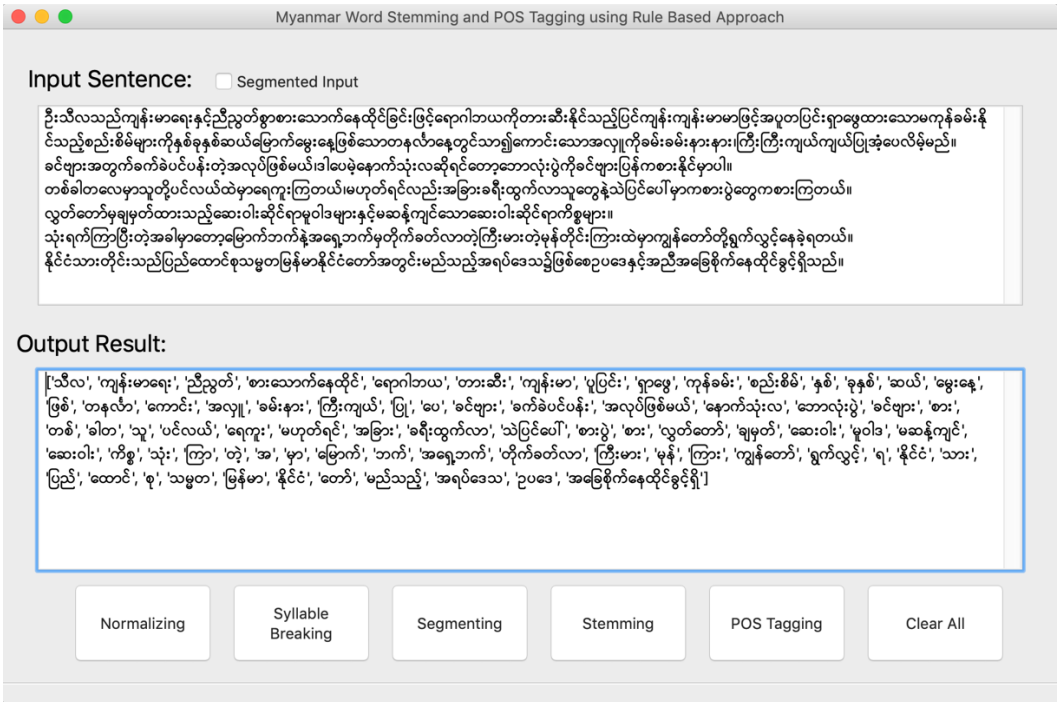

Figure 4.14 Output of Stemming tested with Unsegmented Sentences

The result of POS tagging is displayed with word and tag keyword connected with hyphen and separated with comma from the next word as shown in figure 4.15.

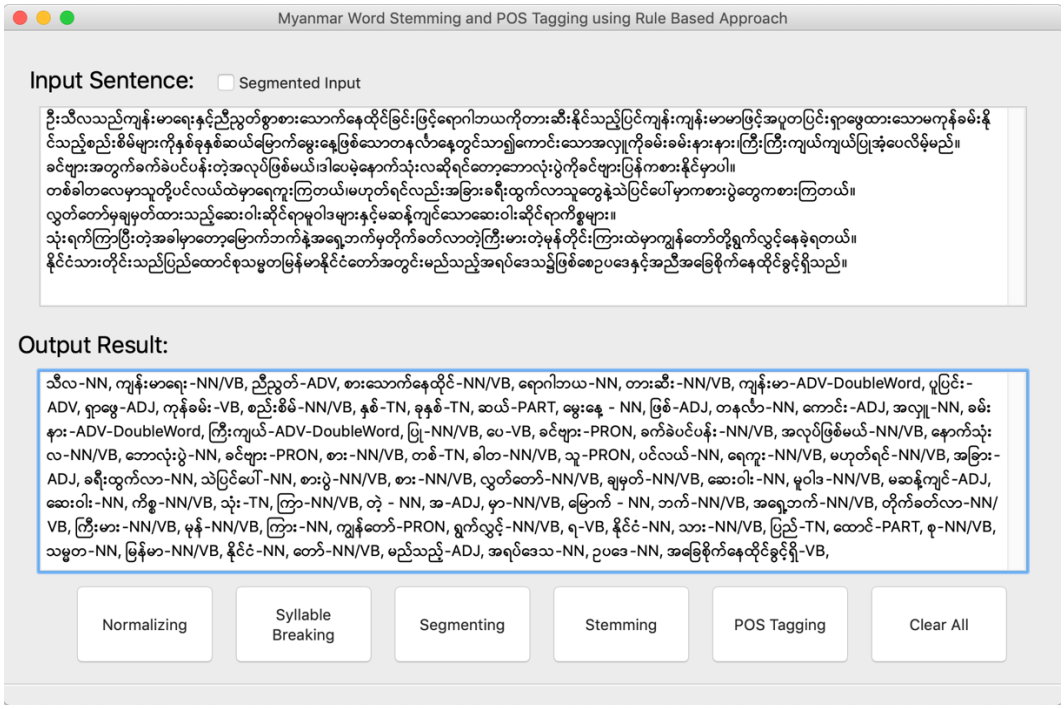

Figure 4.15 Output of POS Tagging tested with Unsegmented Sentences

4.6.2 Evaluation of the Segmented Myanmar Sentences

The figure 4.16 describe the input sentence with segmented Myanmar sentences.

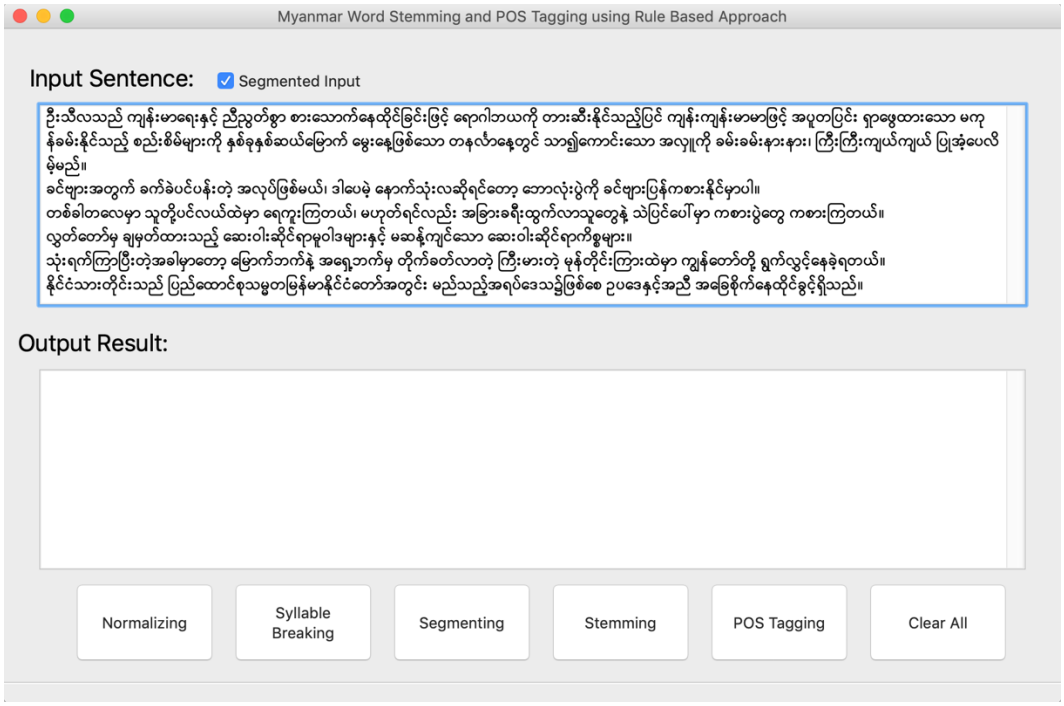

Figure 4.16 Input text of Segmented Sentences

The normalized function works the same as unsegmented sentences which is shown in figure 4.17.

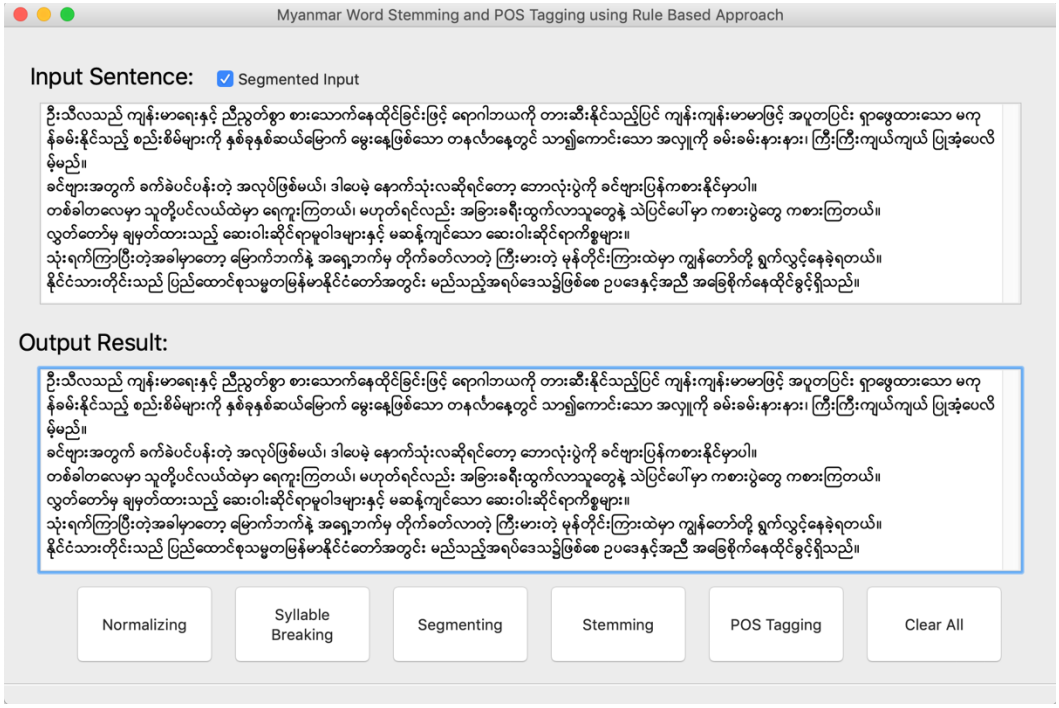

Figure 4.17 Output of Normalization tested with Segmented Sentences

In syllable breaking process, the result contains the extra space which is followed by the input segmented space as shown in figure 4.18.

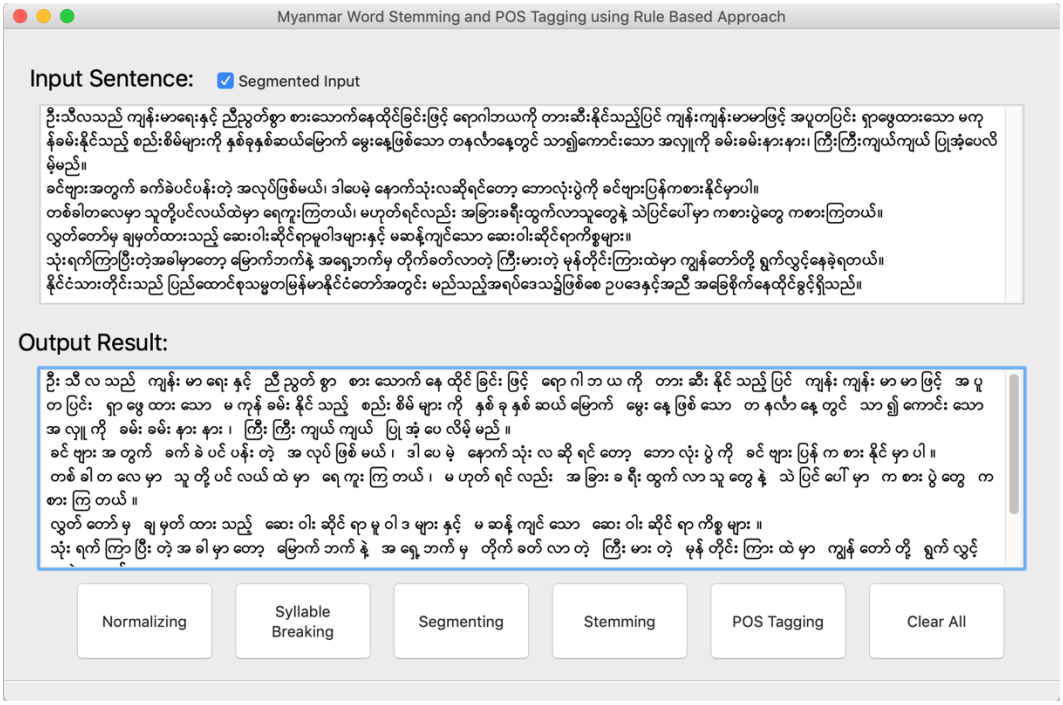

Figure 4.18 Output of Syllabification tested with Segmented Sentences

The input text is the output of syllabification and skips the segmentation step. The output of segmented stemming can be shown as in figure 4.19.

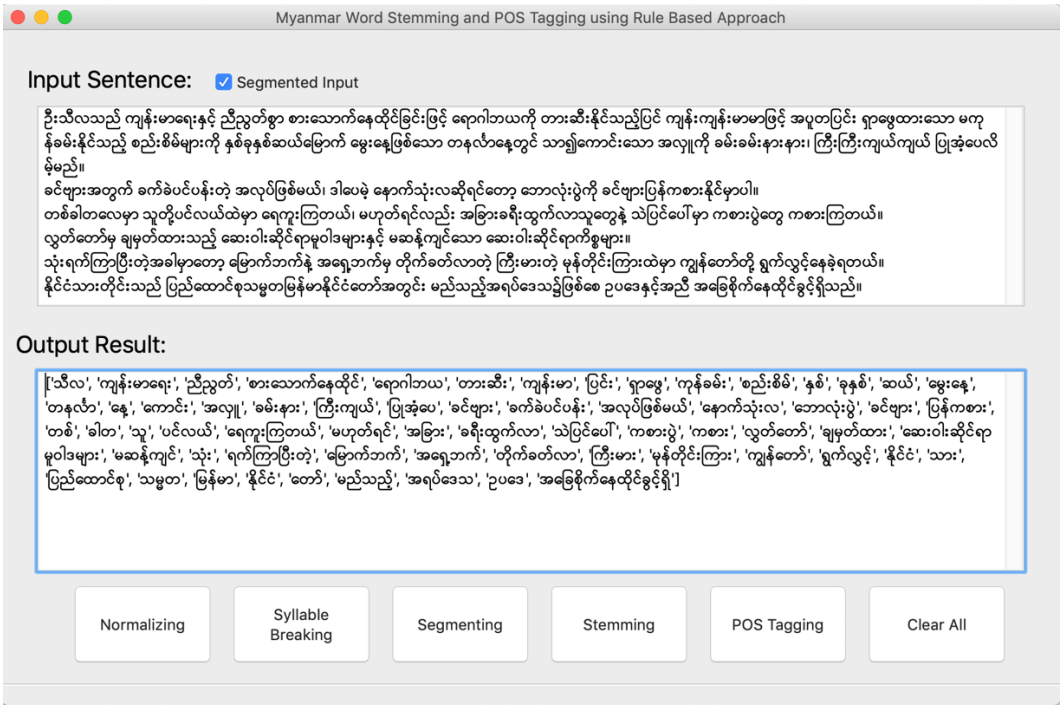

Figure 4.19 Output of Stemming tested with Segmented Sentences

The final step of POS tagging for segmented sentences is presented in the following figure 4.20.

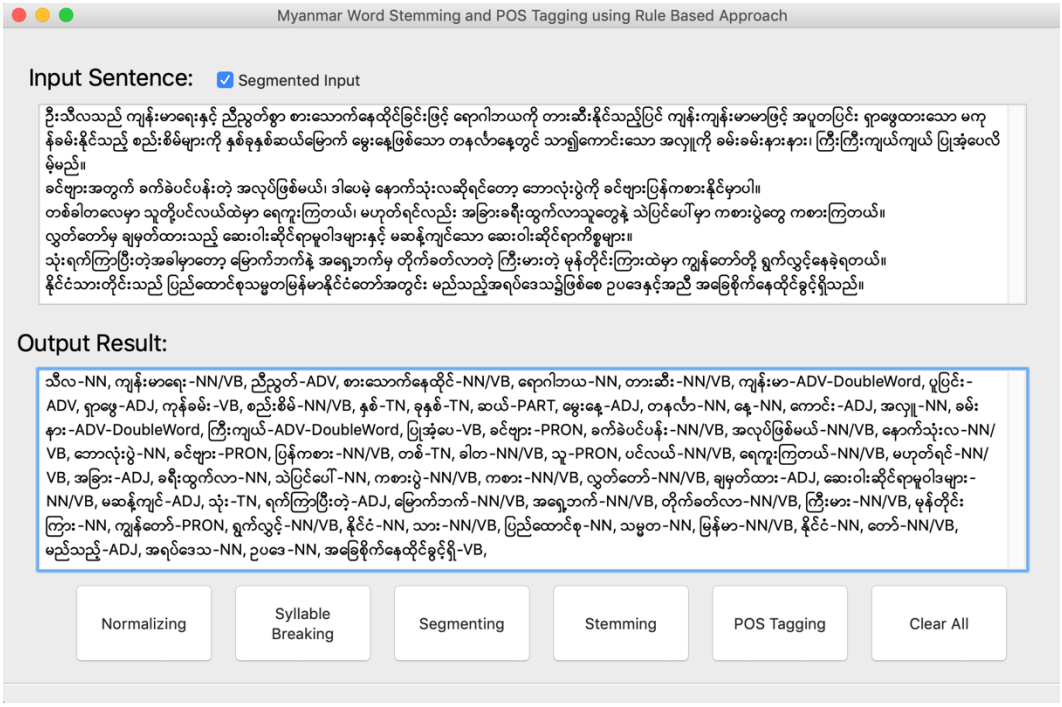

Figure 4.20 Output of POS Tagging tested with Segmented Sentences

4.7 Experiment Result

Some complete experimental results, performance measures and accuracy of the system for particular testing data are discussed in this session.

The performance of the developed system is measured by the accuracy. Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. The equation for calculating accuracy is defined as equation 4.1.

$$Accuracy = \frac{True\ Positive+True\ Negative}{True\ Positive+False\ Positive+False\ Negative+True\ Negative} \quad 4.1$$

,where
True Positive = the system intend to segment, stem, tag and the output is actually do
True Negative = the system is not intend to segment, stem, tag and the output do
False Positive = the system is intend to segment, stem, tag but the output do not
False Negative = the system is not intend to segment, stem, tag but the output do

The following table 4.1 shows the results for the evaluation of 300 sentences with segmented and unsegmented. The final calculation of segmented input sentences is 93% and the result of unsegmented input sentences is 89%.

Table 4.1 Accuracy of Segmented and Unsegmented Input Sentences

	Input with 300 segmented sentences	Input with 300 unsegmented sentences
True Positive (TP)	3108	3065
True negative (TN)	259	165
False Positive (FP)	207	359
False Negative (FN)	52	37
Accuracy	93%	89%

In this experimental results, the evaluation metrics for the data set are precision, recall and F1 Score. The calculated equation of precision is described in equation 4.2, recall in equation 4.3 and F1 Score in equation 4.4 respectively.

$$Precision(P) = \frac{Number\ of\ words\ correctly\ segmented,\ stemmed,tagged\ by\ the\ system}{Total\ number\ of\ words} \quad 4.2$$

$$Recall(R) = \frac{Number\ of\ words\ \ segmented,stemmed,\ tagged\ by\ the\ system}{Total\ number\ of\ words} \quad 4.3$$

$$F1\ Score\ (F1) = \frac{(\beta^2 + 1)\ PR}{\beta^2(\ R+P)} \quad 4.4$$

Where β is the weighting between precision and recall and typically β = 1.

The following calculation shows the evaluation result of 300 sentences which contain over 3000 words. Table 4.1 presents the result of segmented input sentences which the score is considerable higher than unsegmented input sentences. The F1 score for the unsegmented input sentences is described in table 4.2.

Table 4.2 Evaluation Results of Segmented Input Sentences

Category	P (%)	R (%)	F1 (%)
Stemming	90.35	93.87	91.59
POS Tagging	85.74	90.45	88.32
Average	88.05	92.16	89.96

Table 4.3 Evaluation Results of Unsegmented Input Sentences

Category	P (%)	R (%)	F1 (%)
Segmentation	83.67	88.80	86.16
Stemming	80.59	82.35	81.46
POS Tagging	73.78	83.95	78.54
Average	79.34	85.03	82.05

CHAPTER 5

CONCLUSION AND FUTURE WORK

This research work implemented Myanmar word stemming and POS tagging application using rule based approach. It includes five parts: normalization, syllabification, segmentation, stemming and POS tagging. The first part uses rules to do the normalization of Myanmar language. The next step, syllabification, uses a regular expression to break a sequence of words into syllables. After the syllabification step, the system uses N-gram approach to segment the sentence matching from the lexicon. The main and fourth step is stemming and it uses grammar rules and additional rules to strip the prefix, suffix and infix of the segmented word into stemmed word. To solve the POS tagging, the final step takes place into account by using 14 kinds of tag-set which is working under the same framework with stemming algorithm. The effectiveness of method on corpora in Myanmar has been evaluated.

The work of this system is based on three source: overall 300 sentences from Myanmar Wikipedia website, Myanmar Computer Federation (MCF) dataset and Myanmar Grammar Book published by Myanmar Language Commission. This system, tests and compares word segmentation techniques, stemming and POS tagging using rule based approach. The experimental results show that the system can get 80% to 85% accuracy and is also useful as a quick stemmer.

The output of this study can be used as further study research on text classification, categorization, information retrieval, machine learning, text mining and etc. However, no language in the world strictly follows a deterministic set of rules, so it is difficult to achieve this purpose systematically. That is why a perfect stemmer, which is able to accurately obtain the stems of any term independently of its features, does not exist. This paper algorithms are limited to work for words containing conjugated letters, only verb and noun inflections; the inflections for other parts of speeches are not considered.

5.1 Benefits of the System

There are some benefits in the developed system. This system implemented as a rule based approach so that it can work faster than other approach and require only low storage and processing power. Performance can be improved by adding rules and lexicons. This system is enable to use as a pre-processing tool in Myanmar text processing such as Machine Translation, Information Retrieval and Search Engine using Myanmar language. This system is developed under open source license that can be used freely for every research.

5.2 Limitation and Further Extension

This system is organized by a total of over 3,000 words that have been manually spell-checked. Segmentation and stemming error can occur when the words are not listed in predefined lexicon. In current research, not every lexicon contains for every possible word of Myanmar language. There always exist out-of-vocabulary words such as new derived words, new compounds words, morphological variations of existing words and technical words. Moreover, there have compound words that cannot be stemmed, additional rules can be conflict with each other and complex grammar rules are required. The overall accuracy of the current system is over 80% but, the accuracy can be improved by adding lexicons in all three types. It can be compared with other rule based algorithms and also with other types of stemmer to construct a hybrid approach. It can be used as the rough data for specific POS tagging research.

AUTHOR'S PUBLICATION

[1] Kyaw Htet Minn, Khin Mar Soe, *"Myanmar Word Stemming And POS Tagging Using Rule Based Approach",* to be published in the Journal of Parallel and Soft Computing (PSCJ 2019), Yangon, Myanmar, 2019.

REFERENCES

[1] A. Paul and A. Dey, *"An Affix Removal Stemmer for Natural Language Text in Nepali"*, International Journal of Computer Applications(0975-8887), Volume 91, No.6, April 2014.

[2] C. C. Ding, Y. K. Thu, M. Utiyama, and E. Sumita, *"Word Segmentation for Burmese (Myanmar)"*, Association for Computing Machinery (ACM) Transactions on Asian and Low-Resource Language Information Processing, Volume 15, No.4, pp. 1-22, June 2016.

[3] C. D. Paice, *"An Evaluation Method for Stemming Algorithms"*, Proceedings of the 17th Annual International Association for Computing Machinery Special Interest Group on Information Retrieval (ACM SIGIR) Forum, pp. 42-50, 1994.

[4] C. D. Paice, *"Another Stemmer"*, Proceedings of the 13th Annual International Association for Computing Machinery Special Interest Group on Information Retrieval (ACM SIGIR) Forum, Volume 24, Issue 3, pp. 56-61, Fall 1990.

[5] E. Tamah, Shammari-Al, *"Towards An Error- Free Stemming",* International Association for the Development of the Information Society (IADIS) European Conference on Data Mining, pp. 160-163, 2008.

[6] H. H. Htay, K. N. Murthy, *"Myanmar Word Segmentation using Syllable Level Longest Matching"*, Proceeding of the 6th Workshop on Asian Language Resources, pp. 41-48, 2008.

[7] J. B. Lovins, *"Development of a Stemming Algorithm",* Mechanical Translation and Computational Linguistics, Volume 11, pp. 22-31,1968.

[8] J. Dawson, *"Suffix Removal and Word Conflation",* Association for Literary and Linguistic Computing (ALLC) Cambridge bulletin, Volume 2, Issue 3, pp. 33–46, 1974.

[9] J. Savoy, *"Stemming of French Words Based on Grammatical Categories",* Journal of the American Society for Information Science, Volume 44, Issue 1, pp. 1-9, January 1993.

[10] M. Ali, S. Khalid, M. H. Saleemi, *"A Rule Based Stemming Method for Multilingual Urdu Text"*, International Journal of Computer Applications (IJCA), Volume 134, No.8, pp. 10-18, January 2016.

[11] M. F. Lynch, P. Willett and F. Ekmekcioglu, *"Stemming and N-gram Matching for Term Conflation in Turkish Texts"*, Information Research News, Volume 1, No. 1, pp. 2-6, 1996.

[12] M. F. Porter, *"An Algorithm for Suffix Stripping",* Automated Library and Information Systems Program, Volume 14, Issue 3, pp. 130−137, 1980.

[13] Myanmar Language Commission (MLC), "Myanmar Thuddar", Department of the Myanmar Language Commission, Yangon, Ministry of Education, Government of the Union of Myanmar, Volume 1-3, Date of publication: 2016.

[14] Myanmar Language Commission (MLC), "Myanmar-English Dictionary", Department of Myanmar Language Commission, Yangon, Ministry of Education, Government of the Union of Myanmar, Date of publication: 1993.

[15] R. Kansal, *"Rule Based Urdu Stemmer",* In Proceedings of the 21st International Conference on Computational Linguistic (COLING 2012): Demonstration Papers, pp.267-275, December 2012.

[16] R. Krovetz, *"Viewing Morphology as An Inference Process",* In Proceedings of the 16th Annual International Association for Computing Machinery Special Interest Group on Information Retrieval (ACM SIGIR) Forum, pp. 191-202, July 1993.

[17] W. B. A. Karaa, *"A New Stemmer to Improve Information Retrieval",* International Journal of Network Security & Its Applications, Volume 5, No. 4, pp. 143-154, July 2013.

[18] Web link: https://machinelearningmastery.com/natural-language-processing/
Date of Access: February 15, 2019.

[19] Web link: https://www.myanmarlanguage.commission.myn.asia/
Date of Access: February 15, 2019.